ZERO ROBOTICS

SPHERES ISS CHALLENGE

Middle School Program

There Is Nothing to SPHERE But SPHERES Itself!

EDUCATOR GUIDE

Acknowledgments

This curriculum guide was developed by the Massachusetts Institute of Technology's Space Systems Laboratory in partnership with Education Development Center, Inc. (EDC), Massachusetts Afterschool Partnership (MAP), Aurora Flight Sciences (AFS) and the under/graduate students and staff of the Zero Robotics (ZR) program. Zero Robotics is led by MIT and made possible with support from the Center for the Advancement of Science in Space (CASIS), Northrop Grumman, US DoD's Defense Advanced Research Project Agency (DARPA), TopCoder, the National Aeronautics and Space Administration (NASA) and the crew of the International Space Station (ISS).

We would like to extend special acknowledgements to the following individuals for their support of the Zero Robotics program and the development of this curriculum:

Dr. Alvar Saenz-Otero, MIT Space Systems Laboratory
Wendy Feenstra, Aurora Flight Sciences
Jacob Katz, Space Exploration Technologies
Katie Magrane, Innovation Learning Center
Aviva Siegel, Massachusetts Afterschool Partnership
Barbara Morgan, Former Astronaut & Distinguished Educator in Residence at Boise State University
Dr. Jeffrey Hoffman, Former Astronaut & Director of MIT Man Vehicle Laboratory
Dr. Lorna Finman, North Idaho STEM Charter Academy
Kristin Bjork, Education Development Center
Sarita Pillai, Education Development Center
Rebecca Lewis, Education Development Center

We would like to thank program teachers and student mentors for their feedback on the pilot version of this program—their experiences and insight were invaluable in the development of this guide. We would like to acknowledge the following statewide partners for supporting pilot programs across the country:

California Afterschool Network & NASA AMES Research Center
Florida Afterschool Network, CASIS & NASA Kennedy Space Center
Barbara Morgan & Boise State University
Georgia Statewide Afterschool Network & Georgia Institute of Technology
Massachusetts Afterschool Partnership & Massachusetts Institute of Technology

We hope that this guide will allow a range of educators to implement the middle school Zero Robotics program with youth who are interested—or who might be inspired to pursue—science, technology, engineering, and mathematics.

Table of Contents

Program Overview

Zero Robotics (ZR) is a programming competition where SPHERES satellites (the robots) inside the International Space Station (ISS) are controlled by programs developed by students. In the ZR Middle School Program, student teams create, edit, share, save, simulate, and submit programming code in order to accomplish the task at hand (the competition changes each year). After several phases of competition, regional finalists compete in a championship aboard the ISS. An astronaut will use students' code to conduct the championship competition in microgravity with a live broadcast!

The ZR Middle School Program is a five-week program. The curriculum is designed assuming students participate approximately 15 hours per week over the course of the five weeks (75 hours total). The ZR curriculum can be distributed as desired and incorporated into other programming; for example:

- 5 days per week with 3 hours ZR per day
- 3 days per week with 5 hours ZR per day
- 4 days per week is also successful since it avoids problems with student absences on Fridays

The suggested flow of activities provided in this guide is designed to be tailored to support a variety of programming schedules.

The ZR Middle School Program has several specific objectives. It is designed to engage students in science, technology, engineering, and mathematics (STEM) through the ZR game and to make clear connections between STEM and space science. The curriculum integrates best practices for teaching and learning, including inquiry and collaboration, and encourages authentic assessments of student learning (in particular, formative assessments). Throughout the five weeks, connections are drawn between the ZR game, the middle school ZR program in its entirety, and the STEM content being taught in the middle grades and beyond. There are explicit linkages to STEM educational pathways and careers aimed at informing students about the many varied opportunities for success that STEM fields can offer. The program also provides multiple avenues for educators to grow professionally while supporting students at varying levels of skill and interest in STEM content and computer programming.

It is essential that today's young people understand STEM topics. The primary goal of this program is to excite middle-grades students about STEM, with the hope of encouraging them to pursue their studies in these fields so that they can enter STEM careers. And since this is an informal education program, it should also be fun! While there is a lot of content to cover, it is also important to provide students with many opportunities to get up, move around, and enjoy their time. Therefore, we provide a number of "ice breakers" (see **Appendix 1**) to supplement the programming and STEM activities. Particularly in the first week of the program—where activities that allow students to move around are built into the curriculum—and while students are programming in the third and fourth weeks, it is important to take frequent breaks and have fun!

Getting Started

This section provides some overall information to help you prepare to implement the ZR Middle School Program.

This guide is designed to be read electronically using a pdf application, which allows external links to be included. Teaching from a printed version of the guide is not recommended as you will not be able to follow external links to activities and programming tutorials.

Companion student materials are available at http://zerorobotics.mit.edu/ms/. Students will be invited to access and use these materials as they progress through the curriculum.

Before You Teach

- Make arrangements for field trips or speakers.
    - The following required field trips will be organized in conjunction with your regional State After-School Network (SAN):
        - Field Day: This field trip occurs at the beginning of Week 2 and includes a tour of a college campus, NASA/space institution, local planetarium or science museum.
        - ISS Finals Event: Details will be provided separately.
    - For a list of optional field trips that you can choose to organize yourself, see **Possible Field Trips and Speakers** on page 147.
- Review the program outcomes, information about the alignment between the ZR curriculum and national standards, and information about reflection and assessment within the ZR program.
- Review the Program Schedule, which previews the concepts and activities in each week of the program, beginning on page 7.
- Review the materials needed for the program (see **Appendix 2** for a complete list). Note that for the Bottle Rockets activity in Week 1, you will need a number of 2-liter soda bottles (Coca-Cola Company bottles seem to work best), which may take some time to collect. In addition, you may want to either purchase a bottle rocket launcher or build your own.

  - While we do not endorse any specific product, the following bottle rocker launcher has been tested by ZR staff: Aquapod-Bottle-Launcher. Note that to use this bottle rocket launcher you will also need to obtain a bicycle pump with a built-in pressure gauge.
  - NASA has a Water Rocket Launcher rocket activity (available at http://www.nasa.gov/pdf/153405main_Rockets_Water_Rocket_Launcher.pdf) that details the materials needed and instructions for building your own launcher.
- Set up an account on the ZR website, following the directions in the PowerPoint presentation for the *Create an Account* tutorial in the student materials, and review *How to Invite Students to Your Team* (a tutorial for you, the educator).
- Run through each of the other tutorials:
  - Week 1
    - *Getting to Know the ZR IDE*
    - *Introduction to Arrays and the setPositionTarget Function*
    - *More Simple Arrays and the setAttitudeTarget Function*
    - *More Simple Arrays—Another Way to Initialize Variables*
  - Week 2
    - *The Conditionals: The Basic of "If-Then"*
    - *Conditionals: More Fun with "If-Then" and Logic Operators*
    - *Conditionals with Advanced Logic Operators*
    - *Conditionals Continued: "If-Then-Else"*
    - *Introduction to Game Mode*
  - Week 3
    - *For Loops*
    - *Hints About SPHERES Loop Dynamics*
    - *Applied Conditionals*
    - *Creating Functions*
    - *Functions and the Step Counter Model*
    - *Intramural Game Mode*
    - *How to Submit Code*
- Review the *Game Manual* in the student materials, which explains the rules for this year's game (the game changes from year to year).

Outcomes

Students will develop a range of 21st century skills and will advance their understanding of both STEM content and career opportunities. In addition, this program represents a professional development opportunity for educators who may be familiar with science and mathematics but are less familiar with computer science and engineering.

Student Outcomes

- Improved technology skills
- Improved critical 21st century skills, including communication, collaboration, critical-thinking, and problem-solving skills
- Increased engagement in and excitement about learning STEM content, particularly technology
- An understanding of the connection between what they're learning and its relevance to their life, both now and in the future

Educator Outcomes

- Improved technology skills, especially around computer science and programming
- Strengthened instructional practice
- An understanding of the value of integrating technology into teaching

Alignment to Curriculum Standards

The Common Core State Standards and Next Generation Science Standards are two national efforts to provide a clear, coherent, and consistent framework and related benchmarks for the skills and knowledge that students need to succeed in college and careers. This curriculum provides a rigorous and engaging informal experience that will allow your students to achieve these new standards (see **Appendix 3** for more information).

Activities

Activities in the ZR program are color-coded with circles to denote the activity type. Activity types are as follows:

- Exploration
- Instruction
- Active Activity

- Computer Programming Tutorial
- Game
- Assessment & Reflection

Reflection and Assessment

Throughout the ZR program, you should encourage students to reflect on what they are doing and assess their mastery of concepts and skills. We have provided specific reflection and assessment items for the end of each week, but we also encourage you to provide ongoing opportunities for student reflection. For example, you might have students develop a blog or video portfolio that documents their work on programming the player for the ZR game or for the program overall. For a video portfolio, you can interview students, have them interview one another, or allow them to reflect on their own.

Regarding the latter, it is helpful for students to have reflection modeled for them, so try to reflect on your own experiences and learning using whatever format(s) you ask them to use. Share your own professional development goals with students at the beginning of ZR—for example, if you aren't familiar with computer science and programming, you can share that you hope to learn along with students. Then reflect on your progress—are you feeling more comfortable as the ZR program progresses? Do you wish you had more support? The best way to get students to open up is to be open yourself!

Questions for Reflection

For ZR overall: What did you learn this week? How do you know you learned it? What got in the way of your learning? What helped your learning? How do you feel about the ZR program and your participation in it?

For ZR programming: How far along are you with developing your code (whether team code, class player, or regional)? What challenges are you having? What are some possible solutions that will help you address these challenges?

Schedule

	Week 1	Week 2	Week 3	Week 4	Week 5	Championship!
Science and Mathematics Concepts	• Cartesian coordinates • Forces • Grids and graphing	• Cartesian coordinates • Dimensions • Forces • Grids and graphing • Motion • Newton's Laws • Order of operations • Vectors	• Kinematics • Mass vs. Weight • Speed vs. Velocity		• Dynamics • Degrees of Freedom	
Science and Mathematics Activities	• Bottle Rockets, Part 1 • Mystery Coordinate Grid • *Optional:* Coordinate Hunt and/or Battleship	• Seeing in 3-D—How to Visualize Space! • Vector Hunt • Thruster Balloons • *Optional:* Chairs on Wheels • Optional: Dry Ice Dynamics	• Mapping with Speed • Kinematics Scenario		• STEM programs and careers • Smooth Moves • How Many Degrees of Freedom Do You Have? • Bottle Rockets, Part 2	
Computer Science Concepts	• Conditional statements • Data types • Debugging • Logic • Loops • Variables	• Functions • Logic operators • Operators				

	Week 1	Week 2	Week 3	Week 4	Week 5	Championship!
Computer Science Activities	• What's the Logic? and/or *Optional:* Robot Logic • Logic Competition	• Field Day • Conceptualizing team strategy (Field Day, Let the Games Begin)	• Devising team strategy • Acting out and refining team strategy	• Devising class strategy • Acting out and refining class strategy		
Programming Tasks	• *Create an Account* tutorial • *Getting to Know the ZR IDE* tutorial • *Introduction to Arrays and the setPositionTarget Function* tutorial • *More Simple Arrays* tutorials (2)	• *The Conditionals* tutorials (4) • *Introduction to Game Mode* tutorial	• Tutorials: ○ *For Loops* ○ *Hints About SPHERES Loop Dynamics* ○ *Applied Conditionals* ○ *Creating Functions* ○ *Functions and the Step Counter Model* ○ *Intramural Game Mode* ○ *How to Submit Code* • Programming players	• Programming regional player • Reviewing regional winner's/own winning code and refining		
Competition	None	None	• Intramural Competition • Practice Regional Competition	Regional Competition	Prepare for finals	National Competition

Week 1:

To Infinity and Beyond . . .

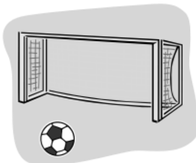
Overview

During the first week of the ZR program, students are introduced to ZR and begin to learn about the SPHERES, the ZR game, and computer programming. In addition, they are introduced to space and the ISS. Given that there are fewer physical activities integral to the curriculum during this week, it is especially important to use some of the “ice breakers” from Appendix 1 to give students the opportunity to get to know one another and be active.

Getting Started

Before Week 1, review the information below and gather the materials needed. In addition, be sure that the Week 2 Field Day arrangements are settled (see **Possible Field Trips and Speakers** on page 147 for more details).

Goals



Students understand what SPHERES are, how the middle school ZR game relates to the SPHERES satellites, and what the ZR Integrated Development Environment (IDE) is and how it works.

Student Learning Outcomes

By the end of this week, students should do the following:

- Understand the basic overview of the middle school ZR program and that they will be programming satellites on the ISS Understand that the SPHERES satellites are maneuvered by computer programs, rather than by humans
- Understand the "problem" they must address in their player design
- Understand the basic structure and elements of a computer program and what is involved in debugging a program
- Be aware that math and physics are involved in maneuvering the SPHERES satellites
- Be able to complete short programming tasks in the ZR IDE using the instructions provided
- Be able to work together in teams to solve problems

NOTE: The ZR IDE is a tool students will be using to do the programming. It provides them with what is known as a high-level programming language. This is a computer language that makes programming easier and more user-friendly for those who have not programmed before. It is a great way to learn the logic of programming before learning the trickier syntax of other programming languages. The high-level language students will use for ZR uses the dragging and dropping of various "puzzle piece" procedures, variables, etc. to teach you to think like a programmer.

Before You Teach

- Make copies of the reproducible masters (RMs) needed for each activity.
- Review the suggested flow of activities below and estimate what activities will be scheduled for each day based on the time you have.
- Preview and select which video clips and other multimedia related to space and space exploration to share with students. Make sure that selected videos can be played at your site.
- Preview the student materials for Week 1, including the slideshows.
- Review the **Background Information** (beginning on page 124) for the programming concepts introduced and used this week: *conditional statements, data types*, *debugging*, *logic*, *loops*, *Newton's laws*, and *variables*.

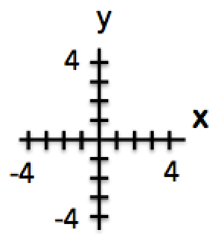
- Prepare and post three sheets of newsprint (marked with "K," "W," and "L," respectively).
- For "Moving Off Earth," review the directions for building a model of the ISS; there are three sections, so divide the group into three teams (or into six teams and have them build two models).
- For "Bottle Rockets, Part 1," review the materials list for this activity. Note that when the bottle rocket is pressurized, it can cause severe injury. Please review and follow the safety precautions listed on **Make Your Own Rocket! (RM 1.1).** Also, be sure to select an open area—clear of people, cars, windows, and so on—for your launch site.
- For the **Mystery Coordinate Grid** in "Try Your Hand at Programming," you will need to display the following table on newsprint or chart paper:

Point	(*x* coordinate, *y* coordinate)
A	(1, 3)
B	(–1, –2)
C	
D	
E	
F	

- For the optional Coordinate Hunt (an extension of "Try Your Hand at

Programming") you will need a significant amount of time to set up the course:

- Print the **Coordinate Hunt Pieces** double-sided (the front and back coordinates will correspond) and cut out the clues along the lines.
- Use the clues themselves to create the grid coordinate system as shown here:

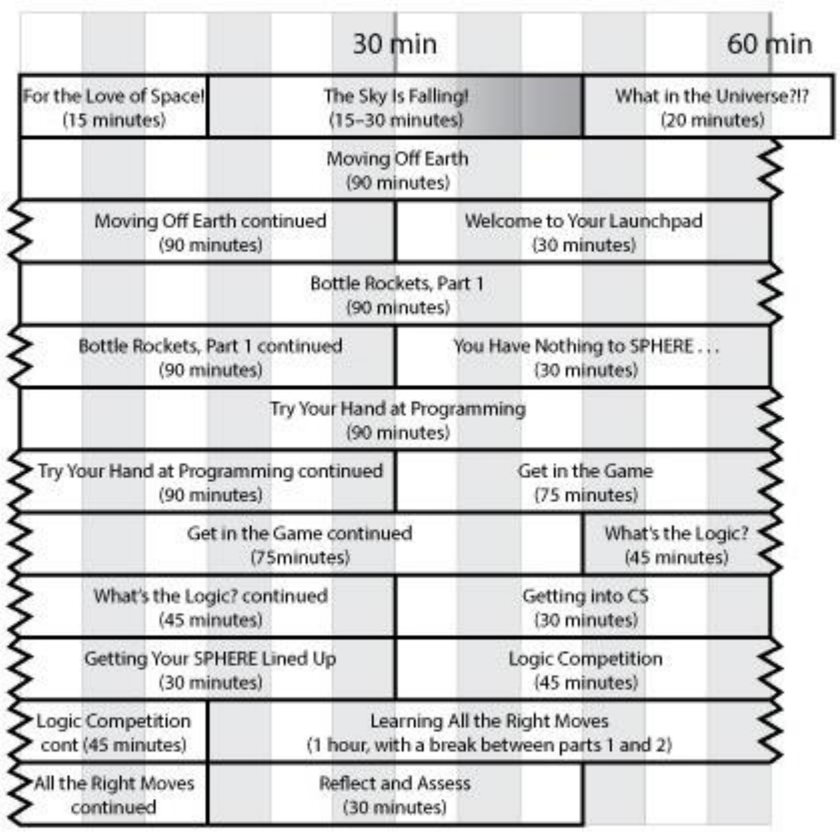


(81 clues total)

  - The clues can be set up on desks, around the room, or in a grid outside the classroom (secure clues so they don't get knocked off desks or blown away).
  - The farther apart the clues are, the better (outside on a field would be optimal), as it forces students to think spatially where the next clue is in terms of a coordinate system so they do not waste time—*and* they get to move around a lot more.
  - Alternatively, create four game boards, and make the clues small enough to fit on the game boards.

- Build the three structures you will need for "Logic Competition."
- Determine which items from the "Reflect and Assess" section you will use.

Suggested Flow of Activities*



* Note that "ice breakers" and other activities to get students outside and moving around are not included in this schedule; you should incorporate these based on your students' needs.

Activities

Bring It Home

On the first night of the program, send home an introduction letter for students to share with their families. You can use the provided Family Letter Template on page 150 to craft this letter.

Further on in the program, you might want to send letters home with other "Bring It Home" materials to explain the purpose and content of the activities that students are engaging in.

For the Love of Space! (15 minutes)

Materials Needed

For each student:

- Program Introduction Letter—see box above

For the group:

- Computer with Internet connection and projection equipment or TV and DVD

Activity Description

Your students are about to embark on a five-week program about programming and space. This short kick-off activity is designed to generate excitement, give them a broad perspective about space and the solar system and why learning about, and understanding, them is important.

1. Why do we explore space? Show students the *Why We Explore* video (2:35) from NASA to answer this question (available in the student materials).
2. Show students a video (4:32) on the solar system (available in the student materials) and ask some questions about the content of the video, such as:
   a) What does the video tell you about the origins of the solar system?
   b) What are the planets in our solar system and what does the video tell you about what makes a planet a planet?
   c) What did you learn from this video that is new and exciting for you?

The Sky Is Falling! (15–30 minutes)

Activity Description

High drama is often very effective with young people. For this introduction to ZR, we suggest that you begin by looking at the potential for an asteroid collision with Earth and then move into activating students' prior knowledge about space.

Show a video clip and/or look at data:

1. Optional: Show a brief clip from either *Armageddon* (1998) or *Deep Impact* (1998).
2. Have students read the "The Sky Is Falling!" section in the student materials. Point out the listed asteroids that have—or are predicted to—come near Earth.
3. Ask students to think about why you have asked them to consider these data.
4. Begin a K-W-L on space and space science with the group.

   Note: Students should recognize that the ZR program relates to space and space science. If they do not mention NASA and its work, you might want to share with them the organization's mission, which is space exploration and understanding the place of Earth in the universe (NASA- What's next).

   a) Ask students to share what they know about space and space science, and record their answers on the "K" sheet.

   b) Ask students to share what questions they have or what they would like to know about space and space science, and record their responses on the "W" sheet.

Materials Needed

For the group:
- Three prepared sheets of newsprint (see "Before You Teach" on page 10) plus extra sheets of newsprint
- Markers

Optional:
- Computer with Internet connection and projection equipment or TV and DVD
- Optional: Clips from *Armageddon* and/or *Deep Impact*

c) Keep all three sheets posted in the room.

> *Teacher Tip: Revisit the "W" sheet every other day for the first two weeks, then weekly thereafter, and add any new questions students have. During discussions, encourage students to refer to the K-W-L sheets. At the end of each week, and again at the end of the program, ask students to fill in the "L" sheet with what they have learned).*

What in the Universe?!? (20 minutes)

Materials Needed

For the group:

- Computer with Internet connection and projection equipment
- Some short video clips and/or other multimedia highlighting space and space exploration, such as the Mars rover landing, the ISS, and the moon landing, selecting from the following resources:
  - NASA Johnson Style (3:47)
  - Space Shuttle Launch (3:52)
  - Challenges of Getting to Mars: Curiosity's Seven Minutes of Terror (5:08)
  - Top 5 Historical NASA Videos (24:99)
  - Mars in a Minute: How Do Rovers Drive on Mars? animation (1:00)

Activity Description

For this introductory activity, we recommend sharing with students some visuals related to space and engaging them in thinking more about what they know and wonder about space and space exploration.

1. Show the short video clips and other multimedia highlighting space and space exploration that you chose.[†]

2. Ask students if they would like to add anything to the "W" sheet for the K-W-L. Record their responses.

[†] Some of these videos are also available in the student materials.

Moving Off Earth (90 minutes)

Activity Description

This activity introduces students to the ISS, where the SPHERES satellites "live" in space.

1. Show students the ISS using the resources chosen from the list below (or have students explore the resources).

> ***Teacher Tip:*** *We suggest that you show at least the first video, Further Up Yonder: Time Lapse of the Earth from the ISS, to the group.*

- Further Up Yonder: Time Lapse of the Earth from the ISS (2:28)
- Everything About Mission Control (7:40)
- See How the ISS Was Assembled (2:13)
- Virtual Tour of ISS

> ***Note:*** *Show students how to click on arrows to move around station and click on white triangles inside blue circles to watch narrated videos of points of interest and then let them explore.*

- A Narrated Tour of the ISS

> ***Note:*** *"Station Tour: Harmony, Tranquility, Unity" (8:41) shows you life on the ISS including where and how the astronauts sleep, what they eat and the bathroom facilities.*

2. Use the ISS Facts PowerPoint presentation to see whether students picked up any of the included facts about the ISS.

3. Have students build a model of part of the ISS using the directions provided by NASA.

Materials Needed

For each student:
- Computer with Internet connection and headphones (optional)

For each team:
- Scissors (at least one pair)
- Tape (regular clear tape, masking or duct tape)
- Glue (optional)

For the group:
- Computer with Internet connection and projection equipment to show videos and ISS Facts PowerPoint presentation
- Visual resources to explore the ISS, chosen from the list in step 1 of the activity description (available in the student materials):
- Heavy 8½" x 11" paper (to print ISS paper model templates from NASA on)
- Nylon fishing line
- Wood skewers or ice pop sticks (for one team)

❸

Welcome to Your Launchpad (30 minutes)

Materials Needed

For the group:

- Computer with Internet connection
- Email address

For the group:

- Computer with Internet connection and projection equipment to show the PowerPoint presentation for the *Create an Account* tutorial
- Group list for students to record their username and password

Activity Description

Introduce students to the ZR website and the ZR Integrated Development Environment (ZR IDE), where they will work throughout the program.

❶ Explain that the ZR IDE is students' "workbench"—this is where they will write their computer code and perform simulations once they begin programming.

❷ Have each student set up a Google account that they can use to log in to the ZR website, which will allow them to gain access to the world of Zero Robotics:

a) Walk students through the process by showing the PowerPoint presentation for the *Create an Account* tutorial in the student materials, which is the first tutorial for students.

b) Show students the tutorial slide by slide, having them follow the steps to create their own accounts. Ask each student to record his or her Google Account email address and password on the class list in order to avoid lost information.

c) For students to proceed with slide 5, you will need to invite them to join your team. You will have been assigned a team number and made a team leader so that you can invite each of your students to join your team. Instructions for how to invite your students to your team are included in the *How to Invite Students to Your Team* tutorial that you previewed before beginning the program. You will need the information that you collected about the email address for each student to do this. It will take a little time to complete this step so you may want to plan to let students explore the ZR website or take a break while you are doing this.

d) Have students check their email and accept their team invitation. Congratulate them on joining the ZR team!

What's Going On?

The ZR IDE allows students to create programs to move the SPHERES satellites. Students will be programming extensively in the ZR IDE over the next four weeks. The purpose of this first tutorial is to get them familiar with the ZR IDE by creating an account and a password they will use to log in to the website (from where they will access the ZR IDE). In addition, students will be invited to join a ZR team (for which they have received an email invitation). Once logged in, students should be encouraged to click on different parts of the website, including the ZR IDE. Much of what they see will be unfamiliar to them—which is expected—but this experience will get them accustomed to using the website.

Bottle Rockets, Part 1 (90 minutes)

Materials Needed

For each student:

- Copy of **RM 1.1: Make Your Own Rocket!**

For each team of students:

- At least 1 2-Liter soda bottle (Coca-Cola Company brands work best)
- Construction paper and/or cardstock
- Duct tape to attach nose cone/fins to bottle (do not substitute with other tape)
- Scissors
- Computer with Internet connection

For group:

- Bottle launcher—information is provided on page 5-6 of this guide regarding purchasing or building a bottle rocket launcher.
- Launch retainer pin
- Air pump
- A few metal stakes
- Ample supply of water

Activity Description

Start this fun activity early in the ZR program to get students excited about what they will be doing over the next few weeks. Part 2 of this activity will occur in Week 5, when students will be challenged to apply what they have learned about physics to redesign their rocket, so teams will need to save their bottle rockets.

This activity can be dangerous—never stand over the bottle once it is in the launcher, and do not allow students to operate the pump without close supervision.

If you are uncomfortable doing this activity, you could use one of the following instead:

- 3 . . . 2 . . . 1 . . . PUFF! rocket activity from NASA
- Foam Rocket activity from NASA

❶ Explain that students are going to work in teams to develop their own rockets.

❷ Split the group into teams of two to four students, and give each student **Make Your Own Rocket! (RM 1.1).**

❸ Tell students to review the first section of the handout. Point out that the two items they will design and build are the nose and fins. Describe the purpose of these two components of the rockets and how their design can affect flight:

a) Since a tapered point is far more aerodynamic than the blunt end of the soda bottle, having a nose in the front of the rocket helps to streamline it. Cones can also be used to carry weights to shift the center of mass (the point at which the forward and rear weight are equal) of the rocket, which should be forward of the middle of the bottle.

b) Fins at the back of a rocket help to stabilize the craft, help it to maintain its course, and affect speed. Students may determine the number, shape, and size of the fins used; some of these differences may have a huge impact, while others may not. For example, size is extremely important—fins that are too small won't provide stability, while fins that are too large will cause too much drag and slow down the craft. The number of fins is less important than whether they are evenly spaced (which means that students will need to include two or more fins). And shape may not be much of a factor—that is, many shapes can work, so long as they are streamlined.

4. Show students the materials they have at their disposal for designing the nose cone and fins, and have teams discuss how they want to design the nose cone and fins for their rockets.

   Note: You may want to allow teams to conduct research on the Internet about nose cones and fins or just allow them to be creative and see what happens! If they conduct research, links to some design tips are provided in the student materials.

5. Before the launch, have students review the safety information on **Make Your Own Rocket! (RM 1.1)**. Ensure that all students are standing far back from the rocket at the time of the launch.

6. After each group has launched its rocket, discuss what occurred during each launch and introduce the kinematics, or mechanics of motion, concepts involved as well as the forces acting on the rockets:

   a) Did the rockets all survive launch and landing?

   b) Did all the rockets go straight up? If not, what do you think might have caused some rockets to move in a different direction?

   c) Did they all go the same distance in the air? At the same speed? For the same amount of time?

   d) What factors may have affected how the rockets flew?
   Answer: Forces that affect rockets in flight are weight (due to gravity), thrust (to overcome weight/gravity), and the aerodynamic forces: lift (from the nose cone and fins) and drag (due to friction with air molecules). The velocity and position of the rockets also play a part.

What's Going On?

Aerodynamic forces (lift and drag): Mechanical forces due to the interaction between an object and a fluid (gas—such as air—or liquid).

Force: A push or pull on an object.

Friction: A force that is caused by two objects rubbing together.

Gravity: A force that pulls everything toward the center of the earth.

Thrust: A force generated by propulsion to move an object through air or space.

For more details about rocket thrust and forces, see NASA's "Forces on a Rocket" and Background Information.

e) How do you think forces might have affected the rockets?

Answer: Gravity (and drag) both slowed the bottle as it was in motion (this is an example of Newton's First Law), while thrust (water being forced out) helped accelerate the bottle.

f) At what point(s) do you think the rockets were changing speed?

Answer: Rockets would have obvious accelerations at takeoff and at the apex (peak) of the flight, but also always had acceleration due to gravity (this is an example of Newton's Second Law).

g) As the rocket launched, what did you notice? Why do you think this occurred?

Note: Do not worry about introducing too much detail about the physics here—this is an opportunity to find out what students know or do not know about forces and motion.

Answer: As the rocket moved upward, water sprayed downward. (This demonstrates an equal reaction force to the rocket propelling upward and exemplifies Newton's Third Law.)

What's Going On?
Newton's Laws of Motion are three physical laws that relate forces acting on an object to the motion caused by forces. If you are unfamiliar with Newton's Laws, see Background Information. However, it is not important to introduce Newton's Laws to students at this point.

7. Explore some "what-if" scenarios related to the bottle rockets, including the following:

a) What do you think would happen if we carried cargo on the rocket? Would it make a difference whether the cargo was inside or outside the bottle?

b) What do you think would happen if we used a different kind of bottle? What if it were made of metal? Glass?

c) What do you think would happen if you could control the bottle's path at the apex—for example, by providing a burst of acceleration from a small booster on one of the sides?

8. Have teams save their bottle rockets, as they will revisit this activity in Week 5.

You Have Nothing to SPHERE . . . (30 min in class; 1 hour at home)

Activity Description

This activity is an introduction to the SPHERES satellites; students will create programs to maneuver the satellites aboard the ISS in the Final Competition at the end of the ZR Middle School Program, so they need to understand what the SPHERES are and how they work.

❶ Show students the YouTube video of SPHERES in action (0:41)

❷ Use the Getting to Know SPHERES PowerPoint presentation to share some facts about the SPHERES. As part of this presentation, you will do the following:

a) Remind students that the YouTube video of "SPHERES in action" described tests of formation flight performed with the SPHERES, then show a YouTube video of the Blue Angels (1:08) to demonstrate what *formations* are.

> *Note:* *There are some things in the slides that students will learn much more about during the program—such as coordinate systems.*

b) Explain that the SPHERES are a "testbed" to test new computer programs. A testbed allows scientists to try out many versions of their programs. The main purposes of SPHERES are:

- To test spacecraft docking: Two satellites come together and attach to each other, forming a new large spacecraft. Scientists investigate both how to do the docking itself, and also how to control the satellites after they have docked (since the new "big" spacecraft behaves differently than the original). This knowledge will help make it easier and cheaper to assemble new large spacecraft, like a new Space Station or a transport to Mars, in the future; it could also help to recover old spacecraft and reuse them

Materials Needed

For each student:

- Drinking straw (or K'nex pieces or equivalent)
- Pen or pencil
- Spherical object, such as a donut hole (plain cake donut is least sticky), Play-Doh, or clay
- A napkin (if choose donut hole above)
- Copy of Make Your Own SPHERES (RM 1.2)
- Piece of brightly-colored cardstock with image of SPHERES printed on it (image is available in the student materials)

For the group:

- Tape
- Scissors
- 8.5 x 11" sheets of paper
  Computer with Internet connection and projection equipment to show video and **Getting to Know SPHERES** PowerPoint presentation

or dispose of them safely (instead of having them float dead forever around Earth).

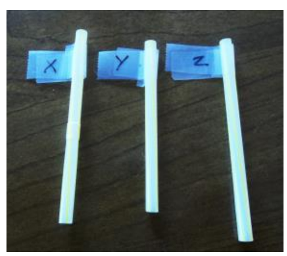
- To test how spacecraft can fly in formation: The satellites remain in a known geometry with respect to each other (forming a triangle, a line, etc.) and keep that shape as the whole formation moves (e.g., the triangle moving in a circular pattern). The formation can be *coarse* (just ensure that satellites never crash into each other—one idea is that future satellites will share features such as computers, antennas, etc.) or the satellites can fly with a lot of precision to create future space telescopes (e.g., the grandson of Hubble, about 15–20 years from now).

Note: *You might show students* *test sessions aboard the ISS*—Spiral Maneuver *(1:42) or* Decentralized Formation Control *(0:24)—which are available on* *MIT's Space Systems Laboratory website* *and in the student materials.*

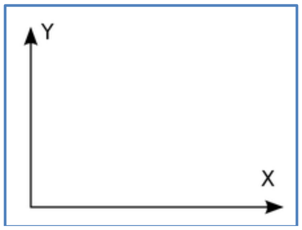
❸ Help students understand global and local coordinate systems by guiding them in the construction of a coordinate system model, and then playing a simple "Simon Says" activity:

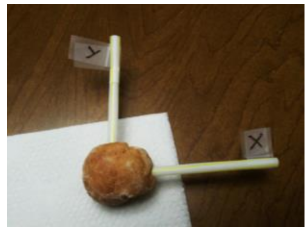
a) Construct a model

- Have each student: a) cut their straw in three parts, b) use tape to create a tag at the end of each piece of straw and c) label the straws X, Y, Z as shown.

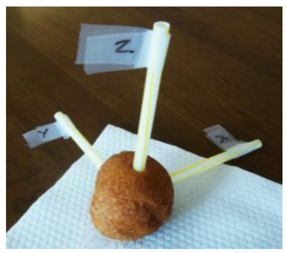


- Insert the X and Y straws into the donut hole (or substitute) to create standard X, Y coordinate axes to represent positive X and positive Y directions as shown. (Point out to students the negative X and negative Y directions)





- Insert the Z straw in the top of the donut hole to create X, Y, and Z coordinate axes with the straws pointing in the positive X, Y, and Z axis directions. (Point out to students the negative X, Y, and Z axis directions.)



b) Now ask the students to pretend the model (donut hole) is the International Space Station.

- Remind students that, as they learned in the Getting to Know SPHERES PowerPoint presentation about the global coordinate system on ISS, the positive Z-axis points toward the earth.
- Have the students turn the donut hole so that the positive Z direction points towards earth. Next, ask the students to rotate their model so that the Y-axis points to the right. Point out to students that this is the same orientation as the coordinate system they see in Free Mode in the ZR online simulation. This represents the global coordinate system for Zero Robotics.

c) Now ask the students to pretend that they are standing on the deck of the Japanese Experiment Module (JEM) in the ISS (where the SPHERES are used)

- Tell the students to assume the wall at the front of the room is the positive X direction and the right wall is the positive Y direction (As you say this tape 8.5 x 11" sheets of papers labelled +X and +Y on the walls).

- Ask students to point toward the positive Z direction. If they point down, congratulate them! They are beginning to understand the global coordinate system on ISS.

d) Next ask the students to pretend that the model they are holding in their hands represents a SPHERES floating in the ISS.

- Tell the students; "Now the X, Y, and Z axes on your model represent the local coordinate system of the SPHERES. This local coordinate system describes the orientation of the satellite and helps the satellite know which way it is facing. "
- Tell the students that the SPHERES points with its negative X face. Ask the students to point their SPHERES toward the positive Z direction in the room. (They should point the negative X axes toward the floor, positive X straw will be pointing up.)

e) Now tell the students to <u>pretend that **they** are the SPHERES and the +X face = front of the student</u>. Ask them "Which side do you, the SPHERE, point with?" (Answer: the negative X face which is their back) What happens if they have to point themselves in the +Z direction? (They would have to lay on the floor with their back (-X) facing the floor)

f) The students are now ready to play "Simon Says" and practice using the global and local coordinate systems.

- Play Simon Says, asking the students to point the SPHERES (themselves) toward the +X, +Y, +Z, –X, -Y, or -Z directions. Following are several examples:
  i. Simon says, "Point the SPHERES toward –X".
  ii. Simon says, "Point the SPHERES toward positive Y".

   iii. "Point the SPHERES toward negative Y". (Note: no "Simon Says" so students should do nothing)

- Add changes in position. Tell the students that the SPHERES move in any commanded direction even if they are not pointing in that direction.
   i. Give the following example: Simon says, "Point in the +Y direction" (students should turn their backs to face +Y) "and move 3 steps in the -X direction and 4 steps in the +Y direction." (Students should move while keeping their back facing the+Y direction)"
- Invite students to take turns being Simon
- Change the global coordinate system by changing which wall represents X and Y. Tell the students that now the back wall is the positive Y direction. Ask students to point to the wall that would be in the positive X direction on the ISS. (They can use their models (with positive Z pointing down) to find the positive X direction by pointing the positive Y straw toward the back of the room and looking to see which direction the X straw points) Move the pieces of paper labelled X and Y to their new locations and continue playing Simon says.

Try Your Hand at Programming (90 minutes)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Table on newsprint or chart paper, prepared according to the directions in Before You Teach
- Computer with Internet connection and projection equipment to show the **Getting to Know the ZR IDE** and **Mystery Coordinate Grid** PowerPoint presentations

Optional Activity: Coordinate Hunt

For each student:
- Copy of Coordinate Clue Solutions (RM 1.3)

For the group:
- Copy of **Coordinate Hunt Pieces**, prepared according to the directions in Before You Teach
- Computer with Internet connection and projection equipment to show the **Coordinate Hunt** PowerPoint presentation

Optional Activity: Battleship

For each pair of students:
- Either the board game "Battleship" or some graphing paper

Activity Description

Now that students have been introduced to the SPHERES and the ZR IDE, they can begin to do some programming. It is really important that students begin to "play" in the ZR IDE so that they can feel comfortable with what they are doing, even if they aren't always sure *what* they are doing—and even if you aren't!

1. Show the Mystery Coordinate Grid PowerPoint presentations. Explain that to find a point on this grid, we need to find out how far along it is on both the *x* and the *y* lines. These lines, labeled in the graph, are the measuring sticks we will use to locate our points. We refer to them as the *x*-axis and *y*-axis.

   a) Explain that the *coordinates* of a point describe its exact location on the graph. The measurement along the *x*-axis is called the *x* coordinate. For the *y*-axis, it's the *y* coordinate.

   b) Coordinate graphing sounds very complex, but it is actually just a visual method for showing relationships between numbers. The relationships are shown on a coordinate grid like this one.

   c) Display the table you prepared and explain that we usually pair the *x* and *y* coordinates. For example, we would use the pair (1,3) for point A on the grid, because this point is +1 unit far along the *x*-axis and +3 units along the *y*-axis. Write "(1,3)" on the table as the pair for point A, and ask students what the *x* and *y* coordinates are for point B. Record their response.

d) Have students locate the coordinates for the remaining points. Fill in the table with their responses (answers right).

Point	(*x* coordinate, *y* coordinate)
A	(1,3)
B	(–1,–2)
C	(3,1)
D	(–2,2)
E	(2,–1)
F	(–1,–4)

2. Next, have students complete the *Getting to Know the ZR IDE* tutorial, which gives them a chance to move one of the SPHERES. You might walk them through the Getting to Know the ZR IDE PowerPoint presentation slide by slide so that they don't have to switch between the ZR IDE and the tutorial instructions.

3. Ask students to share what the experience of completing the tutorial was like for them. Was it hard or easy? Was anything introduced that they didn't understand? Was it fun to move a SPHERE?

> *Note: While the SPHERES satellites move in three-dimensions, the game that middle school students play only has two dimensions.*

4. Some students may mention that they were confused by or did not understand the references to *x*-, *y*-, and *z*-axes. Even if no one mentions axes, go over this concept with the class, as it is an important one for students to understand:

   a) Ask for a volunteer to explain what an "axis" is and to guess what axes have to do with the SPHERES satellites.

   b) Refer students back to the Getting to Know SPHERES presentation, which showed the structure and movement of the SPHERES satellites:

   - The orientation of the satellite (the local coordinate system) is described by a face (*x*, *y*, or *z*) of the satellite; there are six orientations possible (+/–*x*, +/–*y*, and +/–*z*).
   - The position of satellite within the global coordinate system is indicated by a set of *coordinates* (*x*, *y*, *z*); for example, the satellite could be located at (1,–1,0).

What's Going On?

In this tutorial, students will actually use the ZR IDE. Building off of what they have just learned about the coordinate system and locating points in a grid, they will write a short program to move a SPHERES satellite in a straight line (i.e., along the y axis).

What's Going On?
Axis: The basic lines of measurement used to locate points. The first three dimensions typically have axis labeled as x, y, and z. *Coordinates*: The measurements of a point's distance along each axis. The number of coordinates required is the same as the dimension of the space. For more information about graphing, see *Background Information*.

c) Explain to students that in order for us to tell the SPHERES what direction to face and what direction to move in, we need to have some way to orient the satellites. We do this by using a grid and coordinates. To learn more about this, we're going to simplify the example and just look at a points on a flat grid rather than in space—in other words, in two dimensions rather than three.

Optional Activity 1: Coordinate Hunt

This Coordinate Hunt reinforces the concepts of dimensions, graphing, and coordinates. It also gets students up and moving around. Some of the problems in this activity expect students to have had experience with pre- algebra (or Algebra 1). Create the teams accordingly. For this activity the Coordinate Hunt Pieces should be printed double-sided (the front and back coordinates correspond) and clues cut out along the lines. The student worksheet for this activity is RM 1.3 and the solutions are available in Appendix 4.

1. Divide the group into four teams and give each student a copy of RM 1.3: Coordinate Clue Solutions.
2. Show students the scenario (first slide) in the Coordinate Hunt PowerPoint presentation.
3. Explain that each team will get an initial clue that the team needs to solve. The solution will direct them to the next clue (set of coordinates). Teams will continue to solve each clue and will follow the clues to a last clue (set of coordinates) that will help them solve the "meta puzzle" that was asked on the initial clue card.
4. Distribute one initial clue card from Coordinate Hunt Pieces to each team. Explain that the initial clues will send each team to a different point so that students are not running into each other. Discuss the first clues as a group and show slide 2 in the Coordinate Hunt presentation before sending the teams off to hunt for their next clues.

5. Have students complete the Coordinate Hunt, keeping track of their solutions using **RM 1.3: Coordinate Clue Solutions.**

6. Debrief the activity, showing the final answer (slide 3 in the [Coordinate Hunt]() presentation).

> **Note:** *Solutions for each clue are provided in **Appendix 4: Coordinate Hunt Solutions.***
>
> *Try to keep teams as small as possible; the bigger the team, the greater chance that one or two students will dominate the clue solving and leave the others bored or uninvolved in the activity. Encourage team members to work together to solve the clues. Students may want to carry around their clues so that they can revisit or check their work if they get crossed up and/or need information from previous clues to solve their current clue.*

Optional Activity 2: Battleship

Have students pair up to play a game of "Battleship," either using the board game or doing it on graphing paper (see [student materials]() for an example of how to play the game using paper).

Get in the Game (75 minutes)

Activity Description

In this activity, students are introduced to the ZR game, where they work in teams to program a player to achieve the goal set forth in the game. Teams will strategize the best way to move a SPHERES satellite around and use all resources at their disposal in order to earn the highest possible score. After an Intramural Competition, teams will work together as a group to develop the best strategy they can, to compete against all other sites in their region. The Regional Competition determines the regional 1st, 2nd, and 3rd place champions. After the Regional Competition, all

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment to show the [ZR Game Introduction video]() and the [PowerPoint presentation]()

teams in a region will collaborate to improve on their regional winner's code before it is submitted for the National Competition. During the National Competition teams will have a chance to see their region's final code in action on the ISS!

1. Engage students in learning about the problem or issue they will address with their strategy by having them look over the "Get in the Game" section of the student materials and watch the ZR Game Introduction video.

2. Explain the various competitions that students may engage in—intramural, regional, and national:
   a) Reiterate that the goal is to solve the problem at hand. Only the strongest codes will be run on the ISS. Therefore after the practice Regional Competition in Week 3, students will work together to help refine or improve their codes into a final (and hopefully, winning) entry in the Regional Competition.
   b) Despite which team wins the Regional Competition, all teams in a region will have an opportunity to collaborate to improve their regional winner's code before it advances to the National Competition. Teams will join together during the National Competition, conducted live aboard the ISS, to cheer on their region's code.

3. Show the Game Overview PowerPoint presentation. Make sure that students understand the game operation and rules by posing the following questions to the group:
   a) What have you been asked to do?
   b) What is the problem that you must solve?
   c) How have you been asked to solve this problem?
   d) What skills and knowledge do you think you will need to solve this problem?
   e) Is this similar to anything you have done before? How so?

f) How do you feel about the work you will be doing? What, if anything, are you anxious about? Excited about?

> ***Note:*** *You can divide the group into teams or allow them to choose their own teams, but try to ensure that each team includes at least one student who appears to be strong at (or more interested in) coding.*
>
> *While it may be difficult to gauge students' skills at this point, you could ask the group who thinks that coding is interesting or exciting and who would like to learn more; given how challenging—and sometimes tedious—coding work can be, a student's interest can go a long way toward helping him or her persist!*

4. Divide the group into teams of three or four to work together for the first round of competition in Week 3 (Intramural Competition). Explain that these teams will work together to complete the programming for their game—have each team come up with a name. Encourage them to be creative about this!

5. Have each team develop a poster that represents the team's understanding of the game rules and operations (that is, how the game works and what the team must do to win).

What's the Logic? (45 minutes)

Materials Needed

For each student:
- Copies of RM 1.4: Making a Sandwich

For the group:
- 1 jar of peanut butter or other spread (see Note below)
- 1 jar of jelly
- Additional containers to distribute jelly and peanut butter/spread across teams of 4 students
- Ball or similar object to represent the satellite

For each team:
- 2 slices of nut-free bread
- 2 butter or plastic knives
- Lined writing paper
- Pencils
- *Optional:* A small collection of blocks or similar objects

Activity Description

This activity serves as an introduction to computer science and programming, the topic of the following activity. It is designed to help students understand the concept of *logic* in computer science and the necessity for thoroughness while programming; these ideas are introduced through the often strange results of *literalism*. Each of these terms will be described in more detail later. The overarching theme is that computers do exactly what they are told and nothing more. The ability to read between the lines and determine what was meant or implied, rather than what was explicitly said, is a skill that computers lack. Additionally, students are introduced to the concept of *debugging* through iterative attempts to program a "computer" to make a peanut butter and jelly sandwich.

Due to food allergies, it is highly recommended that alternate spreads be used in place of peanut butter. Some options include soy butter, spreadable butter or margarine, or whipped cream cheese.

1. Divide the group into teams. (Ideally, each team should have two "writers" and two "performers.") Distribute **Making a Sandwich (RM 1.4)** to each student.

2. Tell students that in this activity they will develop a set of instructions—a "program"—to tell a "computer" (you) to make a peanut butter and jelly sandwich.

   Explain that teams will each develop their instructions through a process of trial and error, which is how they can "debug" their program. Go over the process described on RM 1.3:

   a) Each team will collaboratively write instructions for making a peanut butter and jelly sandwich.

b) The two “performers” will test the instructions, while the “writers” record successes and challenges.

> *Note: Slide 12 in the Getting Into CS PowerPoint presentation shows a successful set of instructions for making the sandwich.*

c) Teams will revise their instructions based on the test, and the performers will repeat the challenge one more time using the revised instructions. Teams will then finalize their instructions.

3. Assign each team to a station. Set a time limit of 10–15 minutes for teams to test and finalize their instructions. Have teams conduct the activity.

4. Collect each team’s final instructions. Tell the class that you are a computer that is going to follow each team’s instructions as a computer would. Proceed by interpreting the instructions for each team in the most literal manner possible (which is likely to result in errors).

> *Note: For example:*
>
> 1. *Take a slice of bread*
> 2. *Put peanut butter on the slice*
> 3. *Take a second slice of bread*
> 4. *Put jelly on that slice*
> 5. *Press the slices of bread together*
>
> *. . . would result in you taking a slice of bread, putting the jar of peanut butter on top of the slice, taking a second slice of bread, putting the jar of jelly on top of that slice, then picking up both slices of bread and pushing them together*

What’s Going On?

To write effective programs, you need to think like a computer scientist. The most important tool that computer scientists have is not in fact computer programs (or their ability to write programs) but rather their ability to apply logic and reasoning to solve problems. Problem solving is the process of defining the question or challenge that you want to address, finding the solution, and then ‘expressing’ it through spoken or written words or images. Programmers usually have a problem to solve and they must use logic to find a solution to the problem and then ‘write’ a program that implements their solution.

What’s Going On?

Debugging—figuring out why your solution did not work—is an extremely important part of computer science. Sometimes, the reason is simple; for example, maybe you spelled something wrong. Usually, however, the mistake is not so easy to fix. A computer does what you tell it to do—no more, no less. Working backward is important for debugging. You know what *should have* happened and what *did* happen. You can work backward to find out why your program produced the wrong result.

5. Have the entire group debug—identify errors—in each team's program and work collectively to create a master list of instructions.

6. After successfully making a sandwich, invite students to make more sandwiches and enjoy lunch.

> ***Note:*** *You may replace the sandwich activity with this version of the activity if necessary.*

7. Repeat this activity with a space/satellite theme:

   a) Tell the group that they will now repeat this activity as a group, with one person acting as the performer. This person must use a computer program in his or her space laboratory to move a satellite in space.

   b) Show students the object you chose to represent the satellite. You will move according to the instructions of the computer program—that is, what the performer tells you to do.

   c) The performer cannot see you or the satellite and must follow the feedback from the rest of the group about its position.

   d) Throughout the activity, reinforce the concept of literalism, linear sets of instructions, logic, and the debugging process.

8. Before moving on to the next activity, tie this activity back to the ZR game. Ask students what challenges they anticipate, based on what they've done so far, in strategizing and writing a program for their ZR players.

   Possible answers: *Students may note possible challenges in figuring out the logic of how the SPHERES will act, anticipating scenarios that might come up in the game (for example, if they plan on doing something a certain way but another team's strategy interferes), and anticipating all of the needed information for their player.*

What's Going On?

This activity is intended to expose students to the idea that computers follow instructions very precisely, which can be frustrating at times. It also raises an issue surrounding "instruction sets"—that is, whether it's better to have a large set of simple instructions or a smaller but more complex set.

Getting into CS (30 minutes)

Activity Description

Now that students have had a chance to experience how computers operate, you can introduce specific programming concepts, including *data types, variables, loops,* and *conditional statements*.

1. Introduce the concepts of *data types* and *variables* in computer science by showing students the Getting into CS PowerPoint presentation.

2. Relate these concepts to students' task from the previous exercise: making one sandwich. Show slide 12 in the Getting Into CS presentation, which contains the final instructions for making a sandwich. Ask students to think about an instruction they can add to tell the "performer" or a computer how to make 100 sandwiches in a cafeteria, which, they can assume, has unlimited supplies of the materials needed.

3. Ask some volunteers to share their solutions. Discuss any flaws in these solutions with the group.

   Possible answers: Students might suggest the following:

   - *Do this 99 more or 100 times*

     The flaw here is that the computer or a performer isn't necessarily going to understand what "this" means. Does it refer to the last step, the entire set of steps, or something in between?

   - *Repeat steps a–g until you've made enough sandwiches*

     The flaw here is that the computer or a performer does not know when to stop and could end up making sandwiches forever (remember, you have told students to assume that the cafeteria has an unlimited supply of the materials!).

Materials Needed

For the group:

- Computer with Internet connection and projection equipment to show the Getting Into CS PowerPoint presentation

What's Going On?

Data types

Computer programs are essentially a set of instructions that act on data to produce certain results. As in math, computer programmers uses specific *data types*—classifications of various kinds of data, i.e., floating-point, integer, or Boolean—to represent mathematical information on a computer. The data type states the possible values for that type, the operations that can be done on that type, and the way that values of that type are stored.

What's Going On?

Variables

When you are writing a computer program, you often want the program to be able to "remember" a value. You can do this by assigning the value to a named variable. Each variable has a name and contains a value. Variables are categorized by the types of data they store. You must tell the computer what type of data you will be putting into the variable. Attempting to put data of the wrong type into a variable will cause an error.

For more details on data types and variables, see *Background Information*.

4. Work with the class to update their computer program to make 100 sandwiches and to debug their new program.

   Answer: The final solution should be something like: Repeat steps a–h 99 more times or Repeat steps a–h until the number of sandwiches equals 100.

5. Point out to students that by repeating the instructions, they are essentially looping through them over and over again. However, they cannot do this forever or they would end up with many more than 100 sandwiches. By specifying "99 more times" or "until the number of sandwiches equals 100," they are setting a limit—which is essentially a *conditional statement*.

What's Going On?

Loops

One of the best things about computers is that they can process huge amounts of information in very short periods of time. *Loops* help make computers fast and efficient by allowing them to carry out the same instructions over and over, but only requiring the programmer to write the instructions once. In this way, loops simplify what could be very complicated programs. There are several types of loops, but in ZR we only use the **for** *loop*.

Conditional Statements

Conditional statements describe the behavior of the computer in response to various situations. A computer will respond differently to different commands. (Humans employ similar logic—if it's snowing outside, we wear a coat and boots; if it's sunny and warm, we wear a T-shirt.) The computer makes decisions based on what's happening in a program. However, since computers are not intelligent, they must be told exactly what to do in every situation.

For more details on loops and conditional statements, see *Background Information*.

6. Have students consider a conditional statement in the cafeteria scenario: Say that the cafeteria serves only PB&J sandwiches except on Fridays, when it serves both sandwiches and pizza. Since most students would rather eat pizza, the cafeteria should make fewer sandwiches on that day. How does this conditional statement change their computer program?

   Possible answer:

   a. Take a slice of bread
   b. Open the jar of peanut butter by twisting the lid counter clockwise
   c. Pick up a knife by the handle
   d. Insert the knife into the jar of peanut butter
   e. Pull the knife out of the jar of peanut butter and run it across the slice of bread
   f. Set down the knife
   g. Take a second slice of bread
   h. Repeat steps a–f with the second slice of bread and the jar of jelly
   i. Press the two slices of bread together such that the peanut butter and jelly meet
   j. *Repeat steps a–i 99 more times unless today is Friday*
   k. *If today is Friday, repeat steps a–i 29 more times*

Note: *Students can make up a number of sandwiches that should be prepared.*

Getting Your SPHERE Lined Up (30 minutes)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment to show the **Sample Arrays** and **Introduction to Arrays and the setPositionTarget Function** PowerPoint presentations

What's Going On?

An array is a data type that stores multiple pieces of data in one place. Essentially, an array is a list of other data types. Students will use arrays extensively while programming the SPHERES satellites. In this tutorial, students will create a program that uses arrays and a SPHERES control function, called setPositionTarget, to move the satellite to a target position.

Activity Description

In this activity, students are introduced to the concept of *arrays*, a data type that stores multiple pieces of data in one place. Students complete a tutorial in which they program the SPHERES to move to a specific point in the field (which happens to be a three-dimensional space). They create a project in the ZR IDE and then view the C code behind what they create (that is, they get to use both the graphical editor and the text editor) while learning about arrays.

❶ Explain to students that there can be different kinds of arrays. Tell them that an array is essentially a list of other data types; it can hold as many pieces of data as needed. Show them the examples in the Sample Arrays PowerPoint presentation.

❷ Have students complete the ZR tutorial, *Introduction to Arrays and the setPositionTarget Function,* using the PowerPoint presentation. (You may want to have them follow the steps one by one as you project the slides.)

Logic Competition (45 minutes)

Activity Description

This activity reinforces several computer science concepts, the first of which is *explicitness and thoroughness in programming*. Although this concept is explored in the sandwich activity, it is important and bears reinforcement. It is important to remind students that computers are not smart—they only do what their programmers tell them to do. This activity also reinforces the idea of *debugging* (also introduced in the sandwich activity), of seeing the output of your program, and changing your program to give the output you desire. Conditional statements and conditional logic are further explained in this activity.

In this activity, the Writers will be able to see how the Doers interpret their instructions, and will then be given an opportunity to "debug" their instructions to produce a better output. The Writers will have to describe the actions needed to produce one of three possible outcomes, and the different courses of action that the Doers must take to figure out what to do (conditional logic).

1. Distribute **Writers-Doers: What You Need to Know (RM 1.5).** Pair students up, assigning one to be the Writer and the other to be the Doer for this team competition.

2. Distribute **What to Do (RM 1.6)** to the Doers, and tell them they can do one of the activities while they wait. Have the Doers leave the room.

3. Show the Writers the three structures. Tell the Writers they have 15 minutes to describe in writing for the Doer how to build one of three structures. Their instructions should also help the Doer determine which of the 3 structures they are being asked to build based on the

Materials Needed

For each student:
- Copy of RM 1.5: Writers-Doers: What You Need to Know

For each pair of students:
- Copy of RM 1.6: What to Do
- Paper
- Pen or pencil
- A set of materials to build one of the three structures

For the group:
- Three structures made from Legos or building kits, such as the following:
- Building kits with connectors
- Marble Runs
- Keva structures

materials they are given. Go over the following ground rules:

a) The Writer cannot touch any of the structures.

b) The Writer may use letters, numbers, and symbols, but no drawings.

c) The Writer must define any abbreviations used.

d) The Writer does not need to write in complete sentences.

4. Give the Writers time to complete their task.

5. Bring Doers back to the room. Give each Doer a set of materials, and have Writers give them their written instructions. Remind students that the only form of communication between the Writer and the Doer is the Writer's instructions—in other words, no talking! Allow 10 minutes for Doers to create one of the structures.

6. Give Writers five minutes to issue additional written instructions to help Doers "debug" their structures.

7. Give Doers another five minutes to try to "debug" their structure.

> ***Notes:*** *Doers should get more parts than they need in their assembly kits. Encourage the Writers to look at the structures from different angles and points of view. Suggest that students consider the qualities of the individual components themselves, the spatial relationships between the components, and the directions in which the components are facing. Also suggest that students consider the relationships of the components in both an absolute sense (where they are in relation to the viewer) and a local sense (where they are in relation to each other).*

8. Score the teams based on the overall number of correct connections. If there is a tie, the winner will be the Doer who completed the structure faster.

9. Before moving on, reiterate to students that the activities they have been working on will help them to

remember that computers can only do what they are told to do—no more and no less. This lesson will be extremely important when they begin programming their player for the ZR game!

Learning All the Right Moves (60 minutes)

Activity Description

Earlier, students learned about arrays in the *Introduction to Arrays and the setPositionTarget Function* tutorial. In this activity, they delve more deeply into learning about arrays, which are used extensively in the ZR IDE to control the SPHERES satellites.

1. Review what students did in the *Introduction to Arrays and the setPositionTarget Function* tutorial, when they programmed a SPHERES satellite to move to a specific point:
   a) They used the ZR IDE to declare an array and assign values to an array
   b) They used the setPositionTarget function to move the SPHERES to a point
   c) They created a project in the ZR IDE using the graphical editor and then looked at the program translated to C code

2. Tell students that they will now learn how to use arrays to program the SPHERES to rotate and point in a specific direction.
   a) Show students the Rotation in 2-D PowerPoint presentation

3. Have the students complete the following two tutorials:
   a) *More Simple Arrays and the setAttitudeTarget Function*
   b) *More Simple Arrays—Another Way to Initialize Variables*.

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment to show:
- **Rotation in 2-D** PowerPoint presentation
- PowerPoint presentations for the *More Simple Arrays and the setAttitudeTarget Function* and *More Simple Arrays—Another Way to Initialize Variables* tutorials

What's Going On?

Arrays are very important when writing programs to manipulate the SPHERES and are used to both set the position of each satellite and to move and rotate the satellites (as well as specific SPHERES controls functions). In *More Simple Arrays and the setAttitudeTarget Function*, students use a SPHERES control function called setAttitudeTarget to rotate the satellite to point in a specific direction. In *More Simple Arrays—Another Way to Initialize Variables*, they learn another way to initialize variables and then use the setAttitudeTarget function again to rotate the satellite in multiple directions.

❹ The following is the solution to the challenge in the tutorial *More Simple Arrays—Another Way to Initialize Variables*:

1) point to +X (pointposx =(1,0,0))

2) point to –X (already in front) (pointnegx =(–1,0,0))

3) point to –Y (pointnegy =(0,–1,0))

4) point to +Y (pointposy =(0,1,0))

5) point to +Z (pointposz =(0,0,1))

6) point to –Z (pointnegz =(0,0,–1)

Note that the initial position of the satellite must be configured correctly in the simulation settings box as directed in the tutorial, and the variables must be correctly initialized. Refer to the figure below.

Reflect and Assess

At the end of the week, provide students with opportunities to reflect on what they have learned so far using the Questions for Reflection on page 6; you might then want to incorporate some of the following reflection and assessment items.

<table>
<tr><th>Computers Are Only as Smart as You Make Them</th><th>Jeopardy!</th></tr>
<tr><td>Lead a class discussion by asking students, “Just how smart are computers?” Tell them that they must give supporting evidence for any claims they make. You might also ask some of the following questions to further probe students’ thought processes:

1. What does it mean to be “smart”? What about “intelligence”? Can something that is not alive have either?
2. Can you name some everyday objects that “contain” computers? Do the computers inside them make them smarter? (The point here is to determine if students can name objects such as phones or cars and understand that the satellites they are programming are similar to the way these devices are programmed.)
3. Does following instructions require intelligence? Does the creating or writing of instructions require intelligence? Why or why not?

Why might we depend on computers so much in space? What functions can they serve? Why not just use people, if people need to constantly monitor what the computers are doing?</td><td>Develop a class game of Jeopardy! using the materials provided in the Resources section of the ZR website or the information included in this guide. You might want to use the following categories:

1. Space (call it “Lost in Space”)
2. The International Space Station (call it “A Floating Space Laboratory”)
3. SPHERES (call it “Pretty Colored Spheres”)
4. Logic (call it “How Computers Think” and present statements that students need to determine the purpose of; for example, a clue might be “You must first pick up the jar, gripping the sides tightly with one hand. Then grasp the cap of the jar with your other hand and twist the cap until it ‘pops’.” and the correct answer would be “How do you open a jar?”)
5. Programming (call it “Telling Computers What to Do”)</td></tr>
<tr><th>What Do You Know Now?</th><th>Achieving Our Goals</th></tr>
<tr><td>Revisit the K-W-L and add to it based on what students have learned and wonder about at this point.</td><td>Display the student learning outcomes for Week 1, which students can see in the Week 1 student materials. Ask students to reflect on how well they believe they have achieved these outcomes and where they need more support or experience. In particular, ask students to reflect on how it feels to work together in teams to solve problems.</td></tr>
</table>

Do You Know More than a 12th Grader?

Quiz your students on some of the terms and concepts they learned about this week. For example:

1. What is the reason to use a *loop* in computer programming? (**Answer:** To repeat an action multiple times.)
2. How is a computer program like a set of cooking directions? (**Answer:** A computer program is the set of instructions that a computer follows. A computer program can include different instructions for different conditions, in much the same way that a recipe might include tips for adapting the recipe if you are cooking at a high altitude.)
3. What is a *coordinate*? (**Answer:** A coordinate is a number that describes position.)
4. How do you think you will use coordinates in the ZR game? (**Answer:** To find or change the position of the SPHERES satellites.)
5. People often dress up on Halloween and wear regular clothes the rest of the year. However, some people may wear costumes for other occasions. How would you write a set of conditional statements to tell a computer what kind of clothes to put on a person depending on the date or event?

   (**Answer:**

   ```
   if (it is Halloween) {
   wearHalloweenCostume;
   }
   else if(you are in the school play) {
   wearPlayCostume;
   }
   else {
   wearNormalClothes;
   }
   ```
6. Why is the research with SPHERES so important for space exploration? (**Answer:** Since SPHERES are used to test instructions for spaceships to meet up with other equipment and to dock, they allow us to make sure that equipment will work correctly when it is not being operated by people. Since we are interested in exploring space farther and farther from Earth, and it isn't necessarily safe or even yet possible for people to go on these trips, we need to rely on computers. Therefore, we have to make sure that the instructions used by the ships will allow them to perform the necessary tasks.)

Week 2:

Developing a Strategy

Overview

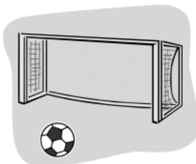
During the second week of this program, students have a chance to participate in a "Field Day" where they simulate the ZR game and learn how to effectively strategize. This helps them begin to develop a program for a player, which they will get a chance to compete with in Week 3 and see how it stacks up against other players in their region. In addition, they learn much more about the math and physics involved in maneuvering the SPHERES satellites and in programming. ‡

Getting Started

Before Week 2, review the information below and gather the materials needed.

‡ Note: Depending on your students' skill levels, you may need to keep the focus on programming and do fewer math and physics concepts during this week. However, remember to continue to keep programming sessions to approximately one hour per day and provide other activities to break up programming time and give students an opportunity to move around and enjoy their out-of-school time!

Goals



Students understand the basic math and physics concepts underlying SPHERES and the ZR game.

Student Learning Outcomes

By the end of this week, students should do the following:

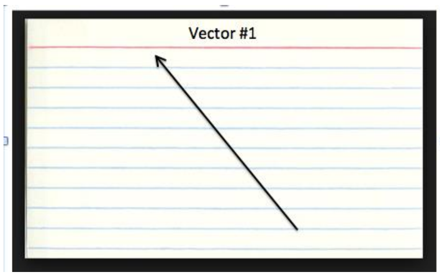
- Understand the basic math and physics concepts involved in maneuvering the SPHERES satellites, including:
    - Forces
    - Motion
    - Newton's Laws
    - Order of operations
    - Grids and graphing
    - Cartesian coordinates
    - Dimensions
    - Vectors
- Understand the basic elements of the ZR game and programming the SPHERES
- Understand how the ZR game controls can be used in a computer program to maneuver the SPHERES satellites
- Be able to complete advanced programming tasks in the ZR IDE using the instructions provided
- Be able to work in a team to solve problems related to activities and their game strategy and coding

Before You Teach

- Copy the RMs needed for the activities—see Materials Needed below.
- Review the suggested flow of activities below and estimate what activities will be scheduled for each day based on the time you have.
- Review the Background Information (beginning on page 124) for the concepts used this week: forces, functions, graphing, motion, Newton's Laws, and vectors.
- For the Field Day activity, download and read through the Acting Out the Game—Instructor's Guide. Details and a schedule for the Field Day field trip will be provided by your regional State After-School Network (SAN). Check with your

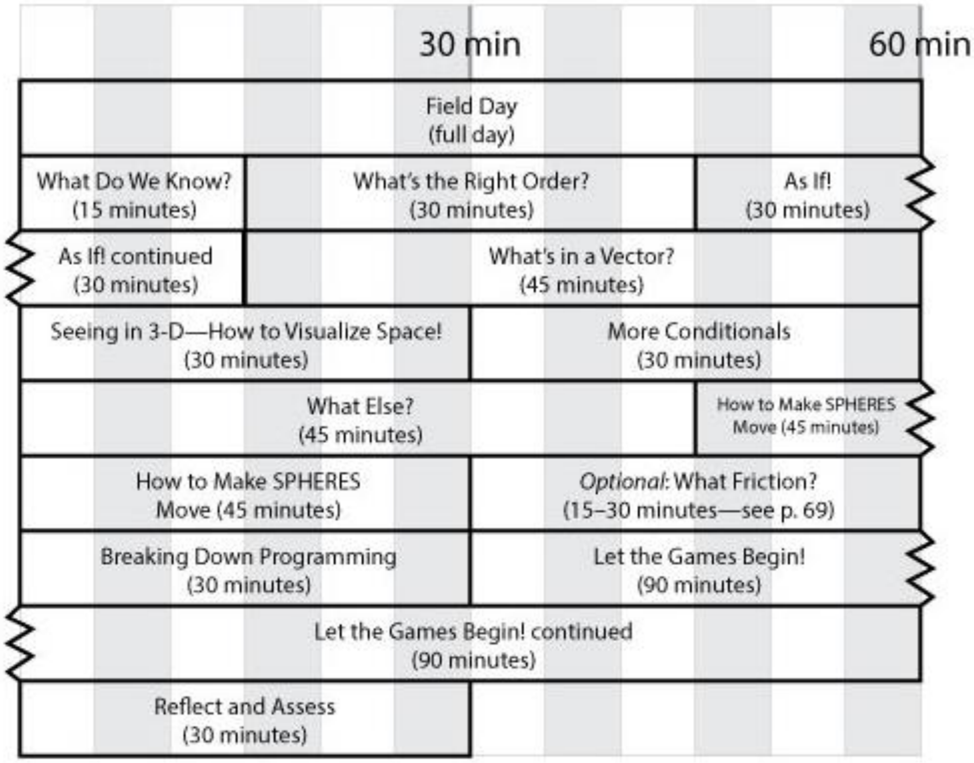
SAN regarding additional activities and materials needed.

- For the optional "What Friction?" activity, prepare as follows:
    - Unroll poster paper onto a table and place a heavy object on each of the four corners. (Make sure that the sheet is completely flat. Wrinkles will cause the blocks of dry ice to stop moving.)
    - Gather the materials needed for the obstacle courses (see Appendix 5).
- For the Vector Hunt activity:
    - Label nine brightly colored 3 × 5 index cards with the numbers 1 through 9
    - Draw a "vector" on each index card 9 with the magnitude and direction (relative to the horizontal lines on the card), as shown:
        - 1 Magnitude: 3 inches; Direction: 120 degrees counterclockwise

          Example: 

        - 2 Magnitude: 2.25 inches; Direction: 50 degrees counterclockwise
        - 3 Magnitude: 2.25 inches; Direction: 130 degrees counterclockwise
        - 4 Magnitude: 2.5 inches; Direction: 90 degrees clockwise
        - 5 Magnitude: 2.5inches; Direction: 90 degrees clockwise
        - 6 Magnitude: 4 inches; Direction: 180 degrees counterclockwise
        - 7 Magnitude: 2.75 inches; Direction: 70 degrees counterclockwise
        - 8 Magnitude: 2.75 inches; Direction: 70 degrees counterclockwise
        - 9 Magnitude: 3 inches; Direction: 180 degrees counterclockwise
    - On the day of the activity, place the index cards with the vectors around the classroom (or outside).
    - Draw a tenth vector on a large piece of paper or on white board (of a magnitude and direction of your choosing) to use at the front of the classroom or outdoor area, for demonstration.
- For "How to Make SPHERES Move," cut enough strings (10–15 ft. long) for the thruster balloons so that every pair will have one.
    - Prepare and set out the materials needed for each session.
    - Determine which items from the "Reflect and Assess" section on p. 75 you will use.

Suggested Flow/Schedule of Activities



Activities

Field Day! (1 full day)

The Field Day field trip is organized in conjunction with your regional SAN. All ZR Middle School Programs from one region participate in Field Day Activities together. Encourage team spirit and remind students that the other teams they meet during Field Day are the teams they will be competing against in the Regional Competition.

The primary activities for Field Day are listed below:

- Acting Out the Game activity
  - The goal of this activity is to help students understand the middle school ZR game better by acting it out themselves. It will also help them develop a strategy for coding their players in the competitions (additional details for this activity are provided below). You will be involved in running this activity for your group of students.
- Tour of facility
  - Field Day includes a tour of a college campus or NASA/space/science institution. This is an exciting opportunity for students to visit a facility where they can meet with students and STEM professionals in the space, robotics, or other fields, as well as view and interact with science and space-related exhibits.
- Lunch (Pizza)
- Dry Ice activity (optional): If included, the dry ice activity (see page 69), will make for an informative and entertaining activity on the concept of friction.
- Other activities can be added if desired.

Acting Out the Game Activity

Field Day should occur early in Week 2 so that students can start thinking about ZR game strategy before they begin planning their own team's strategy for the Intramural Competition at the end of the week.

1. Explain the rules and object of the game.
2. Go over what will happen during the game:
   a) Two SPHERES will run at a time.
   b) Even if their SPHERE isn't in play, the other teams should watch the game, as they may pick up some helpful ideas.
   c) Sites will run their players against one another and then all come together to discuss what they learned from the simulations.
3. Have teams plan their overall strategy and conduct the competition.
4. Bring the whole group together to discuss what they learned from each round of the game.
5. Complete other Field Day activities such as tour of college campus or space/science center and optional dry ice activities.

What's Going On?

The ZR game changes each year to present a different challenge to students. A guide for this year's Field Day is available in the student materials.

What Do We Know? (15 minutes)

1. Review some of the concepts that were covered in Week 1, using the questions below (organized by overall topic).

   SPHERES

   - What does NASA stand for? (**Answer:** National Aeronautics and Space Administration)
   - What are SPHERES, and what does their name stand for? (**Answer:** They are satellites. The acronym stands for Synchronized Position Hold, Engage, Reorient, Experimental Satellites.)
   - What is the SPHERES game about? What "problem" is their program supposed to address?

(**Answer:** This will vary, depending on the game for the year.)

- How would you describe one of the short videos you saw in Week 1? (**Answer:** This will depend on which videos the students watched in Week 1. Any video is acceptable here as an answer, as long as students can reasonably recollect and describe what the video was about.)

Computer Science and Programming

- How do computers know what to do? (**Answer:** Programs tell computers what to do.)
- What is a program? (**Answer:** A *program* is a set of instructions for a computer.)
- What are some of the basic elements of a program? (**Answer:** Variables, conditional statements, loops, and functions are all acceptable answers.)
- What does it mean to debug a program? *You might need to remind them of the sandwich or writer-doer activity.* (**Answer:** Debugging is the process of figuring out why a program did not work and determining how to fix those errors.)
- What is the ZR IDE and why will you use it in the ZR game? (**Answer:** The ZR IDE is the part of the ZR website where students write their programs.)

Math

- What is the relationship between an axis and a coordinate? (**Answer:** An axis is the basic line of measurement we use to locate a point. A coordinate is the measurement of a point’s distance along an axis.)

Physics

- What was the point of completing the bottle rocket activity? (**Answer:** The bottle rocket activity demonstrated the physical concepts of force and motion, and how forces affect an object in motion. Also, the action of the water being released from the bottom of the bottle, which forces it upward, is similar to the what the thrusters do on the SPHERES satellite.)

Field Day

- Did acting out the game help you think about your game strategy? If so, how?
- What did you learn about your team? What did you learn about the other teams?
- How did you work with your team members?
- What are some of the pros and cons of collaborating with other teams?
- What would you do differently if you were to act out the game again?

❷ Make connections to the concepts students will learn about in Week 2. Explain that students will now learn more about the math and physics behind the SPHERES satellites. They will learn more about programming, including how to use block diagrams to create a schematic or flowchart of their strategy or computer program, and they will increase their skills in programming by completing some more advanced tutorials in the ZR IDE.

What's the Right Order? (30 minutes)

Activity Description

Begin introducing students to some of the math and science concepts that are involved in both programming and maneuvering the SPHERES on the ISS by introducing the concept of the "order of operations".

❶ Show students the following statement: "Let's eat, Grandma!"

❷ Tell students that this simple invitation to join a meal becomes quite sinister when we remove the comma. Show them the resulting statement: "Let's eat Grandma!" Feign horror, and in a dramatic voice note that *commas prevent cannibalism every day*.

❸ Explain that in mathematics, there are objects like the comma that can change the meaning of the function (or, of an equation) entirely. Most often, these objects are parentheses—they indicate that whatever is in the

Materials Needed

For each student:

- Copy of RM 2.1: Order, Order!
- Copy of RM 2.2: Order of Operations Crossword

What's Going On?

GEMA is a common acronym to help people remember the correct order of operations. Each letter stands for what you should do, in the order you need to do it.

G = Grouping: Any symbol that sets part of the function apart, such as ( ) or [ ]. Remind students that when they see these symbols, they need to complete everything inside first. To do so, they still have to follow the other order of operations listed below.

E = Exponents: Exponents are the next part of the equation to solve. Wherever something is raised to a power, you need to evaluate it first and put the result back into the equation.

M = Multiplication: This includes both multiplication and division, as division is simply multiplication of the inverse of a value (for example, ½ is the inverse of 2). Multiplication is evaluated left to right, one sign at a time.

A = Addition: Like multiplication, this also includes its inverse, subtraction. Evaluate all addition and subtraction left to right.

middle of the two parentheses should be done before what is on the outside.

4. Show students the following two examples:

$$4 + 5 \times 3 = 19$$

$$(4 + 5) \times 3 = 27$$

Explain that there is a set of rules called the *order of operations* that ensure, just like commas, that math is consistent, reliable, and doesn't lead to cannibalism (with a wink to students).

5. Show students the acronym *GEMA*, which stands for *grouping*, *exponents*, *multiplication*, and *addition*. This acronym can help them remember the correct order of operations.

6. Demonstrate the order of operations with the following example:

$(9 + 7 \times 3) \times 3 - (6 \div 3)^2 + 8$

Remind students that they need to apply GEMA inside the parentheses too.

G: $(9 + 21) \times 3 - 2^2 + 8$

$30 \times 3 - 2^2 + 8$

E: $30 \times 3 - 4 + 8$

M: 90-4+8

A: 94

Note: In computer programming, multiplication is represented by an asterisk (*) since computers cannot distinguish the letter x from the operator ×.

7. Have students complete the practice problems on **Order, Order! (RM 2.1)**. You might have them work in pairs and then share their answers with the whole group.

Answers:

1. 320
2. 118
3. 80

8 Note that in addition to math operators, such as + and – , in programming there are *logic operators*, including *greater than* and *less than*. In order to program their player, students will need to learn how and when to use both.

What's Going On?

Operators

In computer programming, we use both math operators (e.g., +, –, and =) and logic operators.

Logic Operators

Logic operators include the following:

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
==	equals
!=	not equal

Bring It Home

Have students complete the Order of Operations Crossword (RM 2.2). Time permitting, you may opt to have students work in teams of two or three to complete the crossword on-site. (Note that the crossword and solutions are provided in the student materials.)

As If! (30 minutes)

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to show The Conditionals: The Basic of "If-Then" PowerPoint presentation

What's Going On?

Conditional statements describe the behavior of the computer in response to various situations. Conditionals give the computer instructions about when to do something.

An "if-then" statement is an example of a conditional:

If something is true, then.....

In this tutorial, students use an "If-then" statement and logic operators to program the SPHERES satellites to follow a path to multiple locations (i.e., from a Starting Position, then to a Point A, next to a Point B, and finally to a Point C).

Activity Description

In this activity, students work through the tutorial, *The Conditionals: The Basics of "If-Then,"* which reinforces the concept of conditional statements that they learned about in Week 1. In this tutorial, students combine the use of "If, Then" statements and logic operators to program a SPHERES satellite to move to multiple locations.

What's in a Vector? (45 minutes)

Activity Description

Students learn about *vectors*, which describe direction and magnitude in mathematics. They participate in a Vector Hunt, which reinforces the use of both magnitude and direction in describing a vector, and builds students' spatial and measurement skills.

1. Show students the Directions with No Direction PowerPoint presentation.

2. Explain that directions are an important part of getting around in our world. When you tell your friends how to get to your house, you don't just tell them how far they have to walk—you also tell them what direction to go. Without direction, your friends could end up in a completely different area of town. Not only is direction important in getting to people's houses—it is also important in the world of math and physics.

Materials Needed

For each student:
- Copy of RM 2.3: Is It a Vector?
- Copy of RM 2.4: Vector Hunt Answers
- Optional: Computer with Internet connection
- Paper
- Pen

For each pair of students:
- Ruler
- Protractor

For the group:
- Computer with Internet connection and projector to show the **Directions with No Direction** PowerPoint presentation

What's Going On?

Measurements are an essential part of math. Measurements are made in centimeters, grams, feet, liters, and many other metrics. However, all the measurements mentioned above are *scalar* measurements—they only refer to a magnitude, not a direction. *Vectors* are measurements that involve both magnitude and direction. Vectors can consist of a scalar and a direction—such as 30 meters north. However, without the direction, the 30 meters is once again a scalar. And without the distance, north is just a direction. Put them together, and you have a vector. For more information about vectors, see *Background Information*.

What's Going On?

Measuring Vectors

Vectors are most often pictured on the coordinate plane. There are two ways most commonly used to measure and define vectors on a coordinate plane. The first is by degrees, which is how we commonly measure vectors in giving directions (e.g., make a 90° turn to the right and follow the road 2 miles). The second way simply defines the final point of the vector if its "tail" is at the origin. While this method does not explicitly state a direction, it is defined by the direction that the vector would be pointing toward in order to reach the point that is defined. This method is the one more commonly used in mathematics, and it's what we use with the SPHERES satellites. Students have already used this method to define the Satellite pointing direction with the function setAttitudeTarget. For example, a vector with its tail at the origin and its arrow at (1,0,0) specifies a vector pointing along the positive x axis (pointposx).

For more information about measuring vectors, see *Background Information*.

3. Distribute **Is It a Vector? (RM 2.3)**. Have students decide whether each measurement is a scalar or a vector by determining if it has magnitude and direction, writing yes or no in the appropriate boxes. Review the answers with them.

Answers:

Measurement	Magnitude?	Direction?	Vector?
South	*no*	*yes*	*no*
30 meters northwest	*yes*	*yes*	*yes*
20 feet	*yes*	*no*	*no*
8 miles west + 10 miles south	*yes*	*yes*	*yes*
7 cm up	*yes*	*yes*	*yes*
East of the river	*no*	*yes*	*no*
15 km down the street	*yes*	*no*	*no*

4. Conduct a Vector Hunt. (Review the Week 2 "Before You Teach" for this activity to prepare the necessary materials. The student worksheet for this activity is RM 1.2 and the solution is available in Appendix 5.)

   a) Tell students that nine vectors have been hidden around the room. Show students the one at the front of the room, and demonstrate how to measure both its length and its direction with the ruler and protractor. Remind them that a vector has both size and direction.

   b) Distribute **Vector Hunt Answers (RM 2.4)**. Tell students that they will work in pairs to find the vectors and complete the handout. Have them determine who will be the recorder. Tell them how long they have to find and record the locations of all nine vectors.

   c) Send students off to hunt for vectors!

   d) Have students share their answers with the group.

   e) Have pairs use their answers from the Vector Hunt to answer the challenge question: "What is the Blue

SPHERES' favorite thing to do?" on page 2 of the handout.

Answer: ZR—for "Zero Robotics" (see solutions in **Appendix 6**)

f) If there is a vector on which there is significant disagreement, and if time permits, revisit the vector as a group and show students how to measure it.

5. Have students draw one coordinate axis and a vector that is 45 degrees counter-clockwise from north, then have them draw another coordinate axis with a vector 135 degrees counter-clockwise from east. Ask students whether they look the same. Why or why not? What magnitude do these vectors have?

6. *Optional:* Explain the concept of *dead reckoning*—a way to estimate the position of an aircraft or a ship without astronomical observations. Tell students that computing a dead reckoning position is simply an exercise in vector addition. They will now do an activity in which they'll use vectors to plot their course based on time and speed. They will then correct the positions with vectors representing winds and currents. Have students complete the Vector Voyage activity from the Teach Engineering website; also available in the student materials.

What's Going On?

Often it is best to use vectors in combination—why? Let's see.

Very rarely do we travel in a straight line. More often, the path involves several twists and turns, which in turn means that travelers use several different vectors in order to reach their final destination—with each vector having its own magnitude and direction. By adding the vectors together, it is possible to tell where the final spot is in relation to the beginning point. It is also possible to find the vector that covers the most direct path between the points.

Vector addition is easiest to picture by placing the tail of each vector at the arrow of the previous vector. Keep adding vectors in sequence until you're done, then draw a vector that starts at the beginning point and heads toward the end point. You can easily calculate the end point by adding the components of the vectors you're combining. For example, if there are two vectors, [4, 3] and [2,–4], the sum of the two would be [6,–1]. The x components would sum 4 + 2 = 6, and the y components would sum 3 + –4 = –1.

Though vector addition is not used directly in the ZR Middle School Program, one application of vector addition in ZR that the middle school students could understand at this point in the program is related to the function setAttitudeTarget. The setAttitudeTarget function specifies a pointing direction for the SPHERES.

In the middle school program, students will only need to know how to point the SPHERES along the x, y, or z axes. However, you can explain how vector addition can be used to specify other pointing directions. For example the vector (1,0,0), which points in the positive x direction, and the vector (0,1,0), which points in the positive y direction, would sum to (1,1,0) to specify a pointing direction 45 degrees counterclockwise. Students can also figure this out by assuming a magnitude of 1 (inch) for each vector and drawing the vectors head to tail (as they did in the Vector Hunt activity). The resulting vector (drawn from the first tail to the last head) would point toward the point (1,1,0). (See *Background Information* for examples.) Have students use their protractors to measure the angle and determine the resulting pointing direction (45 degrees).

Ask the students to figure out the direction the SPHERES would point if the following pairs of vectors were combined (they can add the vectors either using pictures or by component addition then use a sketch of the resultant vector to determine pointing direction):

(–1,0,0) + (0,1,0) *Answer*: 135 degrees counterclockwise (–1,1,0)

(–1,0,0) + (0,–1,0) *Answer*: 135 degrees clockwise (–1,–1,0)

(1,0,0) + (0,–1,0) *Answer*: 45 degrees clockwise (1,–1,0)

For more information about combining vectors, see *Background Information*.

Seeing in 3-D—How to Visualize Space! (30 minutes)

Activity Description

This activity introduces students to the concept of dimensions, specifically, three dimensions (3-D).

1. Display the first slide in the Dimensions presentation and tell students that this map can help them get to an ice cream store.

2. Ask them to imagine that they are standing at the point indicated on the center of the map. How would they get to the ice cream store following the grid "blocks"—that is, east-west and north-south?

   Answer: Move three blocks west and then three blocks north (or vice versa).

3. Point out that **two** pieces of information were required to get them to their destination: one coordinate along the east-west line, and one coordinate along the north-south line. Remind students that in Week 1 they learned that a *coordinate* is a number that describes position.

4. Ask students if they have ever heard of or seen a 3-D movie and if they know what the "D" in 3-D stands for. If students do not know, explain that the D stands for *dimension*. The dimension of an object or space is the minimum number of coordinates required to locate a point within that object or space.

   a) Since students had to use two coordinates to find the ice cream store, the map is a *two-dimensional* or 2-D object.

   b) If we go one dimension higher, we see 3-D space, as in a 3-D movie.

Materials Needed

For each student:

- Copy of **RM 2.5: The Plot Thickens**
- Computer with Internet connection

For the group:

- Computer with Internet connection and projector to show the **Dimensions** PowerPoint presentation

What's Going On?

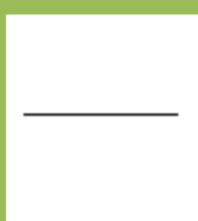
Dimensions

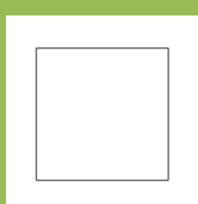
A point has no length, area, or volume. Because it has no size, it is therefore dimensionless, or zero-dimensional.

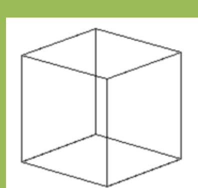
A one-dimensional (1-D) space or object only needs one coordinate to locate a point. For example, a line is a 1-D object.

By expanding the line in a new direction, you can create a square, or a flat plane. Now you need two coordinates to define the position of a point on the flat plane, so the object becomes 2-D.

If you lay the square flat and then stretch it upward, you get a cube (3-D object).







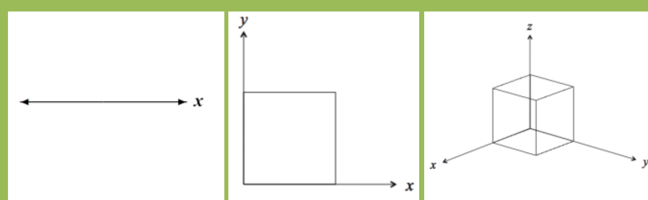
5. Have students think of objects that are 1-D (e.g., a temperature reading on a thermometer), 2-D (e.g., maps, circles drawn on paper, your shadow on the ground), and 3-D (e.g., a basketball or a car).

6. Tell them that these "expansions," from 1-D to 2-D to 3-D, can be viewed by plotting them, for example, on a graph. Show students the second slide in the Dimensions presentation. Explain that engineers and mathematicians alike use *x*, *y*, and *z* to refer to these plots. These lines are also called *axes*, which they learned about in Week 1.

What's Going On?

Plotting Position

Positions in 3-D space are not much harder to graph than those in 1-D or 2-D space. Each plot, from left to right, adds another dimension.



Remember *arrays*, which you learned about in Week 1? In ZR, we represent coordinates by using a float array. To represent 3-D points, we use a float array of size 3:

float coordinate [3] = {0.0, 1.0, 3.0};

These variable coordinates represent the point at x = 0, y = 1 and z = 3.

Likewise, 2-D points are represented using a float array of size 2:

float coordinate [2] = {0.4, 1.2};

These variable coordinates represent the point at x = 0.4 and y = 1.2.

7. Build on the mapping activity to discuss graphing and Cartesian coordinates. Remind students that they already explored graphing in one and two dimensions in Week 1. Explain that so far they've been looking at 2-D grids, but if we go one dimension higher, we'll get to see 3-D space.

8. Show the third slide in the Dimensions presentation and reassure students that points in 3-D space are not much harder to graph than those in 2-D space.

9. Using the graph to illustrate, explain that the only change in 3-D is that there is a third axis, the *z*-axis. If the *x*-axis refers to "length" and the *y*-axis refers to "width," it's easiest to think of the *z*-axis as "height." To

measure the distance along the *z*-axis, we need a *z* coordinate.

10. Walk students through how to plot points in 3-D using point A (2,2,2)—shown with a star on the slide—as an example. Distribute **The Plot Thickens (RM 2.5)** and have students plot the remaining points.

> ***Note:*** *The units on the z-axis are incremented by 0.5 while those on the x and y axes are incremented by 1.0.*

11. Give students more practice with 2-D graphing by having them explore the links in the student materials.

More Conditionals (30 minutes)

Activity Description

1. Have students work through the tutorial, *Conditionals: More Fun with "If-Then" and Logic Operators*, in which they practice programming with "If, Then" statements and logic operators and learn how to use the Debug feature of the ZR IDE.

2. Answer any questions students have and then have them complete the next tutorial, *Conditionals with Advanced Logic Operators*, to learn more about using logic operators.

Materials Needed

For each student:

- Computer with Internet connection

What's Going On?

Conditional statements

Conditional statements describe the behavior of the computer in response to various situations. Conditionals give the computer instructions about when to do something.

An "if-then" statement is an example of a conditional:

If something is true, then ...

Logic Operators

Logic operators include the following:

>	greater than
> =	greater than or equal to
<	less than
< =	less than or equal to

In the *Conditionals: More Fun with "If-Then" and Logic Operators* tutorial, students use an "If-Then" statement and some additional logic operators ('= =' and '! =') to show that the loop they created is called once per second by the SPHERES control system. In the *Conditionals with Advanced Logic Operators* tutorial, students use still more logic operators ('and' and 'or') and conditional statements to change the position and attitude of the SPHERES satellite.

What Else? (45 minutes)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment to show the **What If: An Introduction to Conditional Statements** PowerPoint presentation and the *Conditionals Continued: "If-Then-Else"* tutorial

Activity Description

Students work through the *Conditionals Continued: "If-Then-Else"* tutorial, which allows them to see that these statements allow more flexibility in programming. In addition, students use "Save As" to create a new project and learn to copy and paste blocks of code.

1. Remind students of what conditional statements are and work through an example, using the What If: An Introduction to Conditional Statements PowerPoint presentation.
2. Have students complete the *Conditionals Continued: "If-Then-Else"* tutorial.

What's Going On?

An "if-then-else" statement is another example of a conditional statement:

If something is true, then do this, else do that…

In this tutorial, students use an "if-then-else" statement to move the SPHERES satellite in multiple ways.

How to Make SPHERES Move (30 minutes)

Activity Description

Throughout Week 2, you are introducing students to some of the physics that affect how the SPHERES satellites move around, including dynamics and motion, Newton's Laws, and forces.

1. Ask students to think for a minute about the word *physics* and then share whatever things come to mind. ***Possible answers:*** Rockets blasting into outer space, nuclear explosions.

   If students have a hard time coming up with ideas, ask them, "What do Albert Einstein, two trucks crashing into each other, rockets blasting into outer space, and basketball all have in common?" Parse through students' ideas and record on newsprint or the board all ideas that do in fact relate to physics.

2. Explain that physics includes all of these things, and much more. In fact, physics is at the heart of all sciences—chemistry, biology, oceanography, meteorology—and is fundamental to all engineering. In this program, they will use physics to investigate and understand forces and motion so that they can control the satellites in the ZR game. Therefore, you will all spend some time looking at these concepts.

3. Review the Bottle Rocket activity from Week 1 and introduce Newton's Laws.

   a) Ask students what they remember about the motion of and forces at work with the rockets and revisit some of the questions from that activity. Ask, "How do you think forces, including gravity, might have affected the rockets?"

      Answer: Both gravity and drag—air resistance—slowed the bottle as it was in motion. This is an example of Newton's First Law.

      Tell students that Newton's Laws of Motion are three physical laws that relate forces acting on an object to the motion caused by the forces. The

Materials Needed

For the group:

- *Optional:* 2 chairs with wheels

For each pair of students:

- 2 balloons
- Straw
- Piece of string (10-15 ft., depending on the size of the room)
- Tape
- Newsprint or chart paper to record students' ideas

What's Going On?

Electrical engineers use physics to design computer chips, aerospace engineers use physics to design and launch rockets and spaceships, civil engineers use physics to make sure their bridges don't fall.

forces on the rocket are an example of Newton's First Law—an object at rest remains at rest until acted on by an outside force; an object in motion remains in motion until acted on by an outside force.

Note: *You can introduce Newton's Laws by showing students this video (scroll down the webpage to find the video; start at 0:19 and end at 2:09).*

b) Ask students, "At what point—or points—do you think the rockets were changing speed?"

Answer: Rockets would have obvious accelerations at takeoff and at the apex of the flight, but also always had acceleration due to gravity.

Tell students that if they knew the rocket's mass and the forces acting on it, they could use Newton's Second Law—force = mass x acceleration—to calculate its acceleration.

c) Ask students, "As the Bottle Rocket launched, what did you notice? Why do you think this occurred?"

Answer: As the rocket moved upward, water sprayed downward. The water spraying downward was an equal reaction force to the rocket propelling upward.

Tell students that this is an example of Newton's Third Law—for every action, there is an equal and opposite reaction.

What's Going On?

Newton's Laws of Motion are three physical laws that relate forces acting on an object to the motion that the forces cause.

Newton's First Law (The Law of Inertia): An object at rest remains at rest until acted on by an outside force; an object in motion remains in motion until acted on by an outside force.

What it means: If the total force on a body is zero, the body's motion does not change. A soccer ball sitting still in the grass will stay still unless someone kicks it. That soccer ball will keep rolling until another force (such as drag, or air resistance) slows it down and stops it.

Newton's Second Law: Force = mass × acceleration

What it means: A body undergoes acceleration when it is subject to a total force that is not zero. Therefore, if you know a baseball's mass and its acceleration, you can figure out the force exerted on it. If you know the force exerted on a baseball, and the acceleration of the baseball, you can find out the mass of the baseball.

Newton's Third Law: For every action, there is an equal and opposite reaction.

What it means: If you exert a force on something, that force will exert a force of the same magnitude in an opposite direction on you.

For more information about forces and Newton's Laws, see *Background Information*.

4. Tell students that Newton's Laws focus on forces—but how do we know what the forces acting on an object are and whether they will just cancel each other out? Explain that the easiest way to explore forces is to draw the forces as vectors with magnitude and direction. Forces with equal magnitude and opposite direction when added together would cancel.

   a) Have students imagine that they and their identical twins are standing on either side of a large box, such as one that might hold a refrigerator. Have students draw a picture in which they imagine themselves and their twins each pushing on their side of the box. Then have them draw an arrow showing which way each person is applying the force to the box. Explain that if they both push with the same amount of force, the box does not move at all—but there are still forces present in the system.

   b) Ask students to imagine that they and their older sibling are pushing on the box. What would happen if their older sibling pushed a lot harder than they did? Would the box move? How?

What's Going On?

Friction: A force that resists the motion of one object over another object.

Gravity: A force that pulls any two objects in the universe towards each other. Gravity of the Earth: A force that pulls objects towards the center of the Earth.

Normal force: A force that supports objects when gravity pulls them down.

5 Explain that objects behave differently in space and on Earth due in large part to the lack of gravitational pull in space.

a) Ask students what they know about *gravity*.

b) Ask students, "If gravity is pulling everything toward the center of the earth, why aren't books pulled through tables toward the floor, clothing constantly falling off hangers, and buildings sinking into the ground?"

Answer: Normal force, which is equal but in an opposite direction to the gravitational force applied on an object, helps objects counteract the force of gravity.

c) Have students draw gravity and normal force in their vector drawings. Point out that these forces can balance each other out, though normal force can't keep an object from falling—it just keeps you from being pulled through the floor of your house!

d) Ask students to compare pulling an object across a carpet and a tile floor. What happens to the movement over the carpet, and why?

Answer: Friction—a force that is caused by two objects rubbing together—causes the object to move more slowly over carpet than tile.

Tell students that friction is not present for an object floating in space; ask them why they think this is the case.

Answer: Friction only occurs when two objects rub together. An object in space is not rubbing against anything else—even air resistance is minimal in space.

6 Explain that students will see SPHERES satellites at work in their space environment. Note that it's important to remember that space is free from Earth's gravitational pull—satellites move in more directions in space than they will on Earth. There is also no friction or drag in space. However, since SPHERES satellites

operate aboard the ISS, which is filled with air, they are subject to forces exerted by the air.

❼ *Optional:* Demonstrate an interactive activity highlighting Newton's Laws (available in the student materials).

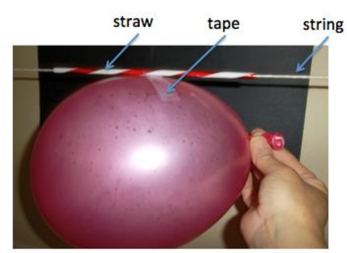
❽ Show students a video of how Newton's Laws affect the movement of the SPHERES on the ISS (scroll down the webpage to find the video; show the video from 2:09 on—also in the student materials).

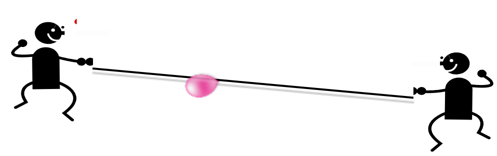
❾ Reinforce the concepts in Newton's Laws of Motion, especially Newton's Third Law, by having students do an activity called Thruster Balloons:[§]

a) Blow up a balloon and hold on to the end. Ask students what they think will happen if you let go. After allowing students to answer, let go and watch the balloon fly around the room.

b) Explain how Newton's Third Law of Motion dictates how the balloon moves. The balloon is exerting a force on the air to push it out of the end. The air then pushes back on the balloon, moving it in the direction opposite to where the end is pointing.

c) Have students get into pairs and give each pair two balloons, a straw, and a piece of string. Let each student have a turn blowing up the balloon, taping it to the straw, and watching it move down the string as the partners hold both ends of the string. Make sure that everyone has an opportunity to participate. For the best results, students should be careful to align their balloon with the string and pull the string tight before releasing the balloon.





[§] Special thanks to science-class.net for help in refining the idea of Thruster Balloons.

d) Ask students to try to explain to you exactly what happened to the balloons and how it relates to Newton's Third Law of Motion. Remind students that similarly each of the thrusters on the SPHERES satellite releases CO_2, creating a force on the satellite in the opposite direction. The balloon can only release air in one direction but the satellite has thrusters on all sides for precise control.

10. *Optional:* If you think that students need additional help understanding Newton's Third Law, have them try Chairs on Wheels:

a) Arrange two chairs with wheels in front of the class.

b) Ask for two volunteers to sit on the chairs facing each other.

c) Have the two students extend their arms straight forward and place their palms together. Make sure that their palms are touching and their fingers are pointing up.

d) Have one student push against the palms of the other. (If only one of the chairs is on wheels, have that student do the pushing.) Ask the rest of the group to observe what happens.

Answer: If both chairs have wheels, both students will move away from each other even though only one student was pushing. If only one chair has wheels, the student that pushed is the one who is moving away.

e) Explain that Newton's Third Law says that when one body exerts a force on another one, then the body exerting the force will experience a force equal and opposite to the force exerted.

What Friction? (15–30 minutes)

Note: *A long flat surface, such as a table, is needed for this activity.*

Activity Description

During Field Day, students may have had the opportunity to see some dry ice demonstrations. If not, it is important to conduct this optional activity because these demonstrations show how the SPHERES move on the ISS. That is, the activity demonstrates *frictionless dynamics* (the study of movement—students learn about this in Week 5) and Newton's Laws. After an initial demonstration, students predict the path of a block of dry ice sliding across a table.

1. If students had the dry ice activity as part of Field Day, introduce them to the concept of *frictionless dynamics* by reviewing that activity. Make connections between the activity and the physics that affect the SPHERES. Ask students, "How did the ice move? How does this compare to how objects you see every day move? Why did the ice move so differently from how you might have expected it to?"

   Answer: At room temperature, dry ice sublimates directly from a solid into a gas. When a block of dry ice is placed on a flat surface, this sublimation results in a layer of gas between the dry ice and the surface, which allows the dry ice to move across the surface with essentially no friction.

2. Explain that on the ISS the SPHERES move essentially without friction, because the air resistance is negligible. This movement may seem counter-intuitive because it differs considerably from our day-to-day experience, where friction quickly dissipates an object's kinetic energy. The dry ice activity helps us understand this type of frictionless movement a little better.

3. *Optional:* Introduce the Dry Ice Dynamics activity if you were not able to do it during Field Day. If you already

Materials Needed

For each student:

- Marker

For the group (to perform optional dry ice activity):

- Roll of poster paper
- 4 heavy objects
- Materials for the obstacle courses
- Dry ice ramp (provided)
- One pair of insulated gloves with long sleeves (welding or winter gloves work well)
- Safety glasses
- 10-lb block dry ice
- Cooler (for the dry ice)
- Hammer
- Large storage tub or laundry basket
- *Optional:* 2 1-ft pieces of 2 x 4 lumber
- *Optional*: Fan

What's Going On?

In this activity, students predict the path that a block of dry ice will take when it slides off a ramp. At room temperature, dry ice sublimates directly from a solid into a gas. When a block of dry ice is placed on a flat surface, this sublimation results in a layer of gas between the dry ice and the surface, which allows the dry ice to move across the surface with essentially no friction. This effect should be described to students and demonstrated by sliding a block of dry ice across the table before the activity begins.

conducted this activity on Field Day, you do <u>not</u> need to repeat it. Tell students that their task is to predict the path of a block of dry ice sliding across a table.

a) Set up one of the obstacle courses in **Appendix 5: Obstacle Courses**. Note that the distance between items is unimportant, but the relative angles between items is extremely important.

b) Give each student a marker and ask them to sketch the path they expect the dry ice to take on the paper "field."

This activity involves dry ice, which has a temperature of –78.5°C when solid. Touching dry ice with bare skin will cause frostbite instantly. Students should NOT handle the dry ice at any time! Take care and only handle the dry ice while wearing insulated gloves. Be sure to explain this to students before the activity begins.

c) Put on gloves and safety glasses and chip a piece of dry ice off the larger block using the hammer. The piece should be small enough to fit onto the dry ice ramp.

d) Smooth any rough edges on the piece of dry ice by rubbing it against a flat surface.

e) Place the tub or laundry basket on the floor where most students predict the dry ice will leave the table.

f) Tell students to step away from the table. They should be close enough to observe but not close enough to touch.

g) Place the chip of dry ice on the ramp and allow it to gently slide onto the table.

h) Compare the actual path of the dry ice to students' predictions and discuss why the dry ice moved as it did.

i) Repeat for each of the obstacle courses.

j) If there is time and dry ice remaining, ask students to place the fan and the two-by-fours to direct the dry ice to a specific target. Note that making the dry ice stop on the table is particularly challenging.

Breaking Down Programming (30 minutes)

Activity Description

Before teams begin to strategize for the Intramural Competition, students learn about functions, which are very useful in programming to break down the code and keep track of the various pieces.

1. Use the Programming Functions PowerPoint presentation to discuss the concept of *functions* in programming and to review other basic programming concepts students have learned. Explain that programmers find it easier to break up their code into separate sections that perform different tasks. These sections of code are called functions. Remind students that they used several SPHERES controls functions in the Week 1 tutorials about arrays.
   a) Review functions.
   b) Use the Code Review slides to direct students to examine code from one of the projects they created in their tutorials and to identify the various programming elements they have learned about thus far.

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projector to show the **Programming Functions** PowerPoint presentation

What's Going On?

Students have learned about the following basic elements of programming:

Arrays: Data type that stores multiple pieces of data in one place.

Conditional statements: Describe the behavior of the computer in response to various situations. A computer will respond differently to different commands.

Data types: Using specific data types allows programmers to represent mathematical information on a computer. Any data used in a program must be "declared" a particular type so that the computer knows to store and use it.

Functions: Separate sections of code that perform different tasks. Programmers often find it easier to break up their code into functions.

Operators: In computer programming, we use both math operators (e.g., +, –, and =) and logic operators (e.g., >, > =, and = =)

Loops: These allow computers to carry out the same instructions over and over again, though the programmer only needs to write the instructions once.

Variables: When writing a computer program, you often want the program to "remember" a value—you can do this by assigning the value to a named variable. Each variable has a name and contains a value.

What's Going On?

The four SPHERES controls that students will use in the middle school ZR game are:

- *setPositionTarget* sets an x, y, and z position target for the satellite to move toward
- *setAttitudeTarget* specifies a unit vector (called the attitude vector) for the satellite to point toward; it can be used to rotate the satellite.
- *getMyZRState* retrieves the ZR state (position, velocity, pointing vector, rates) for the current satellite
- *getOtherZRState* retrieves the ZR state (position, velocity, pointing vector, rates) for the second satellite

c) Show students the SPHERES Controls slides. Explain that these programming functions can be used by students to move their satellite without having to program every aspect of that movement.

> **Note:** *It is important to understand the 'ZR state' of both the current and second satellites before you can manipulate/move them.*

Let the Games Begin! (1.5 hours)

Materials Needed

For each student:

- Computer with Internet connection

For each team:

- Newsprint
- Markers

For the group:

- Computer with Internet connection and projection equipment to display *Introduction to Game Mode* tutorial and **Game Overview** and **Let the Games Begin**

Activity Description

Introduce the rounds of competition and have students begin working on their programs.

1. Introduce the four rounds of competition: the Intramural Competition, the practice Regional Competition (which will not eliminate anyone), the Regional Competition, and the National Competition.

   a) **Intramural Competition:** Tell students that they will begin to work in their teams to develop a program for the ZR game. Toward the end of Week 3, each team will compete their code against the code developed by other teams in the group in an exciting intramural event.

b) **Practice Regional Competition:** At the end of Week 3, based on the results of the Intramural Competition, the group will select one code to submit to a practice competition against the other teams in the region.

c) **Regional Competition:** The results of the practice Regional Competition will be available at the beginning of Week 4. Students will analyze the strategies used by the teams in their region and work together to create their final code. The final code for the Regional Competition will be submitted to the regional competition at the end of Week 4. The Regional Competition determines the regional 1st, 2nd, and 3rd place champions.

d) **National Competition:** In Week 5, every team in a region will have the opportunity to collaborate with the regional winner to improve this code before it is sent to the ISS. Then National Competition will be conducted live aboard the ISS by astronaut referees.

> ***Note:*** *Students will have a lot of programming tasks in Week 4 to finalize their code in time for the Regional Competition.*

2. Have teams review the game manual in the student materials, which they will need to refer back to as they begin developing their programs.

> ***Note:*** *The game manual introduces some game specific functions that students will need and includes information about how to use them.*

❸ Have students complete the *Introduction to Game Mode* tutorial, in which they will learn to use the game mode (as opposed to the free mode they have been using up to this point). The tutorial includes tips for programming in the game mode and learning how to compete against a "standard" player.

> **Note:** *This is a good time to have students look at part 2 of the* ***Game Overview*** *slideshow.*

❹ Show students the Let the Games Begin PowerPoint presentation to help them **start** to develop their programs (they will continue working on them next week).

> **Note:** *The presentation uses a generic example to demonstrate a process that students can apply to start developing code for the game.*
>
> 1) *Have teams review their strategies they developed for Field Day.*
> 2) *As you step through the presentation have one member of each team record their current plans on newsprint using:*
>   a) *Sentences*
>   b) *Lists*
>   c) *Add coordinate information from the game manual*
>
> *Try creating a flow diagram based on time counters*

❺ Have each team start to create their new program in the ZR IDE using the game mode.

6. Afterwards, have teams develop a timeline to guide their programming. Ensure that each team assigns roles, tasks, and some tentative intermediate deadlines for each student on the team. For example, students may decide to help with strategy, review the game manual, develop flow charts, convert plans into code, run simulations, or help debug.

> ***Note:*** *In Weeks 3 and 4, you will need to set aside time in the schedule for planning and organization, and for checking in with each team of programmers. Only teams with a plan will be successful in the ZR game. Teams that have a plan but do not decide who will do what until too late will not succeed unless they have one very capable student who can—and will—do the whole thing alone.*
>
> *You will need to monitor teams to ensure that each member can make a meaningful contribution, no one person dominates, and enough time is put into the programming so that it is finished in time. This may very well be harder than the technical computer science challenges involved in this program.*

Reflect and Assess

At the end of this week, provide students with opportunities to reflect on what they have learned both this week and during the program so far, and to connect the programming and science and mathematics concepts they have learned about to the ZR game.

Achieving Our Goals

Display the student learning outcomes for this week, which students can see in the Week 2 student materials. Ask students to reflect on how well they believe they have achieved these outcomes, using the Questions for Reflection on page 6. In particular, ask students to reflect on how it feels to work together in teams to solve problems.

What Do You Know Now?

Revisit the K-W-L and add to it based on what students have learned and what they wonder about at this point.

Do You Know More than a 12th Grader?

Quiz your students on some of the terms and concepts they learned about this week. Some sample questions and suggestions for content are included below.

1. How many axes represent coordinates in 3-D? Name them. (**Answer**: 3; x, y, and z)
2. Name one logic operator used in programming. (**Answer**: Any of the logic operators listed below is acceptable; students should be able to both write and name the operator)

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
==	equal to
!=	not equal to

3. What does GEMA stand for? (**Answer**: Grouping, Exponents, Multiplication, Addition)
4. What is the difference between scalar and vector measurements? (**Answer**: Scalar measurements involve magnitude only; vectors involve both magnitude and direction.)
5. How many Newton's Laws are there? (**Answer**: 3)
6. What is the first of Newton's Laws often called? (**Answer**: Law of Inertia)

Jeopardy!

Develop a class game of Jeopardy! using the Resources section of the ZR website or the information included in this guide. You might want to use the following categories:

1. Newton's Laws and Forces (call it "May the Force Be With You!")
2. Order of operations (call it "Let There Be Order!")
3. Vectors and dimensions (call it "3-D Vision")
4. Grids, graphing, and Cartesian coordinates (call it "It's a Plane! It's a Plane!")

Week 3:

Time to Play

Overview

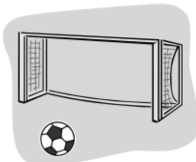
During the third week of this program, students will code their team players for a first trial run of their program. This Intramural Competition will allow students to try out their ideas and see what works well and what doesn't; the best code from the Intramural Competition will be submitted to a practice Regional Competition at the end of the week.

Getting Started

Before Week 3, review the information below and gather the materials needed.

> **Note:** *Students will spend the majority of their time this week programming in order to finish their code for the Intramural Competition.*

Goals



Students understand more complex mathematics and physics concepts underlying SPHERES and the ZR game and begin to write programs for the game.

Student Learning Outcomes

By the end of this week, students should do the following:

- Understand some more complex math and physics concepts involved in maneuvering the SPHERES satellites, including kinematics:
    - Mass vs. weight
    - Speed vs. velocity
- Code their intramural game strategy for the Intramural Competition!
- Understand how, and be able, to write and debug code using the ZR game controls in the ZR IDE for their player
- Be able to work together in teams to solve problems related to activities and their ZR game strategy and coding

Before You Teach

- Copy the RMs needed.
- Review the suggested flow of activities below and estimate what activities will be scheduled for each day based on the time you have.
- Preview the PowerPoint presentations.
- For “Reaching Warp Speed”: Create tables on pages 86, 87, and 89 on newsprint or the board.
- Set out the materials needed for each session.
- Determine which items from the “Reflect and Assess” section you will use.

Suggested Flow of Activities[**]

[**] Include **Take a Break!** activities throughout the week (see p. 93 for more details)

Activities

What Do You Know? (15 minutes)

Using the Week 1 review provided at the start of Week 2 materials, do a very brief re-cap of Week 1 concepts:

- SPHERES and the premise of the ZR game
- The basic components of a program
- How the ZR IDE will be used to create and debug their program
- Forces and Newton's Law of Motion and how they relate to the game

Then engage students in a conversation about what they learned in Week 2 using the questions below (organized by overall topic) and discuss connections to new concepts they will be learning about in Week 3:

Programming

- What does the ZR IDE stand for? What does it do? (**Answer**: The ZR IDE is the Zero Robotics Integrated Development Environment. It has a graphical editor that will make it easier for us to create our ZR game program.)
- Can you identify the following basic elements of a program in this code—data type, variable, conditional statement, loop, function? (Show students a sample of your choosing.)
- What are the SPHERES controls? (**Answer**: They are functions in the ZR IDE that help to maneuver the SPHERES satellites. We will use them to write our programs.)

Math

- What does GEMA stand for? (**Answer**: This is the order of operations in math and it stands for Grouping, Exponents, Multiplication, Addition).
- How many dimensions can they name? What is an example of each? (**Answer**: 1-D, 2-D, 3-D. Possible examples include a line (1-D), a shadow (2-D) and a ball (3-D).)

- What is the relationship between an axis and a coordinate? (**Answer**: An axis is the basic line of measurement we use to locate a point. A coordinate is the measurement of a point's distance along an axis.)
- What is the difference between a scalar and a vector measurement? (**Answer**: Scalar measurements refer only to magnitude. Vector measurements involve both magnitude and direction.)

Physics

- What are Newton's Laws of Motion? (**Answer**: They are three physical laws that relate forces acting on an object to the motion the forces cause.)
- Fill in the blanks in the statements below:
- An object remains at rest until acted on by __ _______ _____. (**Answer**: 'an' 'outside' 'force')
- force = ______ × _______ (**Answer**: 'mass' 'acceleration')
- For every action, there is an _____ ___ ________ ________. (**Answer**: 'equal' 'and' 'opposite' 'reaction').
- Describe what each of the three Laws of Motion mean, in your own words or using examples.
    - The first law (known as the "Law of Inertia") says that an object at rest will stay at rest and an object in motion will stay in motion, until a force moves or stops that object. Students may give various examples, including the apple and worm example from the interactive.
    - The second law describes the relationship between the mass of an object and how much force it takes to accelerate the object. Students may again give various examples, including the worm moving the apple from the interactive.
    - The third law simply says that any action will have an equal reaction. Examples from students might include the blast of water from the bottle rocket as it climbs into the air or any of the demonstrations they have had on this law.
- Is this statement true or false: On the ISS, the SPHERES move essentially without friction? (**Answer**: True)

Connections to Week 3 Concepts

Explain that in Week 3, the program will be building on all of the concepts students learned in Weeks 1 and 2. In Week 2, they learned some of the mathematics and physics concepts behind how SPHERES operate. This week they learn a little more math and physics to explain how SPHERES move and 'act' in space. In Week 3 students use the programming skills they have learned to far to write a programs for an Intramural Competition.

SPHERES: Here, There, Everywhere! (30 minutes)

Materials Needed

For each student:

- Copy of RM 3.1: Name the Parts

For the group:

- Computer with Internet connection and projection equipment to show the **All About SPHERES** PowerPoint presentation PowerPoint presentations

Activity Description

In this activity, students will learn more about the SPHERES and understand the more advanced physics and mathematics concepts behind how the satellites work.

1. Start with a brief review of what students know about SPHERES so far.
   a) Build off the SPHERES parts labeling activity in Week 1 (when students labeled their paper SPHERES). Give each student a copy of **Name the Parts (RM 3.1)**, which has a blank SPHERES image and list of its external parts. Have students 'match' up the parts listed with the image. This will help you assess their knowledge of the satellite's parts. You may choose to do this as teams and then do a whole-class review of the correctly labeled image on slide 2 of the All About SPHERES PowerPoint presentation.

2. Use the All About Spheres PowerPoint presentation to review some basic concepts about the SPHERES before introducing students to some of its more advanced features—the presentation explains in more detail how the SPHERES know where they are and how they control their position. For example, slide 13 explains how SPHERES move:
   a) Explain that each satellite has a carbon dioxide tank and 12 thrusters: Each thruster releases the gas from

the satellite in one direction (indicated by the arrow in the figure), creating a force on the satellite in the opposite direction.

b) Ask students to look at the location of the thrusters on the satellite in the figure and the direction that gas is released. Have them guess which way each thruster will make the satellite move.

❸ Tell students that in order to be able to program the SPHERES to do certain things as part of the game, they need a better understanding of how they work. So far, they learned a lot of the math that will help them figure out a SPHERES position for example. This week are also going to be learning more physics concepts related to how the SPHERES move.

Thrown for a Loop! (30 minutes)

Activity Description

In the *For Loops* tutorial, students work on conditionals, focusing this time on **for loops**. Students use a **for loop** to repeat an action a set number of times. Their goal is to locate the position of the other satellite in the game, then program their satellite to move toward the other satellite but to stop halfway. Some students may have difficulty with this tutorial, particularly with the midpoint calculation. However, two useful SPHERES control functions are introduced in this tutorial: "getMyZRState" and "getOtherZRState," which they will use again later in the week.

Materials Needed

For each student:

- Computer with Internet connection

What's Going On?

Loops allow computers to carry out the same instructions over and over again, but only require the programmer to write the instructions once. Thus, loops simplify computer programs. In ZR, *for loops* can be used to read SPHERES state information and to initialize vectors. Vectors define how the satellite will move, spin, and rotate. In this tutorial, students learn to use a *for loop* to repeat an action a set number of times—in this case, to find the position of the second satellite and program their satellite to move toward the second satellite, but stop halfway

It's Game Time (several hours)

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Planning notes from Week 2
- Newsprint
- Markers

Activity Description

Teams continue to develop programs for the Intramural Competition and for the practice Regional Competition at the end of Week 3 based on the work they began at the end of Week 2.

1. Have teams briefly review their progress from Week 2. Have one team member record their plans on newsprint. Then have team members discuss any possible adjustments they think they should make.

2. After making any revisions to their plans, teams should review roles, tasks, and deadlines for each student on the team and then begin programming.

3. Have teams code their player based on their plans.

   a) Suggest that teams continue to use the Let the Games Begin PowerPoint to guide them, depending on their progress at the end of Week 2.

   b) Remind students to refer to the game manual or game overview for game details and scoring information; they can also refer back to the tutorials.

   c) Suggest that students try creating some code to try out the game-specific functions that were introduced in the game manual. Be sure that students create and test their code in game mode. This task could be split between different teams by assigning each team a different game-specific function to try.

   d) Students should create code, simulate, debug, and then repeat. Remind students that including Debug statements in their programs can be helpful for debugging.

   e) Have students save successful variations of their programs with a new name before they make changes so they can easily go back to the previously successful version if necessary.

f) Encourage students to run their code against "standard players" as described in the *Introduction to Game Mode* tutorial. They can try this even if they have not written code that completes all of the game objectives.

g) Check in with teams and have them revisit tasks and deadlines.

4. Throughout the week, have both students and teams reflect on the ZR programming using the Questions for Reflection from the Program Overview:

   a) How far along are you with developing your code?

   b) What challenges are you having?

   c) What are possible solutions that will help you address the challenges?

Reaching Warp Speed (1 hour)

Activity Description

The next set of topics relates to the field of *kinematics*, which is the study of motion without looking at the forces that affect that motion. Before that, however, it is important that students have an understanding of mass vs. weight and speed vs. velocity.

1. **Mass and Weight:** Make sure students understand that mass and weight are NOT the same!

   a) Begin by asking students "What weighs more, a pound of lead or a pound of feathers?" The answer, of course, is that they weigh the same—at least, on Earth. On Earth, a pound is a pound is a pound. But, space exploration has made it clear that weight is not equal on Earth and in space.

   b) Show a video (for a funny introduction) or another video (to see astronauts on the ISS explain the concepts) explaining the difference between mass and weight.

Materials Needed

For each student:

- Computer with Internet connection
- Copy of RM 3.2: A Question of Kinematics

For the group:

- Computer with Internet connection and projection equipment to show video and the **Mapping with Speed** PowerPoint presentation
- Table on newsprint or board, prepared in Before You Teach

What's Going On?

Mass

Mass is the amount of matter in an object. Mass does not have anything to do with gravity, and does not change depending on where you are on Earth or in the solar system. If you have a mass of 50 kilograms on Earth, you will have a mass of 50 kilograms on Mars, the moon, Jupiter, or the ISS.

Weight

Weight is a measure of the force of gravity on an object. Weight does change depending on the local force of gravity. On Earth, your weight will be the same everywhere—if you weigh 100 lbs. in the U.S.A., you'll weigh 100 lbs. in Australia. However, you'll weigh about 38 lbs. on Mars, 17 lbs. on the moon, 236 lbs. on Jupiter—and on the ISS, you'll basically weigh nothing! (You can have students explore their weight on other worlds using this website.) So, remember that although SPHERES satellites might be weightless on the ISS, they will still have the same mass that they do on Earth.

c) *Optional:* To teach students about mass and weight—and show them the difference between Earth and the ISS—conduct this activity from NASA.

> ***Note:*** *Materials and preparation for this activity are not listed in this guide.*

❷ **Speed and Velocity:** Explain that speed and velocity both measure how fast a point or object is moving. Speed, however, has no direction while velocity does.

a) Show slide 2 in Mapping with Speed PowerPoint presentation and tell students to imagine that two cars are both moving at 45 miles per hour (mph), but on opposite sides of the highway (i.e., they are going in opposite directions). They have the same exact speed, but their velocities are different because their motions have opposite directions. So speed only has magnitude, whereas velocity has magnitude and direction.

b) This is where the vectors students have learned about come in handy. Consider the arrows in the image in slide 2. You can see that they have the same size. They have the same magnitude, and therefore identical speed. Let's tack on their directions: Car 1 is going at 45 mph to the right and Car 2 is going at 45 mph to the left. Once you express speed with its direction, you've found the velocity.

c) Show slide 3 in the **Mapping with Speed** presentation and have students use the diagram shown in the slide to fill out the first table on **A Question of Kinematics (RM 3.2)**:

Answer:

Vehicle	Direction	Speed	Velocity
Helicopter	NW	100 mph	100 mph NW
Airplane	NE	500 mph	500 mph NE
Eagle	SE	30 mph	30 mph SE
Runner	W	15 mph	15 mph W
Rocket	N	25,000 mph	25,000 mph N
Satellite	E	17,000 mph	17,000 mph E

3 Now, introduce students to kinematics.

a) Ask students to draw (on a piece of paper, whiteboard, chalkboard) a sign for speed limits similar to any they may have seen on a road of highway. Show students slide 4 in the **Mapping with Speed** presentation.

I. Ask students, "What is this sign telling you?" (The sign is telling you not to go faster than 65 miles per hour.)

II. Tell students, "Now suppose you are not driving so fast" (that is, your speed is much lower than 65 miles per hour). To go faster, what do you do? (**Answer:** You press the gas pedal, and your car accelerates so that its speed goes up.)

III. Then ask, "What happens to your position/location while you are driving?" Students should recognize that by the time they are done driving, they are miles away from where they were initially (that is, their position has changed).

b) Explain that *kinematics* is the study of motion when things accelerate as they just considered in the example above. In kinematics, we don't deal with forces, just with how our movement changes over time.

c) Now have students imagine a scenario that involves kinematics: two friends (say, Jane and Paul) want to race their rocket car down a very long straight street. The rocket car starts parked at mile marker "0" at 8 a.m. and fires its rocket engine. The rocket fire creates a force that causes the car to accelerate.

   I. Ask students to try to fill in the second table on A Question of Kinematics (RM 3.2) to show what they think will eventually happen.

   Answer:

	Start	Finish
Position	mile marker 0	much farther away
Speed	zero	zero
Acceleration	none	none
Time	8 a.m.	Later that day

   II. Then ask students to share their answers to the following questions:

   1. What will happen when the rocket engine stops firing?

      Answer: The rocket car will slow down and eventually stop.

   2. Why would you expect the rocket car to stop?

      Answer: Because of friction.

What's Going On?

Position: where you are

What Does It Mean? Usually, we define a point that has zero distance before we measure anything else. This could be the origin of a graph or the start line at a 100-meter dash, for example. The position of any point is its distance from this reference point.

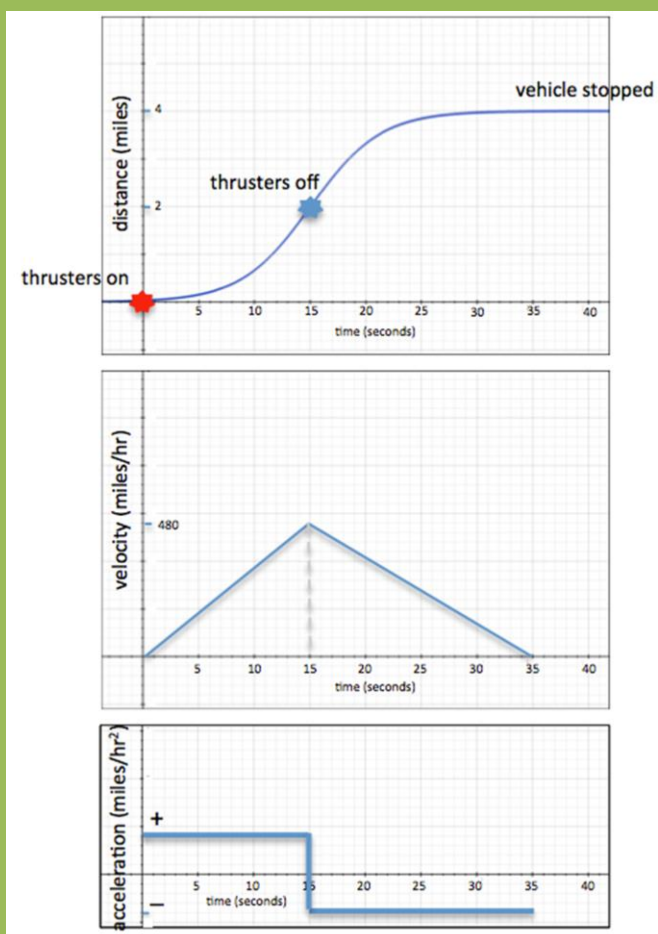
Speed: how fast you are going

What Does It Mean? Speed measures how far the point (or any object) will move over a certain period of time. Speed is easy to calculate:

average speed = distance ÷ time

Acceleration: how your speed is changing

What Does It Mean? This can be a bit tricky to understand. It's easiest to think about what would happen if there was no acceleration. If Paul stood still, he would remain still. If he were walking, he could keep walking at the exact same speed (no acceleration). Acceleration changes speed or direction. The negative form of acceleration is deceleration. While driving, pushing the gas causes the car to accelerate and go faster; pushing the breaks causes the car to decelerate and thus slow down.



d) Now ask students to imagine that Jane and Paul are racing their rocket car in space; students should assume that they start their race outside the International Space Station.

I. The rocket car starts at mile marker "0" at 8 a.m. and fires its rocket engine once. The rocket fire creates a force that causes the car to accelerate. What will happen when the rocket engine stops firing?

Answer: The car will continue to move at a constant speed equal to its speed when the rocket engine stopped firing.

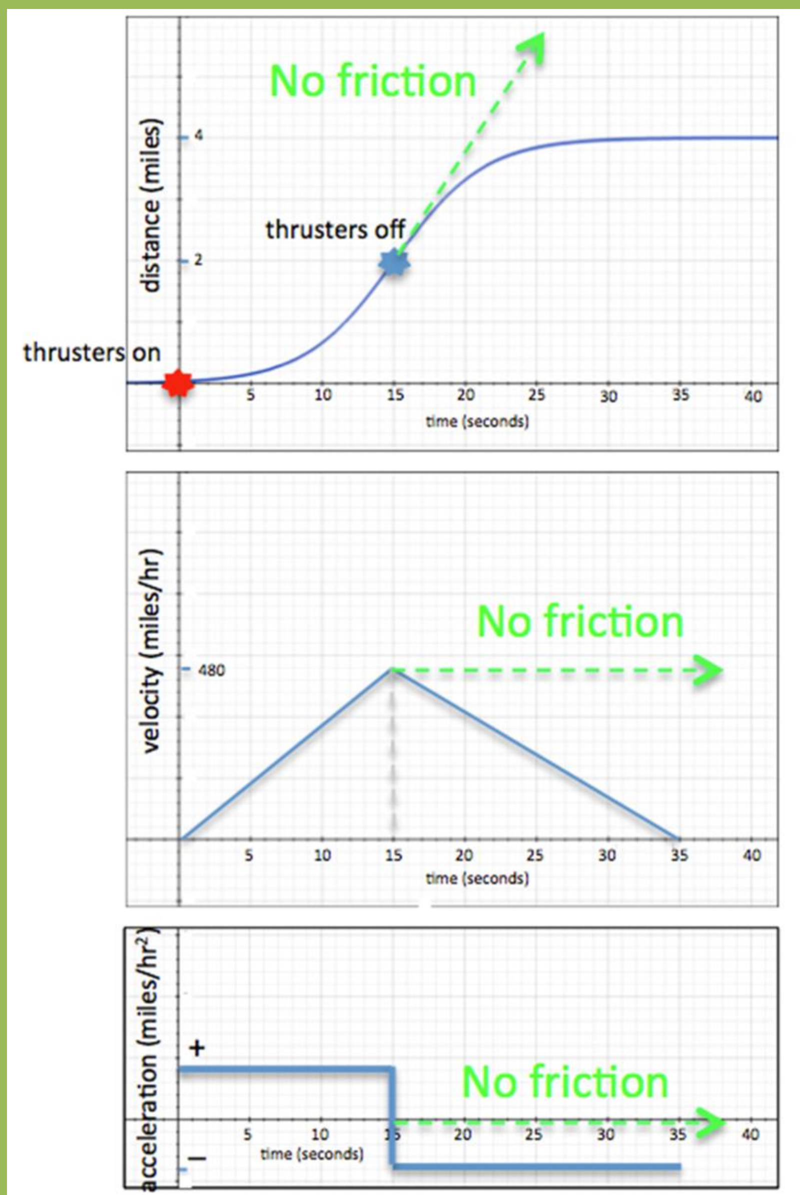
II. Ask students to try to fill in the third table on A Question of Kinematics (RM 3.2) to show what they think will eventually happen (assuming no friction).

Answer:

	Start	Finish
Position	mile marker 0	much farther away and still moving
Speed	Zero	Same speed it was travelling when the rockets stopped firing
Acceleration	None	none
Time	8 a.m.	Later that day

What's Going On?

The plots show how the resulting position, velocity, and acceleration of the rocket car will change in the absence of friction. The rocket car will continue to move further away from mile marker 0 at a constant velocity equal to its velocity at the moment the rocket stopped firing it thrusters. (The top graph is included on slide 5 of the *Mapping with Speed* presentation if you want to show it to students.).



❹ Students need to be aware of kinematics when they create their programs for the SPHERES. If a program tells the SPHERES to move to a point, it will activate its thrusters to create a force that will move it in that direction. Once it reaches the point, it will stop activating its thrusters but will continue to travel in the same direction unless it is given a command to stay at that point (in which case it will activate the necessary thrusters to keep it at that point). Have students try the simple *Hints About SPHERES Loop Dynamics* tutorial to demonstrate this point.

If the SPHERE Is Here, Then . . . (30 minutes)

Materials Needed

For each student:
- Computer with Internet connection

Activity Description

In the *Applied Conditionals* tutorial, students will learn more about how to program the SPHERES using conditionals. Specifically, students learn how to use information about the satellite's state to make decisions. The SPHERES Control "getMyZRState" retrieves the following information about your satellite:

- Position (x,y,z)
- Velocity (v_x,v_y,v_z)
- Pointing vector (N_x,N_y,N_z)
- Rotation rates (ω_x, ω_y, ω_z)

Student can use this information in conditional statements to implement their player strategy. Note that so far students have been developing programs based on time counters instead of satellite state information. They may or may not want to use the concepts presented in this tutorial to modify their programs for the Intramural Competition. However, they should be encouraged to incorporate these concepts into their programs during Week 4 for the Regional Competition.

After this activity, engage students in a "Take a Break!" activity or continue "It's Game Time."

Keeping Track of Things (30 minutes)

Activity Description

In this two-part tutorial, students will learn to create and use functions to simplify their main loop and keep track of the different components of their program. In the first part, *Creating Functions*, students learn how to create a function to help them to organize their programs; in the second part, *Functions and the Counter Step Model*, students learn to use a step counter in their program to ensure all steps happen in the right order. They also get more practice creating functions.

Materials Needed

For each student:

- Computer with Internet connection

Game On (30 minutes)

Activity Description

Conduct the Intramural Competition to select which code will be sent to the practice Regional Competition. Plan to make the Intramural Competition a special event with stickers or awards for all participants. All students should be commended for what they have accomplished so far. Teams should be encouraged to compete even if they have not been able to complete all the game objectives with their code. (If there is only one code from within the group, it is still nice to make this a special milestone event and hold the competition by running against a “standard player” in lieu of another team player.)

Materials Needed

For the group:

- Computer with Internet connection and projection equipment to show the Intramural Competition
- Stickers or awards
- Newsprint or board to record results of the Intramural Competition

❶ Follow the instructions in the *Intramural Game Mode* tutorial to:

a) Show students how to share their game code with the group.

b) Learn how to run a competition between two teams.

2. Invite the two teams competing to briefly describe their strategy and then project the competition between those teams.
3. Assign students to record the scores for both teams on a board or newsprint.
4. Repeat steps 3 and 4 until all teams have played against each other.
5. At the end of the Intramural Competition, select the team with the most wins or the highest total score, to submit their code to the practice Regional Competition.
6. Have all students watch as you follow the instructions in the *How to Submit Code* tutorial to submit the winning code to the practice Regional Competition. Be sure to submit code even if it is not able to meet all the game objectives, as this is not an elimination competition. (It is also a useful exercise to walk through the process before the final code deadline at the end of Week 4.)
7. Tell the students that the results from the practice Regional Competition will be available at the beginning of Week 4.

Take a Break! (varied times)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment

Activity Description

To break up the programming during this week, show videos related to and conduct activities to further explore the concepts students have learned (you might also conduct some of the ice breakers listed in **Appendix 1**). Links to some resources are available in the student materials.

Reflect and Assess

At the end of this week, provide students with opportunities to reflect on what they have learned both this week and during the program so far, and to connect the programming and science and mathematics concepts they have learned about to the ZR game.

Achieving Our Goals	What Do You Know Now?
Display the student learning outcomes for this week, which students can see in the Week 3 student materials. Ask students to reflect on how well they feel they have achieved these outcomes, using the Questions for Reflection of page 6 to guide the reflection. In particular, ask students to reflect on how it feels to work together in teams, and between teams, to solve problems.	Revisit the K-W-L and add to it based on what students have learned and wonder about at this point.
Do You Know More than a 12th Grader?	**Jeopardy!**
What is the difference between mass and weight? (Answer: Mass is the amount of matter in an object. Weight is a measure of the force of gravity on an object?) 1. One iPhone 5S weighs 4 oz. in Boston. How much will 2 iPhone 5S's weigh in California and why? (**Answer**: 8 oz. because there are two of them and their weight is the same everywhere on Earth.) 2. What do speed and velocity have in common? (**Answer**: They both measure how fast a point or object is moving) 3. When you first throw a bowling ball down a bowling alley, is it accelerating or decelerating? (**Answer**: Accelerating) 4. What is the benefit of using functions in programming? (**Answer**: Functions help programmers to simplify and organize their programs.)	Develop a class game of Jeopardy! using the resources in the student materials section of the ZR website or the information included in this guide. You might want to use the following categories:: 1. Kinematics (call it "Kine-what-ics?") 2. Mass vs. Weight (call it "To Weight or Not to Weight?") 3. Speed vs. Velocity (call it "Speed Up!") 4. Programming (call it "Building Your Game")

Week 4:

Going the Distance

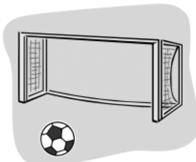
Overview

During the fourth week of this program, students will primarily be completing the programming of their game. However, in order to break up the time spent coding, they will also be spending time exploring the STEM concepts underlying the SPHERES and their game programming through a variety of activities and videos. By the end of Week 4, the group will select a single team's code to be submitted to the Regional Competition; winning codes from each region will then be submitted for preparation for the National Competition.

Getting Started

Before Week 4, you should review the information below, preview the online activities and videos, and gather any other materials needed.

Goals



Students understand how to program their player for the ZR game and are able to create a strategy for successfully accomplishing the task(s) required in the game.

Student Learning Outcomes

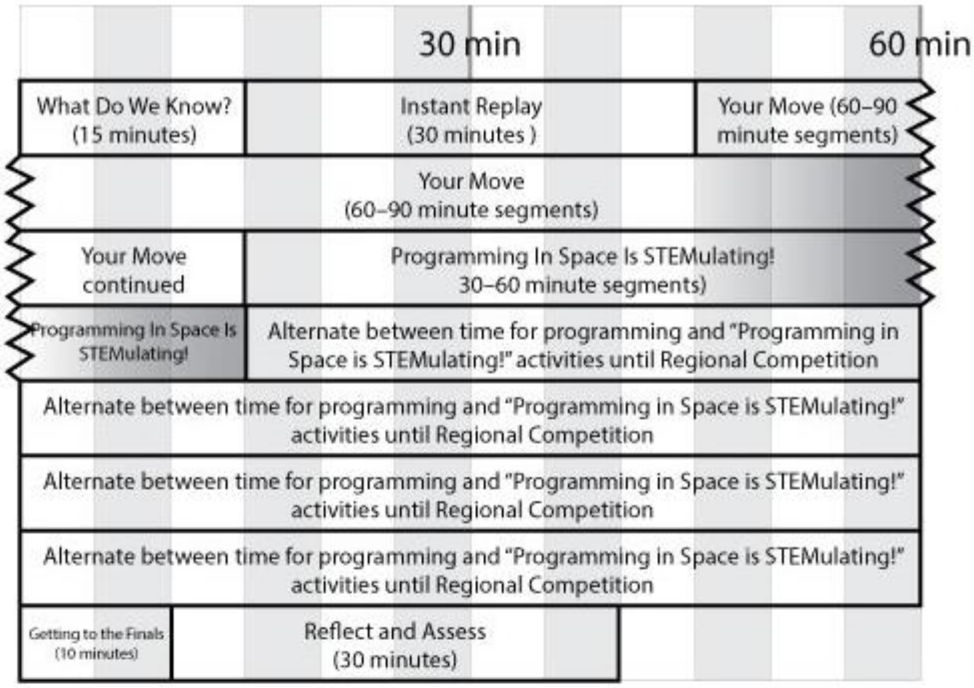
By the end of this week, students should:

- Have reviewed the results of the practice Regional Competition
- Understand and be able to describe their final game strategy
- Complete their programs and select a single entry from the whole group for submission to the Regional Competition
- Be able to work together in teams to solve problems related to their game strategy and coding

Before You Teach

- Review the suggested flow of activities below and estimate what activities will be scheduled for each day based on the time you have available.
- Set up a free account on the PBS LearningMedia website (http://www.pbslearningmedia.org/) to be able to view some of the suggested videos and activities.
- Review the numerous non-programming activities to determine which you will have your students do. You will likely not have time to do them all, so it will help for you to review them all ahead of time and decide which ones will work best with your schedule and for your students.
- Determine which items from the “Reflect and Assess” section you will use.

Suggested Flow of Activities



Activities

What Do You Know? (15 minutes)

Using the reviews provided at the start of Week 2 and 3, do a very brief re-cap of concepts students have learned so far:

- SPHERES and the premise of the ZR game
- The basic components of a program—conditional statements, data types, functions, logic, loops, variables
- How the ZR IDE will be used to create and debug programs
- What SPHERES controls are and how they are used in the ZR game
- Coordinate systems, graphing and Cartesian coordinates and how they are used in the game
- Order of operations and why it matters in mathematics and in programming; how it is similar and different between the two
- Dimensions—the differences between 1-D, 2-D, and 3-D
- Vectors—what they are, how to measure them, and how to combine them
- Forces and motion and how they relate to the SPHERES

Connections to Week 4

Explain that students have learned a lot of different mathematics and physics concepts that explain how SPHERES move and 'act' in space. In Week 3, they also learned some new programming concepts. They will review the results of the practice Regional Competition and decide what changes they will make to their programs this week. Now they really have a chance to put to work all that they have learned in the ZR Middle School Program so far by developing a player for the group.

Instant Replay (30 minutes)

Activity Description

The ZR team will provide instructions at the beginning of Week 4 about how to obtain the scoring results from the practice Regional Competitions. The instructions will include a link and an explanation about how to replay the simulations of each competition.

1. Review the scoring results with your students. Remind students that the results are based on preliminary code from each of the teams—some of the teams may not have submitted code that met all the game objectives. All of the teams will be working to improve their code during the coming week.

2. Analyze simulations from the practice Regional Competition.

   a) Show students all of the competitions involving your code, and several competitions between other groups, especially those that scored well.

   b) For each competition:

      i. Have half of the students watch the Blue SPHERE and half watch the Red SPHERE.

      ii. Have students describe what they think was the strategy of the SPHERE they were watching. Discuss what worked well and what did not work well.

   c) Ask students how they might modify their strategy to improve their score. Record these ideas on posted newsprint or a board and leave visible through Week 4.

   d) Lead the group in developing an outline of the overall group strategy based on their analysis of the practice competition; ask one student to record the outline on the board or posted newsprint.

      i. The strategy might build off the submitted strategy and incorporate some the best

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to replay simulations from practice Regional Competition
- Newsprint or board and markers

components from other team's strategies or be an entirely new one if the group feels they have some new ideas they would like to try.

ii. A few students may be interested in reviewing all of the practice competitions on their own computers while the rest of the group works on programming. Have these students report back to the group with any additional ideas for strategies that worked well and that did not work well.

e) Have the group come up with a name for the entire group—it may be a blend of the team names or something completely new.

Your Move (4–5 hours)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment
- Newsprint or board and markers to record group strategy

Activity Description

Teams improve on the programs they began in Week 3. Remind students that by the end of Week 4, they will need to have completed their programs and selected a single entry for the Regional Competition.

1. Have students code their programs: The group may choose to work together to develop code that supports the new strategy or work in separate teams and see which code is strongest at the end of the week.
   a) Provide newsprint or space on the board and markers for the group or for teams to use to develop and discuss their plans.
   b) If the group works together, consider having the ZR IDE projected so that all students can watch and provide feedback to the program as it is developed.
   c) If the group splits into separate teams:
      i. Have teams share their code with one another at different points during the week to provide suggestions, help look for bugs, and learn from one another's programming.
      ii. Have teams compare the performance of one another's code against standard players (see

Introduction to Game Mode tutorial) or hold another Intramural Competition during the week to determine which code to submit. See "Game On" activity in Week 3 and the Intramural Game Mode tutorial for details about running an Intramural Competition.

2. Students should create code, simulate, debug, and then repeat.

   a) Have students save successful variations of their programs with a new name before they make changes so they can easily go back to previously successful versions if needed.

   b) Encourage students to run their code against "standard players" as described in the *Introduction to Game Mode* tutorial.

   c) Students should set goals for what they should accomplish each day. This will help them keep track of their progress so that they can be sure to have completed their program by the end of the week.

> **Note:** *During this week, you will need to set aside time in the schedule for planning and organization, and for checking in with each team of programmers. As noted previously, only teams with a plan will be successful in the ZR game.*

3. Plan to spend no more than 60–90 minutes on programming at a stretch before having students take a non-programming break using the activities below.

4. At the end of each session of coding, have both students and teams reflect on the ZR programming using the Questions for Reflection in the Program Overview:

   a) How far along are you with developing your code for the Regional Competition?

   b) What challenges are you having?

   c) What are possible solutions that will help you address the challenges?

5. On the last day of the week, have students compile their final program code to submit to the Regional Competition. Invite students to watch as you follow the instructions in the *How to Submit Code* tutorial to submit the group entry to the Regional Competition.

6. Tell students that the results from the Regional Competition will be available at the beginning of Week 5 but, no matter what the results, you are proud of them for what they have accomplished. Invite them to give themselves a rowdy and rousing round of applause!

Programming In Space Is STEMulating! (30–60 minute segments)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment
- Other materials needed depending on activities selected

Activity Description

In order to break-up the time spent programming the game, have students engage in some of these hands-on and online activities to explore and reinforce the connections between programming and the STEM/space exploration content. You may pick and choose between the hands-on activities you do with students and online activities you have students do on their own, depending on the time available.

Space Exploration

- Accidental Discoveries and the associated discussion questions allow students to explore the 1963 Nuclear Test Ban Treaty that led to the launching of several satellites to detect nuclear energy blasts. These satellites ended up detecting blasts from unexpected and random places all over the sky, resulting in the accidental discovery of gamma-ray bursts.
- This interactive History of the Universe timeline of the birth of the universe begins with the "Big Bang" and ends with the "Dark Era". Discussion questions are also included.
- Students can play some of the games available in the student materials, including:

- Astronomy crossword and other games at: http://www.kidsastronomy.com/crossword.htm
- Play a game or take a quiz at NASA's SpacePlace
- Play Space Word-a-rama and other learning games

- This set of career-focused resources might be appropriate for a whole-group review followed by a discussion on what they have learned. You can also explore what additional questions this information raises for them:
  - Atmospheric Science:
  - Computer Science:

STEM and Programming

- Friction:
  - The NASA Friction and Inertia unit includes a number of activities and discussions questions.
  - Fascinating Friction! is a hands-on activity from Teach Engineering that includes a vocabulary guide and additional extension activities.
- Coordinates, Grids & Graphing:
  - The Coordinate Game
  - Billy Bug
  - In this video segment from *Cyberchase,* Using a Coordinate Grid, the CyberSquad have become separated on an island. In order to find each other they bring up a map on their Skwak Pads and try to compare positions using a coordinate grid and landmarks on the map.
- Dimensions: Quantus, a KET animation, shows how everything that we see around us can potentially be described with numbers. A computer does this to display objects, movement, lighting, and color, based upon numerical values—this is the basis of how most video games and animated movies are created! Digital modeling involves concepts such as *x*, *y*, and *z* axes/coordinates and 3-D space.
- Programming:
  - In this Tic-tac-toe player, student teams create sets of instructions (a program) to play the game of tic-

tac-toe. They then participate in a tournament between the different teams' programs to see which plays best (mimicking what students did in the Intramural Competitions).

- In this Create-a-face activity, students make an affective (relating to moods and emotions) robot face out of card, tubes, and themselves. The robot needs to be programmed to react to different kinds of sounds (nasty, nice or sudden) and show different emotions (sad, happy, surprised), thereby requiring students to use conditional statements. Students can then come up with other facial expressions and program sets of rules to have the face react to sounds.
- In this Treasure Hunt activity (Activity 7 on p. 45), students learn that computer programs often need to process a sequence of symbols such as letters or words in a document, or even the text of another computer program. Computer scientists often use a finite-state automaton (FSA) to do this. An FSA follows a set of instructions to see if the computer will recognize the word or string of symbols. In this activity they will be working with something equivalent to an FSA—treasure maps!
- The Orange Game (Activity 6 on p. 40) teaches students about logic and reasoning. When you have a lot of people using one resource (such as cars using roads, or messages being sent through the Internet), there is the possibility of "deadlock." A way of working cooperatively is needed to keep this from happening.
- In Programming a Robot, students can watch another video segment from Cyberchase. The CyberSquad has shrunk Matt and sent him inside Hacker to insert a memory chip that will change Hacker from evil to good. When a force field causes problems, the CyberSquad must program a robot to rescue Matt. And in order to get the robot to do what they want, they must break a task into a sequence of simple steps that the robot can follow.

Getting to the Finals (10 minutes)

Activity Description

Explain to students that despite who wins the Regional Competition, at the beginning of next week, all teams in their region will have an opportunity to collaborate to try to improve the 1st place regional winner's code. Teams from their region can to try to beat the regional winner's code and then share their solution with the regional winner. The regional winner will submit the final code from their region prior to the National ISS Finals Competition code submittal deadline. (Your regional coordinator will provide the date for the National ISS Finals Competition.)

The winning codes from each region will be uploaded to the SPHERES satellites located on the ISS. The ISS Finals competition is not performed in simulation but is conducted by astronauts on the ISS using real SPHERES satellites.

The ISS Finals competition will be broadcast live so that all the teams involved in the ZR Middle School Program can watch and cheer on the code submitted from their region. Tell students that next week they will also be preparing posters and/or videos for the event. Encourage the students to plan to attend this exciting event.

Materials Needed

For each student:

- Computer with Internet connection

Reflect and Assess

At the end of this week, you should provide students with an opportunity to reflect on what they have learned and connect the concepts.

<table>
<tr><th>Achieving Our Goals</th><th>What Do You Know Now?</th></tr>
<tr><td>Display the student learning outcomes for this week, which students can see in the Week 4 student materials. Ask students to reflect on how well they feel they have achieved these outcomes using the Questions for Reflection of page 6 to guide the reflection. In particular, ask students to reflect on how it feels to work together in teams, and between teams, to solve problems, and what benefits and challenges they find in teamwork.</td><td>Revisit the K-W-L and add to it based on what students wonder about at this point.</td></tr>
<tr><th>Do You Know More than a 12th Grader?</th><th>Jeopardy!</th></tr>
<tr><td>Provide students with a quiz on some of the terms and concepts that they re-visited and used this week. Focus the questions on the non-programming activities and discussions that the group did as a whole. Sample questions and suggestions for content are below, based on specific activities:

Name an 'accidental discovery' made by satellites in space. (Answer: They discovered gamma-ray bursts. NOTE: Use this only if you did the 'Accidental Discoveries' Activity.)
The beginning of the universe is known as the Big Bang period. The end of the universe (many millions of years from now!) is referred to as the _____ _____. (Answer: Dark Era. NOTE: Use this only if you had students view the 'History of the Universe' timeline.)
In order to be an atmospheric scientist, you need a degree in Atmospheric Science or _______. (Answer: Meteorology. NOTE: Use this only if you had students review the career resources on 'Atmospheric Science'.)
When members of the CyberSquad team become separated on an island they try to find each comparing positions using a ______ _____and landmarks on a map. (Answer: coordinate grid. NOTE: Use this only if you had students watch the Cyberchase video on 'Using the Coordinate Grid'.)
The Orange Game teaches us about a programming concept we learned back in Week 1 called _______. (Answer: Logic or reasoning. NOTE: Use this only if you did the 'Orange Game' Activity.)
</td><td>Develop a game of Jeopardy! using the Resources section of the ZR website or the information included in this guide. You might want to use the following categories:

Programming (call it "Do Not Compute" and focus on logic and elements of coding including debugging)
Mathematics (call it "How Can I Get There?" and focus on how mathematics helps the SPHERES to move around)
Physics (call it "All the Right Moves" and focus on how forces and motion affect the SPHERES—you might include such answers as "I'm afraid that I do not like how you are facing, young man/lady!" to which students could ask "What is attitude?")
Space (call it "Around the Universe in 80 Days" and focus on, for example, how the universe was formed, and why satellites—of any kind—are launched into space)
</td></tr>
</table>

Week 5:

Reach for the Stars

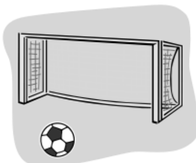
Overview

During the final week of the regular program, in addition to programming for the National Competition students will have one more task to complete for the finals—to highlight their strategy. This is also a time to have some fun with science and engineering—to explore careers, find out more about space science and exploration, and perhaps to go on a field trip to a planetarium or other space or computer science-related site.

Getting Started

Before Week 5, you should review the information below, make plans for any field trips, and select which activities to use.

Goals



Students understand what career pathways exist in STEM and how they impact society, as well as learning some additional content related to the SPHERES.

Student Learning Outcomes

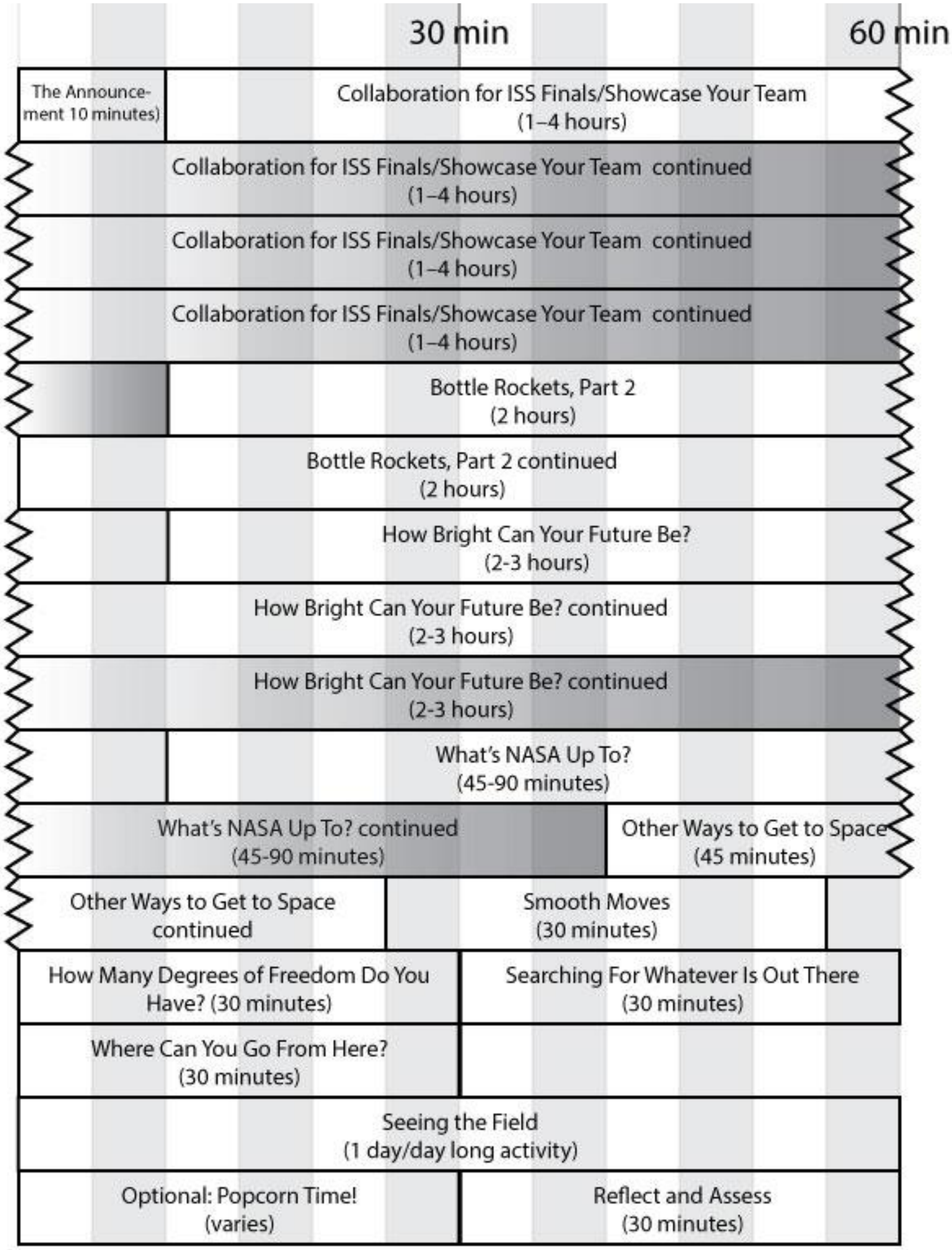
By the end of this week, students should:

- Understand how science and engineering concepts relate to the 'real' world
- Understand more about how SPHERES move, including degrees of freedom and dynamics
- Understand why space exploration is important
- Understand what some careers in STEM are and what skills and knowledge are needed to be successful in those careers

Before You Teach

- Review the suggested flow of activities below and estimate what activities will be scheduled for each day based on the time you have.
- For “How Many Degrees of Freedom Do You Have?,” create a large sign with the title “NASA” on it.
- Determine which items from the “Reflect and Assess” section you will use.
- Prepare a handout to send home to families providing the date, time, and location of the ISS Finals Event and encouraging students to attend this exciting event.

Suggested Flow/Schedule of Activities



Activities

The Announcement (10 minutes)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment

Activity Description

The ZR Team will announce the winners of the Regional Competition at the beginning of Week 5.

1. Review the scoring results with your students.
2. Tell students that whether or not their team is the winner of the Regional competition, all teams have the opportunity this week to collaborate to try to improve on their 1st place regional winner's code prior to ISS code submittal deadline. All teams will be attending the ISS Finals Event to watch the competition conducted by astronauts aboard the International Space Station. Encourage students to plan to show off their state pride and to cheer for their region at the ISS Finals Event.
3. Tell students that their team will get a chance to showcase the work they have done.

Collaboration for ISS Finals

Materials Needed

For each student:
- Computer with Internet connection

Activity Description

Explain to students that all teams are invited to collaborate to try to improve on the regional winner's code in their region prior to the ISS submittal deadline.

1. The ZR team will share the code submitted by the regional winner's team with all the teams in their region to give them the opportunity both to see the winning code and to run intramural competitions against their code.
2. Encourage students to work together to write code to try to beat the regional winner's code. They can either work to:

   a) Improve on the code they submitted in Week 4

b) Improve on the code submitted by the regional winner's team

3. Run an intramural competition between your student's new code and the regional winners code.

4. If the student's new code beats the regional winner's code share the student's new code with the 1st place team from the regional competition.

5. Tell students that by 5 pm on Thursday of this week, the regional winner must submit the final code from their region to the National ISS Competition.

6. Tell students that no matter what the results, you are proud of them for what they have accomplished!

Showcase Your Team (1–4 hours)

Activity Description

Now that the final codes have been submitted for the ISS Finals, it is time for every site to highlight their group, region, and game strategy. Posters or videos made by the group will be shared with the other groups during the ISS Finals Event.

1. Have students make posters with your group name and highlighting your region.

2. Create a poster or a video describing the group's strategy. Videos should be no longer than forty-five seconds in length.

3. Give students a handout describing the date, time, and location of the ISS Finals Event. Encourage them to attend this exciting event.

Materials Needed

For each student:

- Handout to send home to families providing the date, time, and location of the ISS Finals Event

For the group:

- Poster boards
- Markers
- Video equipment or cell phones
- Computer with Internet connection and projection equipment

Bottle Rockets, Part 2 (2 hours)

Materials Needed

For each team:

- Bottle rockets from Week 1
- Computer with Internet connection
- Cloth, wood, duct tape, and a variety of other materials students might use for their designs
- *Optional*: 1–2 eggs

Activity Description

Now that students have explored the physics that affect the SPHERES a bit, they can put some of this new knowledge into practice and try to improve their bottle rockets from Week 1.

1. First, allow teams from the "Bottle Rockets, Part 1" activity (in Week 1) to research and re-design their rockets, building on what they have learned during the program.
2. As an optional additional activity, have each team design a carrier to hold an egg and redesign their bottle rockets to safely land it.

How Bright Can Your Future Be? (2–3 hours)

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Activity Description

In this activity, students will explore a variety of career information resources. The goal is to structure their exploration in such way that students are making connections between their interests, programming and possible space careers, learning about specific careers, looking for additional information on specific careers, identifying skills needed for success in those careers, and sharing what they have learned with one another.

1. Have students watch this video, What Most Schools Don't Teach, about programming and the many fields in which it is used. Lead a group discussion using the following questions:

   a) Why should 'everyone learn how to program a computer' as this video says?

   b) Do you recognize any of the people in this video? If so, how do you know who they are?

c) How old did the people in the video say they were when they started programming? What did some of their first programs do?

d) What is one thing that surprised you most in this video?

e) How does this video make you feel about, and think about, programming?

2. Have students take a career interest inventory, such as one of the resources included in the student materials. Inventories are online (or paper) surveys that have students go through long lists that describe what they are like or what they like to do. After they have checked off all the options that they agree with, the survey will present them with a series of careers or career areas to explore further. It is ok for this activity to seem broad at first—the idea is to get students thinking about what they are like, what skills they have, what they like to do and making the connection between those things and possible careers. After students take the survey(s), lead a group discussion using the following questions.

   a) What kinds of thing did you choose that interested you?

   b) What career areas did the survey show you?

   c) What surprised you about the survey?

   d) What did you learn about:

      i. Yourself?

      ii. One possible career or career area for you?

3. Have students explore space-related careers using the resources provided in the student materials.

4. Have students identify a subset of careers that interest them based on what they have done so far. Have them research these careers using the resources provided in the student materials and then prepare a paragraph or two that answers the following questions about *two* of the careers that interest them most:

   a) What interests you about this career? What do you dislike about it, if anything?

b) What skills do you have that relate to this career?

c) What skills do you need to be successful in this career?

d) What education and training do you need to do this career?

e) What classes in middle school (and high school) will help you prepare for this career?

5. Have students use what they have learned in this activity to create a short presentation on the two careers they have researched. The presentation should be no more than 10 minutes in length and students can choose to read from something they write up, create a PowerPoint presentation, create mock-ups with paper and pencil, and so on. The presentation should have same categories as the five questions they answered in step 3, namely:

a) What interested them about the career

b) Skills needed for success in this career (and how their skills match these)

c) Education requirements

d) Middle and high school classes they need to take

What Is NASA Up To? (45–90 minutes)

Materials Needed

For each student:
- Computer with Internet connection

For the group:
- Computer with Internet connection and projection equipment

Activity Description

At this point, students may be interested in learning more about NASA—what they do, what we've learned from their work, and what they are hoping to do in the future. Now is the perfect time to look more at NASA in the past, today, and in the future.

1. Review current and planned NASA missions for 2013 and beyond using the following resources:

a) Current Missions

b) Future Missions

2. What happens when things go wrong in space? Watch

this clip from the movie *Apollo 13*: "Houston, we have a problem!"

3. What have we learned from space exploration? Have students review the resources in the student materials to learn what knowledge and information has been gained from space exploration over the years.

> ***Note:*** *In Rising Above the Problem, students are introduced to Earth-orbiting satellites that give us not only a 'big' picture of where rivers flow and mountains stand, but also allows gives us detailed, sharp images of the earth's surface. To introduce this activity, you could show this CloudSat Image of Hurricane Sandy.*

4. What were some of the important space missions and what did we learn from them? Have students review information on the history topics. Then have students do a Q&A with each other where they test each other's knowledge.

5. Have students spend more time looking at and learning about the ISS using the resources in the student materials. Use this as an opportunity to remind them that the regional winners' codes will be run on the ISS, and they will watching it happening live next week!

6. *Optional:* Have students make a video showing what NASA is doing. They can browse the clips available at http://www.nasa.gov/audience/foreducators/diypodcast/index.html (the right side of the webpage includes short video clips, audio clips, and images that students can include in their video) and make a short video that incorporates the clip(s) of their choice.

Other Ways to Get to Space (45 minutes)

Activity Description

Students learn about private and commercial space travel using the resources in the student materials.

Materials Needed

For each team:
- Computer with Internet connection

1. Split the group into small teams to investigate one of

the companies listed in the student materials (or another company of their choosing) for 30 minutes to find a product or service the company is offering (or hoping to offer in the future).

2. Bring students back together and have each team pitch one of the company's products or ideas to the group.

Smooth Moves (30 minutes)

Materials Needed

For the group:

- Computer with Internet connection and projection equipment to show the **Smooth Moves** PowerPoint presentation

Activity Description

Now that students have an understanding of kinematics, introduce them to the concept of *dynamics*. Dynamics, which is the study of motion and changes in motion, deals with any kind of object, but for our purposes, we will look only at *rigid body dynamics*.

1. Refer students back to the dry ice activity. Explain that the activity showed *frictionless dynamics*—or how objects move when they aren't subjected to friction.

2. Explain rigid body dynamics.

   a) Ask students what 'rigid' means. Can they identify the opposite of rigid? Can they name one object that is rigid and one that is not? Tell students that a rigid body is just what it sounds like. It's not bendy or floppy like rubber, just stiff. Any object whose shape can't be changed, regardless of the forces exerted on it, can be classified as a rigid body.

   b) Give students the following example:

   Suppose you are pushing a crate to a distance five feet away from you. Every particle of the crate, from top to bottom and side to side, is moving the same distance. Therefore, the crate is a rigid body. Water, on the other hand, spills and slides around—its particles are constantly swirling because it's a fluid and definitely not rigid.

   c) Finally, explain that dynamics, the study of motion and changes in motion, deals with any kind of object—including both the crate and water, but for our purposes, we will look only at rigid body dynamics.

3. Explore the motion of SPHERES.

a) Explain that for a single point in 3-D space, using three coordinates to define its position is a simple task. Ask students, “If we look at an object with length, width, and height (like the SPHERES) instead, could we use the same coordinates to find its position?”

b) Show slide 2 in the Smooth Moves PowerPoint presentation and ask students, “Imagine that this cube—which has six identical faces—is sitting on a floor. If we spin it 90 degrees (a quarter of a circle), what would its final position be like?”

c) Show slide 3 of the cube’s final position, which looks the same as its initial position. Explain that this can be problematic for the SPHERES satellites. For example, only one face has a docking port—if we can't determine whether the right face is pointing towards the dock, the satellite could just bounce away and float into space. To fix this problem, we have to look at another piece of information: rotation.

d) Explain to students that they are familiar with *translation*, which describes the simple, straight movement of an object forward, sideways, or upwards. Movement really only occurs along the x, y, and z directions.

   Rotation, on the other hand, occurs around one specific point in the body. When we are assessing the body’s orientation, or *attitude*, we are looking at how much it has rotated—not how much it has moved.

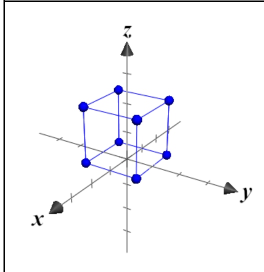
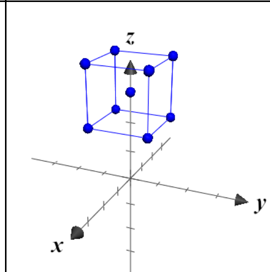
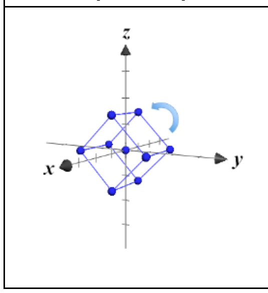
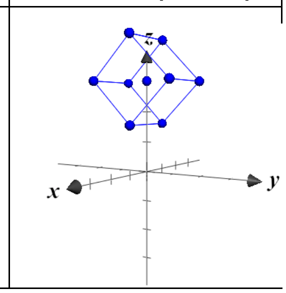
e) Show slide 4 in the Smooth Moves presentation. Tell students that they are looking at an asteroid in space and give the following scenario:

   This asteroid is moving along its central point, hurtling at blinding speed towards Earth. Along the way, it knocks into another asteroid, veers off track (thankfully!), and begins spinning about its center.

f) Ask students to describe the asteroid's current motion in terms of translation and rotation. (Remind them that rotation and translation are two separate motions, although they may occur at the same time.)

g) Now, students need is to understand translation and rotation on a graph-see Table 5.1 below for an overview of steps i–iv.

Table 5.1

Beginning (slide 5)	Translation (slide 6)
	
Rotation (slide 7)	**Translation and Rotation (slide 8)**
	

iii. Show slide 5 in the Smooth Moves presentation, which shows the cube from earlier.

iv. Explain that movement in terms of translation is just a matter of changing x, y, and z coordinates, as shown in slide 6. Ask students how the cube has moved in this slide. (Answer: The cube moved two units in the +z direction.)

v. In slide 7, the cube is back at the origin and has rotated 45 degrees around the x-axis.

vi. Note: In 3-D, the cube can rotate in three ways: around the x-axis, y-axis, and z-axis.

vii. Translation and rotation are combined in slide 8.

viii. Ask students: Can you see how the two motions are entirely different, despite whether or not they occur at the same time?

How Many Degrees of Freedom Do You Have? (30 minutes)

Activity Description

This "Simon Says" activity demonstrates degrees of freedom and the difference between how SPHERES can act in 2-dimensions (on the flat floor) and in 3-dimensions (in space).

Materials Needed

For the group:
- Newsprint with "NASA" written on it, prepared in Before You Teach

❶ Have students spread out in a grassy area or a large room with "NASA" (you!) at the front.

❷ Explain that students will be moving like the SPHERES as directed by NASA—but they can only act when you begin an instruction with "NASA Says . . . ".

a) Explain that first they will be doing the 2-D moves (like the SPHERES do on the flat floor on Earth, where they are affected by gravity).

b) Then they will be doing more advanced 3D moves (like the SPHERES on the ISS, where they are not affected by gravity).

❸ Explain the 2-D moves with students: there are six basic moves (each move represents a degree of freedom) in 2D. After explaining all of the 2-D moves, call out random moves (using "NASA Says . . ." before each instruction) and see if students can follow along.

a) y-axis (translational):

i. plus y (forward step)

ii. minus y (backward step)

b) x-axis (translational):

iii. plus x (side step to the right)

iv. minus x (side step to the left)

c) z-axis (rotational):

v. spin to the left

vi. spin to the right

4. Explain the six additional 3-D moves with students. After explaining all of the 3-D moves, call out random moves (using "NASA says . . .") and see if students can follow along.

d) y-axis (rotational):

vii. log roll to the left

viii. log roll to the right

e) x-axis (rotational):

ix. forward roll

x. backward roll

f) z-axis (translational):

xi. jump up and down (going up is positive, coming down is negative)

5. After students have learned all the moves, play a game of "NASA Says..." calling out the steps in a random order and sometimes not saying "NASA Says..." before giving the instruction.

> ***Note:*** *This activity is probably best done outside in a grassy area or on soft ground somewhere (so that forward and backward rolls don't hurt). Some kids may not want to roll on the ground. They can represent 2-D SPHERES only, while those who are willing to do rolls can represent 3-D SPHERES.*

Searching for Whatever Is Out There (30 minutes)

Activity Description

The fascinating work of planetary exploration deserves more attention. In the fall of 2012, NASA landed the Mars rover *Curiosity*. This incredible engineering feat captured the attention of much of the nation—and the world. It may be difficult for students to imagine, but with the resources provided in the student materials, they can try. In addition, you might want to:

1. Walk students through the Mars Rover Kids website.
2. Show additional videos about the Mars rover.

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Where Can You Go From Here? (30 minutes)

Activity Description

The end of this five-week program does not mean that students' fun with STEM and programming needs to end. Students may choose to continue in the ZR program through middle school and even into high school, but there are also a number of other programming and robotics competitions and programs they can explore. Use this time to share information about some of these options with your students, using the resources provided in the student materials.

Materials Needed

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Seeing the Field (1 day)

At this stage of the program, it is highly recommended that you consider another field trip to a planetarium, space, or science center in your area. Planetariums, which may be either stand-alone or permanent installations within science museums or science centers, offer educational and interactive shows and information about astronomy and celestial navigation. If this is not an option, check with your local library or art/history museum for information on space-related exhibits that are current or upcoming. Alternatively, any STEM related field trip (e.g., to local a STEM/space/aviation manufacturing company) would help reinforce connections between what students have learned thus far and real-world applications of the concepts.

Popcorn Time! (depends on selection)

Materials Needed

For the group:

- Computer with Internet connection and projection equipment or television and DVD player

Activity Description

Show students one of the following full-length movies:

- *October Sky* (1999)
- *Apollo 13* (1995)
- *Armageddon* (1998)—note that this movie is PG-13
- *From the Earth to the Moon* miniseries (1998)
- *IMAX Hubble 3D* (2010)
- IMAX *Space Station 3D* (2002)
- IMAX *Magnificent Desolation: Walking on the Moon 3D* (2005)
- *In the Shadow of the Moon* (2007)

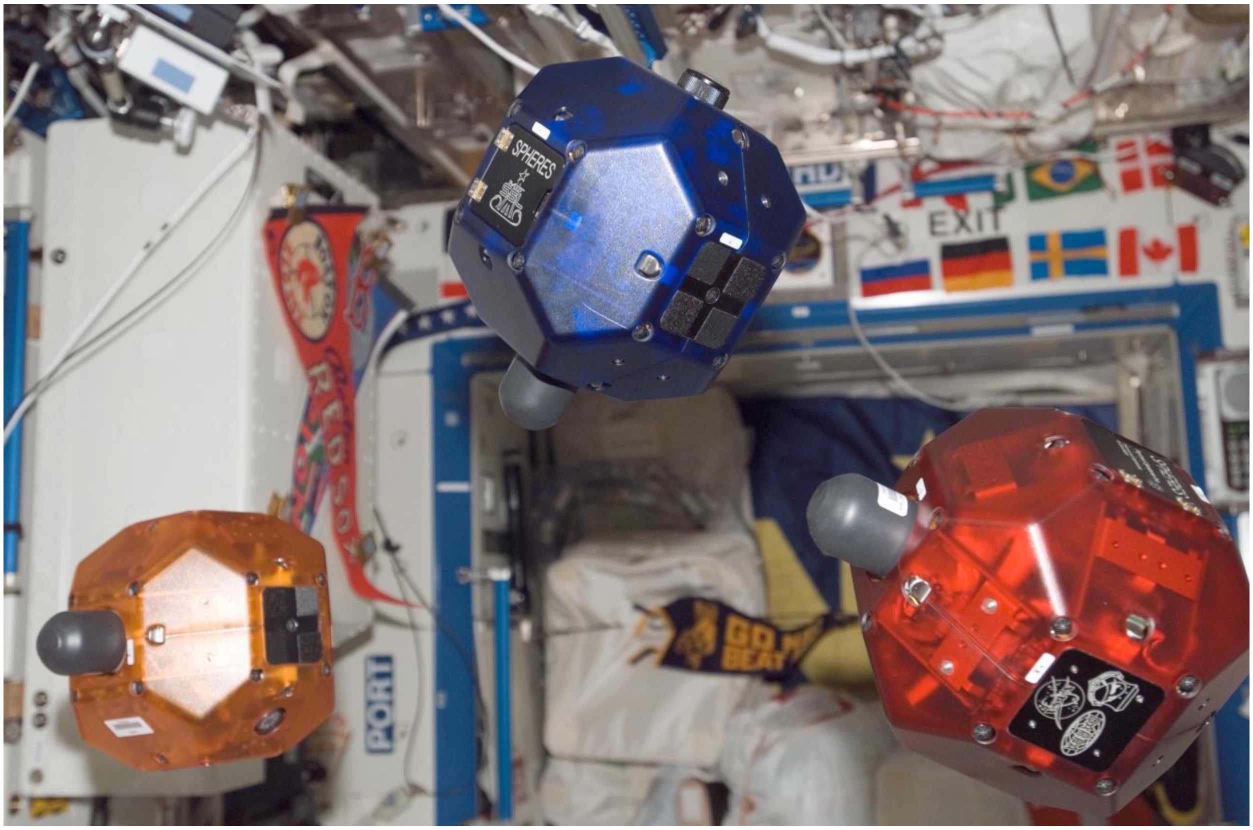
Reflect and Assess

At the end of this week, provide students with the opportunity to reflect on what they have learned during the week and throughout the program.

1. First review the K-W-L and fill in what students have learned "L".
2. Display the student learning outcomes for this week, which students can see in the Week 5 student materials. Ask students to reflect on how well they feel they have achieved these outcomes.
3. Then ask students to reflect on their experience overall using the following questions to guide the discussion:
   a) What did you learn during the program?
   b) How do you know you learned it?
   c) What got in the way of your learning? What helped your learning?
   d) How do you feel about the ZR program and your participation in it?
   e) What goals do you have for yourself due to this experience?

CHAMPIONSHIP!

Finally the code submitted from each region will have the chance to compete! All students in the program are invited to watch a live broadcast of the ISS Finals Event as astronauts aboard the International Space Station conduct the ISS Finals competition using student programs loaded onto SPHERES satellites!



Background Information

List of included topics

Conditional statements

You may know conditional statements described as "if, then" statements. These statements describe the behavior of the computer in response to various situations. The computer makes decisions based on what's happening in a program. However, since computers are not intelligent, they must be told exactly what to do in every situation.

Real-Life Example: You are going to make a pie, for which you need exactly five apples. You need to decide if you need to go to the grocery store to buy more apples or not. Consider the following situations, and describe what you would do in each situation:

- You have 1 apple: *Go to the grocery store to buy 4 more apples.*
- You have 5 apples: *Do not go to the grocery store*
- You have 8 apples: *Do not go to the grocery store*
- You have –3 apples: *What? You must have made a mistake! Negative apples make no sense.*

Now, let's rewrite the problem using words that a computer would understand. The words "IF", "ELSE IF", and "ELSE" are used in programming.

- IF you have less than 0 apples: *then you have made a mistake and should count them again*

- ELSE IF you have less than 5 apples (AND the IF was not executed, so you have at least 0 apples): *then you need to buy more apples*
- ELSE (if the above are false, you must have 5 or more apples): *then you don't need to buy more apples*

We'll rewrite this problem one more time, with the special C language format. Now "a" will represent the number of apples you have.

```
if (a < 0)
        {
        Count again
        }
        else if (a < 5)
        {
        Buy more apples
        }
        else
        {
        Don't buy more apples
        }
```

Note that only one of the statements in the IF-ELSE block is executed. The first statement in the list whose condition is true will be executed; if none are true, the ELSE will execute.

Data types (see also Variables)

In computer science and programming there are different *data types* that programmers use to represent the mathematical information they have on their computer. These data structures include *integers, unsigned integers, float numbers and arrays*.

The data type is vital to how a computer will recognize the data. If you declare two pieces of data that have the same name, but are different data types (e.g., one is an integer and the other a float), the computer will not think that they are the same.

Integers

An *integer* in programming is the same as it is in mathematics. An integer is any whole number, positive or negative, including the number 0. You can store very large numbers in integers, but integers are NOT allowed to have any decimals. In programming, integers are important in bookkeeping, storage of information, and use in conditional statements. For example, the data output by a counter which counts up or down using whole numbers represents integer type data.

Example: int counter;

Unsigned integers

An *unsigned integer* is a subclass of integers, so they act very similarly. The only difference is that unsigned integers, as the name suggests, cannot be negative. Therefore, unsigned integers can be positive or 0. For example, the data output by a counter which only counts up using whole numbers from 0 represents unsigned integer type data.

Example: unsigned int counter;

Float numbers

Float numbers consist of any number, either positive or negative, that has at least one digit to the right of the decimal point. *Float numbers* consist of any real number, either positive or negative. In C, single-precision floats (they type used by SPHERES) are denoted by an f after the number. Examples of floats are 1.1f, 2.0f, –5.1111f, and 3.69f. Just as with integers, the limits on the size of numbers that can be stored in floats are very large. Floats are more accurate than integers are because the extra decimal places allow for more precision. Floats are often used in programming as data or the result of calculations you perform. In Zero Robotics, the satellite position is represented by float numbers because its *x*, *y*, *z* coordinates are described with decimal places.

IMPORTANT NOTES: If you want to use a float value that happens to be an integer, put a decimal point and 0f after it. The number two as a float would be 2.0f, not 2. This is especially important when dividing numbers, because you want to make sure you use float division rather than integer division. In C code, 3/2 is 1 because the remainder is truncated when integers are divided; however, 3.0f/2.0f is 1.5f. Also, be warned that due to limited precision, 2.0f will likely actually be stored as something like 1.999999f or 2.000001f, so you should NEVER use the logical == operator with floats, only < and >.

Example: float myPosition[3];

Arrays

A data type that stores multiple pieces of data in one place. Essentially, an array is a list of other data types. An array can hold as many pieces of data as needed.

Examples:

- {1, 2} is an array of two integers (similar to an (*x*,*y*) coordinate point)
- {1.2, 3.0, –2.5} is an array of three floats (similar to a three-dimensional vector)
- {99, 95, 82, 90, 76, 91, 93, 85, 100, 65} is an array of 10 integers (similar to a set of test scores)

- { {1,2},{1,4},{2,4},{2,2} } is an array of two-integer arrays (similar to a set of (*x*,*y*) coordinate points)

Note that declaring an array follows a slightly different format than declaring another data type.

Example: To declare an array called "myPosition" that holds a list of three values you would write the following:

```
float myPosition[3];
```

Each value can be assigned individually:

```
float myPosition[3];
  myPosition[0] = 0.5;
  myPosition[1] = 0.7;
  myPosition[2] = 0.8;
```

The numbers [0], [1], and [2] after the name are called the array indices; they indicate the 3 individual float variables in the array.

NOTE: The declaration indicates a size of [3], so why are the values for the array [0], [1], and [2]? Since computers start counting the array index at [0], an array of size 3 ends at index [2], but you need declare an array of size [3] to accommodate the three values!

Debugging

This is the process used in computer science of figuring out why your solution did not work. A computer does what you tell it to do—no more, no less. Working backwards is important for debugging: you know what *should have* happened and what *did* happen. You can work backwards to find out why your program produced the wrong result.

Forces

A force is an interaction between any two objects; likewise, it can be considered a push or pull on an object. Forces can be the result of direct contact, such as pushing a shopping cart through a grocery store or pulling on a rope during tug-of-war. A force can also act at a distance, such as gravity.

Forces cause acceleration in an object. When you push a cart, the cart accelerates from rest (zero velocity) to some final speed. However, forces can also be present when nothing is moving, since the forces can cancel each other out.

Aerodynamic forces (lift and drag)

Mechanical forces due to the interaction between an object and a fluid (gas—such as air—or liquid).

Friction
Friction is caused by two objects rubbing against each other. A frictional force is always exerted in a direction opposite of the motion of the object. This causes the object to slow down and eventually stop. Although friction is present everywhere on Earth, friction is not present for an object floating in space.

Gravity
Gravity is a constant force in our lives. Everywhere on Earth, no matter what we are doing, gravity is acting on us, pulling us toward the Earth.

Normal force
A force exerted by a surface that, for example, helps push things back up when gravity is pulling them down.

Thrust
A force generated by propulsion to move an object through air or space. Created by causing a fluid (air for airplanes and fuel for rockets) to move opposite the direction of travel.

Functions

Programmers find it easier to break up their code into separate sections that perform different tasks. A programmer puts a set of instructions for a particular task into a *function*. This function can then be "called" and used in their program repeatedly.

The SPHERES Control Functions and Game Functions interface to the SPHERES control systems. Students can incorporate the SPHERES Control Functions into their programs as needed to perform certain tasks.

The ZR Middle School Program curriculum teaches students how to create procedural functions. Procedural functions allow students to split their programs into different sections to help keep their programs organized. Student-created functions are created as separate pages in their programs. Rather than repeating duplicate code multiple times, students can also create functions that they "call" repeatedly in their programs.

The curriculum does not teach students how to create functions that include arguments with return values. Many of the built-in functions return a value, such as an int, float, or boolean (true/false) value; these can be used in code as if they were variables with these types.

Graphing

To find a point in an *x*-*y* plane, you need to find out how far along the point is on both the *x* and *y* lines. These lines are the measuring sticks we use to locate our points. We refer to them as the *x*-axis and *y*-axis.

The coordinates of a point describe its exact location on the graph. The measurement along the *x*-axis is called the *x* coordinate. For the *y*-axis, it's the *y* coordinate.

Kinematics

The study of motion without consideration of the forces that cause the motion. Kinematics considers only the position, speed, and acceleration or deceleration of an object.

Loops

Loops allow computers to carry out the same instructions over and over again, but only require the programmer to write the instructions once. Thus, loops simplify computer programs. In Zero Robotics, we use the **for loop**. **For loops** exist in some form in all programming languages.

For Loop Example: Start with a variable, *x*. We will say that *x* = 0. We want to add 1 to *x* until *x* = 3.

We write a loop so that the computer will change the initial value, 0, by adding 1. Each time the computer adds 1, it will check to see if the new value of *x* is 3 and stop repeating the instruction when it is 3. Here is an example of the computer's thought process:

Beginning value of x: 0

First time through loop: Is 0 equal to 3? No, then increase value of x by 1: 0 + 1 = 1

New value of x: 1

Second time through loop: Is 1 equal to 3? No, then increase value of x by 1: 1 + 1 = 2

New value of x: 2

Third time through loop: Is 2 equal to 3? No, then increase value of x by 1: 2 + 1 = 3

New value of x: 3

Fourth time through loop: Is 3 equal to 3? Yes, so the program is finished!

Instead of the programmer writing x + 1 three times, the loop allows the programmer to write one set of instructions. For such a short program, like the one above, it might seem like it doesn't matter, but with longer and more complicated programs, loops become invaluable.

Now, we will write a loop in C syntax, the language in which the computer will actually read the loop. In C, the general structure for a **for** loop looks something like this:

for (starting condition; repeat condition; update)

```
{
code to repeat
}
```

For the program we wrote earlier, the **for** loop would look like this:

```
for (x = 0; x != 3; x = x + 1)
{
code to display the value of x;*
}
```

However, for this code to work, we would have to define the variable x as an integer at the beginning of our code, so the entire code would look like this:

```
int x;
for (x = 0; x != 3; x = x + 1)
{
code to display the value of x;††
}
```

In the ZR Graphical IDE, the line "int x;" used to create (declare) the variable x is replaced by the New Variable dialog.

SPHERES Example

For loops are often important in SPHERES, as they help define new vectors. Vectors are needed to define how the satellite will move, spin, and rotate. Let's say we have been given a vector that points to an asteroid called Opulens. We want to go halfway to Opulens in order to revolve around it. We can't just type in our code "go halfway to Opulens"—we need to define a new vector that is halfway to Opulens, and then tell the satellite to go there. This is done using a **for loop:**

```
float halfOpulens[3];
int x;
for (x=0, x!=3, x=x+1)
halfOpulens[x]=.5f*Opulens[x];
```

In this **for loop**, the computer starts x at 0. Then for each point in the array less than 3, it divides the number in Opulens in half, and substitutes that in the appropriate spot in the halfOpulens array. Arrays are how the ZR IDE represents the vectors that the SPHERES satellites use. It is important to note that for a three dimensional vector (for

[††] For simplicity's sake, this statement is presented in English and not C code.

example, [a,b,c]), a is in the 0 spot, b is in the 1 spot, and c is in the 2 spot. So even though there are three parts to the vector, the computer only performs the loop up to where x < 3, NOT when x == 3.

If we then wanted our satellite to head towards our new vector, we would simply enter the line of code:

```
ZRSetPositionTarget(halfOpulens);
```

Thus, we have successfully programmed the satellite to go halfway toward the Opulens asteroid.

Motion

In physics, motion is a change in the position of an object. Motion can be described in terms of displacement, velocity, acceleration, and time. Dynamics is the study of motion and changes in motion.

Newton's Laws

Sir Isaac Newton is one of the most celebrated physicists of all time. Several scientific discoveries commonly attributed to Newton include the law of universal gravitation, the law of cooling (describes how the temperature of a cup of coffee cools off in the surrounding air), and the development of calculus. Newton's laws of motion are three physical laws that relate forces acting on an object to the motion that the forces cause. The standard unit for a force is even named after him: forces are generally measured in newtons.

First Law (The Law of Inertia): An object at rest remains at rest until acted on by an outside force; an object in motion remains in motion until acted on by an outside force.

What it means: If the total force on a body is zero, the body's motion does not change. For example, a soccer ball sitting still in the grass will stay still unless someone kicks it. That soccer ball will keep rolling until a force slows it down and stops it.

Second Law: Force = mass × acceleration ($F = m \times a$)

What it means: A body undergoes acceleration when it is subject to a total force that is not zero. Therefore, if you know an object's mass and its acceleration, you can figure out the force exerted on it. Likewise, if you know the force exerted on an object and its acceleration, you can find out the mass of the object.

Third Law: For every action, there is an equal and opposite counter-reaction.

What it means: If you exert a force on something, that force will exert a force of the same magnitude in an opposite direction on you.

Variables (see also Data Types)

Unlike in mathematics, in programming a variable is a container for data. Before you use a variable in your program you must tell the computer what type of data you will be putting in it. Attempting to put data of the wrong type into a variable will cause an error message or undesired behavior.

More examples of variables:

- Declare the variable x as a container for floats and then store the value 5.5 in it:

```
float x;
x=5.5f;
```

- Declare the variable y as a container for unsigned integers and store the value of 10 in it:

```
unsigned int y;
y=10;
```

- Writing the following will cause an error because you haven't declared what type of data you will be putting into the variable y:

```
y=–7;
```

By assigning a value to a variable, you *initialize* it. The variables can then be reused. For example, you can declare z as a container for integers and then assign it the value 3:

```
int z;
z=3;
```

Later, you can increase the value of z by 2 (making it 5) by writing:

```
z=z+2;
```

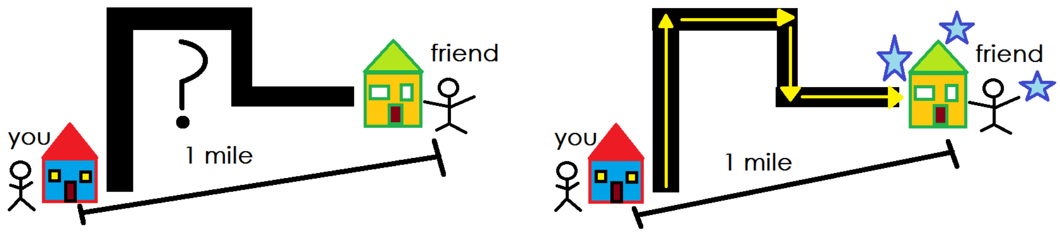
Students will use two variable types most often. They are integer and floating-point number (both are also explained in the *Getting to Know the ZR IDE* tutorial). An **integer** (**int**) is a whole number, positive or negative, including the number 0. Integers are NOT allowed to have decimals. Examples of integers include 0, 1, 2, –1, –2, –3. A **floating-point number** (**float**) is a number, either positive or negative, that has at least one digit after the decimal. Floats allow for decimal values. Numbers should end with f to show that they are float values. Examples of floating point numbers include the following: 1.1f, 2.0f, 3.69f, –5.111111f

Vectors

Directions are an important part of getting around in our world. When you tell a friend how to get to your house, you don't just tell them how far they have to walk—you also tell them what direction to go. Without direction, your friend could end up in a completely different area of town. Not only is direction important in getting to people's houses—it is also important in the world of math and physics. Vectors are used to describe direction and magnitude in mathematics. They are necessary to describe motion, forces, and movement in the grid in the ZR game.

Vectors vs. Scalars

Measurements are an essential part of math. Measurements are made in centimeters, grams, feet, liters, and many other metrics. However, all of these measurements are scalar measurements—they only refer to a magnitude, not a direction. Vectors are measurements that involve both magnitude and direction. Vectors can consist of a scalar and a direction—such as 30 meters north. However, without the direction, the 30 meters is once again a scalar. And without the distance, north is just a direction. Put them together, and you have a vector.



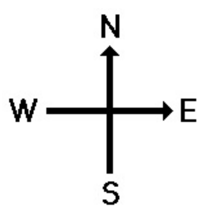
Scalar Vector

Measuring Vectors

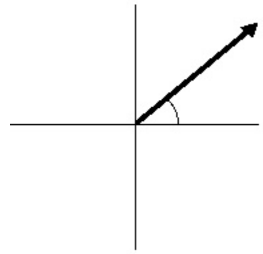
Vectors are most often pictured on the coordinate plane. There are two ways most commonly used to measure and define vectors on a coordinate plane. The first is by <u>degrees</u>, which is how we measure vectors in everyday life. It helps give a direction in a usable format to someone walking on the ground or driving in a car. The second way, commonly used in mathematics, defines the <u>final point</u> of the vector if its 'tail' is at the origin (0,0). While this second method does not explicitly state a direction, it is defined by the direction that the vector would be pointing toward in order to reach the point that is defined.

<u>Degrees</u>

A vector in the coordinate plane can be measured by how many degrees of separation it is from one of the coordinate axes. This is often simpler if we picture the coordinate axes as a compass with the arrows pointing North, West, South, and East. Any vector can then be measured by how many degrees it is separated from a specific direction.



For example, the vector below is “separated” from East by 40 degrees in the counter-clockwise direction:



You could also say that the vector is rotated 50 degrees clockwise from North. However, most vectors measured in this manner are measured counter-clockwise from the East so that all measurements can be compared. If we always choose a different direction to measure our vector from, it’s hard to see the relationships between the vectors. By establishing a common place to measure from, relationships between vectors can be more easily established.

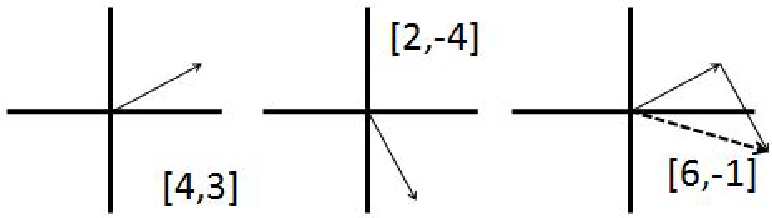
Final Point

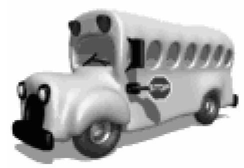
In mathematics, a vector is defined and measured by the final point it reaches if its tail is at the origin. This point is described by how far along it is on each axis, and putting this between brackets, as such: [4,3]. A vector defined by its ending point still has a magnitude and direction. The magnitude is how long the vector is, and the direction is where it is pointing. We can leave the vector defined by its ending point.

Combining Vectors

While vectors can be useful by themselves, often it is best to use them in combination with other vectors. Very rarely are travel paths in a straight line. More often, they involve several twists and turns, which in turn means that travelers use several different vectors in order to reach their final destination—with each vector having its own magnitude and direction. By adding the vectors together, it is possible to tell where the final spot is in relation to the beginning point. It is also possible to find the vector that covers the most direct path between the points.

Vector addition is easiest to picture by placing the tail of each vector at the arrow of the previous vector. Keep adding vectors in sequence until you’re done, then draw a vector that starts at the beginning point and heads toward the end point. You can easily calculate the end point by adding the components of the vectors you’re combining. For example, if there are two vectors, [4, 3] and [2, –4], the sum of the two would be [6, –1]. The x components would sum 4 + 2 = 6, and the y components would sum 3 + –4 = –1.


[4,3]
[2,-4]
[6,-1]



Possible Field Trips and Speakers

The following are suggestions for field trips and speakers that you might want to include in the ZR program. Possible questions for students to pose are included for field trips and speakers; you should also guide students in creating their own questions by introducing the event and its purpose, and having them brainstorm questions in pairs, then share with the group.

1. FIELD DAY: Field Day is a chance for students to act out the ZR game using their own bodies as their player and to think about their strategy. This event should take place at the beginning of Week 2 of the program and should bring all the sites from across the region together at a central location. However, if that cannot happen, the regional ZR team will try to bring several program sites together to hold various competitions.

   Given that intramural competitions will occur first, each group of students will break into teams for their intramural competitions at the beginning of Field Day. Students will work in their teams and within their entire site group to brainstorm strategies for intra- and inter-site competitions. In addition to strategizing and acting out the game in these various competitions, Field Day may provide additional activities for students to study the physics of the SPHERES.

2. For speakers who participate in the ZR program, you may want to provide some possible questions to help them prepare what they will say to the students. In addition, you would also ask students to develop a set of questions about things they would most like to learn about the speaker's job. The questions below are possible options for either scenario:
   - What is your job? What is a typical day like doing your job?
   - What skills and knowledge are important in your job?
   - How did you get where you are today (courses in high school, college major, other activities, career path)?
   - Did you always know you wanted to do this job? If so, why? If not, why not and what else did you consider doing/did you do before this job?
   - How does what you do in your job help people?

3. A planetarium may be a site for a field trip. Below are some possible questions to pose to students before and after the trip. How do their answers compare?

- Before visit:
    - What do you think a planetarium is?
    - What do you expect to see there?
    - What kind of people do you think work at a planetarium? What do you think they do?
    - Why do you think planetariums exist? Are they important? If so, why? If not, why not? What do you think they can teach us?
- After visit:
    - What is a planetarium?
    - What did you see on our visit? Was it what you expected? If so, how? If not, why not?
    - Can you name two jobs that people at planetariums do? What does their work involved?
    - Are planetariums important? If so, why? If not, why not? What do they teach us?

1. A science, technology, or engineering museum or business may also be a site for a field trip. Again, below are some possible questions to pose to students before and after the trip. How do their answers compare? Begin by briefly describing or naming the museum or business they will visit.
    - Before visit:
        - What do you think this museum/business is about/does?
        - What do you expect to see there?
        - What kind of people do you think work there? What do you think they do?
        - Do you think this museum/business is important? If so, why? If not, why not? What do you think this museum/business teaches us/does for us?
    - After visit:
        - What is this museum/business about/do?
        - What did you see on our visit? Was it what you expected? If so, how? If not, why not?
        - Can you name two jobs that people have there? What does their work involved?
        - Do you think this museum/business is important? If so, why? If not, why not? What do you think this museum/business teaches us/does for us?

References

Try Your Hand at Programming

Hayes, J. P. (1988). *Computer Architecture and Organization* (2nd ed.). New York, NY: McGraw-Hill Publishing Company.

IEEE Computer Society. (2013). Block diagram. *SEVOCAB: Software and Systems Engineering Vocabulary.* Retrieved from http://pascal.computer.org/sev_display/search.action;jsessionid=5016E8A3FE633AB1052F1778004173AE

Family Letter Template

Dear Family of ________________________________,

Welcome to the Zero Robotics Program! We are very excited to have your child take part in this program. Zero Robotics (ZR) is a five-week robotics programming competition where the robots are SPHERES satellites inside the International Space Station (ISS). Your child will work as part of a team to develop a computer program that will use the SPHERES satellites to solve a space-related 'challenge'. After several phases of virtual competition, finalists will be selected to compete in a live championship aboard the ISS. An astronaut will conduct the championship competition in microgravity with a live broadcast!

The program is designed to engage students in science, technology, engineering and mathematics (STEM) through the ZR competition and by making clear connections to space and space science. It is essential that today's young people understand STEM topics. The overall goal of this program is to excite middle-grades students about STEM, with the hope of encouraging them to pursue their studies in these fields so that they can enter STEM careers.

An important note: During the last few weeks of the program, specifically weeks 3 and 4, your child will be learning to write computer programs. It is very important that they practice programming—or at least thinking about programming/coding—at home as well. Based on our prior experience with this age group and the design of the ZR game, it will be easier for them to write a good program for this competition if they do some extra work at home. Please encourage this!

For more information about ZR, please visit our website at http://www.zerorobotics.mit.edu.

There are several activities that you might do with your child to help him or her continue to explore space-related topics. For example:

1. Visit the NASA for Kids website (http://www.nasa.gov/audience/forkids/activities/index.html) with your child. It has a number of interesting hands-on activities you can do together.
2. Look through the newspaper for space-related stories (or if you prefer, visit an online news source for a similar current event).
3. Watch Engineering.com's top 5 Historical NASA video clips (https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/15448/Top-5-Historical-NASA-Videos.aspx) and discuss each clip with your child afterward.
4. Watch the 1995 movie *Apollo 13*, which dramatizes the 1970 Apollo 13 mission with astronauts Jim Lovell, Jack Swigert, and Fred Haise aboard Apollo 13 for America's third moon landing mission.

Also, consider talking with your child about the following:

- The Mars Rover (http://marsrovers.jpl.nasa.gov/home/index.html)
- The ISS (http://www.nasa.gov/mission_pages/station/main/index.html)
- The Hubble Telescope (http://hubblesite.org/the_telescope/)
- Why is space exploration important? How many space-related careers can you think of? What skills and knowledge do you think someone needs to be successful in those jobs? Try finding out more by doing some looking around on the Internet.

Thank you, and please feel free to contact me with any questions!

____

Zero Robotics Program Leader

Reproducible Masters

RM 1.1: Make Your Own Rocket!

Today you will build and launch your very own rocket!

The first step is to design your rocket. Be creative with your design—do you want big or small fins? What shape do you want your fins? What about your nose cone—long and skinny or short and fat? It's all up to you, but you should remember that these parts have specific jobs—to make the bottle rocket stable and aerodynamic.

Once you have finished designing your rocket, you will go outside and launch it! Whose rocket will go the highest, fly with the most style, and stick the best landing?!? The challenge is *on*!

Design Tips

- Place your fins evenly spaced around your bottle. This will ensure the best flight!
- Make sure that your fins and nose cone are sealed tight on the bottle. You don't want your design to fall apart when the rocket is launched with lots of force!
- Try to avoid having one side be heavier than the other—your bottle won't go straight up if it is lopsided.

Launch Information

NOTE: When your bottle rocket is pressurized, it can be a very dangerous object and can cause serious injury. Stay well back from the rocket and be careful.

To launch the rocket:

1. Fill the bottle one-third to one-half full with water and attach it to the launcher, then stake the launcher to the ground.
2. Make sure that the bottle is completely seated on the O-ring at the base of the launch rod.
3. Insert the launch retainer pin, making sure that it will hold down the bottle by passing over the bottle’s neck ring. (This step may differ depending on the launcher being used—see manufacturer’s instructions for details.)
4. Carefully pressurize the bottle.
    - At no time should you stand above the bottle. Assume that it could blast off at any moment.
    - Do not pressurize the bottle beyond 60 psig.
5. When the rocket is fully pressurized, make sure that nothing is in the way of the rocket, stay clear, and pull out the launch retainer pin.
6. Have one team member record the altitude of the rocket if you are using an inclinometer. Be sure to measure or estimate the distance that your partner is standing from the launch pad so that you can compute the rocket’s altitude given its angle above the horizon.

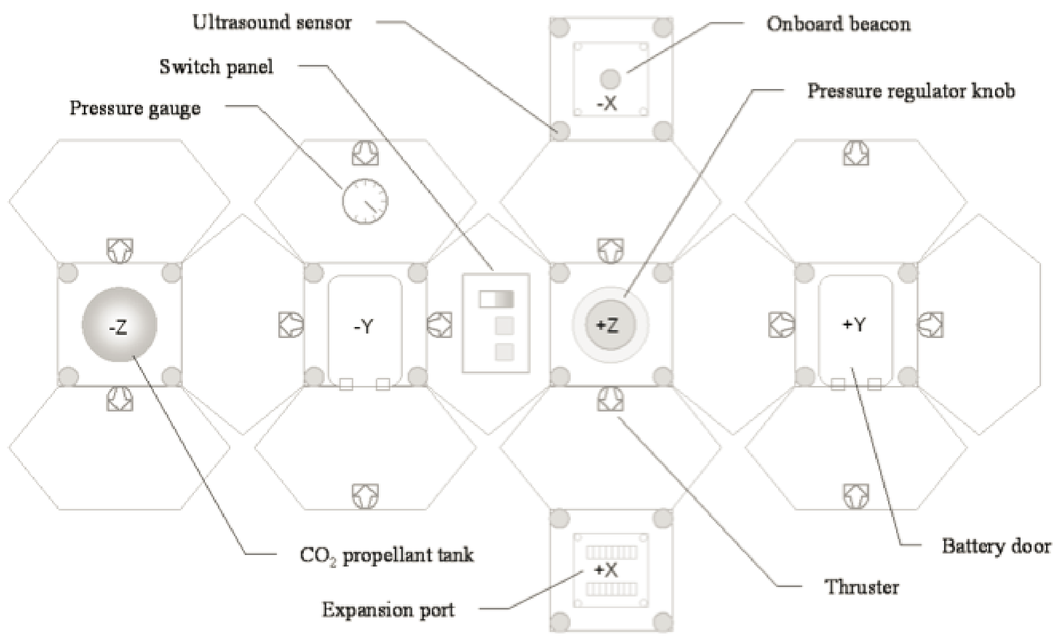
RM 1.2: Make Your Own SPHERE

You will create and label your own SPHERES satellite paper models. Allow yourself at least an hour, as it's harder than it looks to make it fit together cleanly.

To make your first SPHERES model:

1. Cut along all dark lines of the image from the heavy paper you've been given, and fold along the gray straight light lines (the lines connecting the sides and tabs).
2. Attach tabs to the insides of the sides with glue (this is actually a bit tricky!).

Congratulations! You now have a SPHERES satellite. Using the information in the image below, label the parts of your satellite.



Now make two more paper SPHERES, and you can perform your own formation flying experiments!

RM 1.3: Coordinate Clue Solutions

TEAM ___

Clue #	Coordinates	Solution	Work Space
1	Initial clue	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
2	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
3	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
4	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
5	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
6	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
7	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
8	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
9	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	

Clue #	Coordinates	Solution	Work Space
10	(__,__)	x coordinate = ___ y coordinate = ___ **Go to:** (__,__)	
11	(__,__)	x coordinate = ___ y coordinate = ___ **Go to:** (__,__)	
12	(__,__)		

Write answer here: ____

RM 1.4: Making a Sandwich

You will work in a team for this activity, with two “writers” and two “performers.” You need to work together *as a team* to create the final set of instructions (or program) for making a peanut butter and jelly (PB&J) sandwich to give to the computer (your teacher!).

You probably won’t get the program exactly right the first time. You will need to go through a process of trial and error to get the instructions exactly right. This is known as ‘debugging’ your program. You have 15 minutes to complete this activity.

1. Develop the instructions (program) to tell the “computer” (your teacher) how to make a PB&J sandwich. Start brainstorming and write down your first set of instructions. Try not to spend more than five minutes doing this!

2. Have the two performers for your team test the instructions by trying to make a sandwich by following your team’s instructions *exactly*.

> Note to performers: As you are testing the instructions, remember to follow them *exactly as written*. Even if you know that a particular step is needed to make the sandwich, don’t do it unless it’s included in the instructions!

3. While the performers are doing this, the two writers should write down the successes and challenges the performers are experiencing—that is, which parts of the ‘program’ worked and which did not.

4. As a team, revise your initial instructions based on the test and what the writers recorded. You will now have a new, revised set of instructions.

5. Have the performers repeat the process of making a sandwich one more time, using the revised instructions. The writers should again record what worked and what did not. Again, try not to spend more than five minutes doing this.

6. As a team, revise your instructions one last time. This will be your final set of instructions for the computer.

RM 1.5: Writers-Doers: What You Need to Know

In this activity, you will be split into two groups—the Writers and the Doers. You will work in pairs—one of you will be the Writer and one will be the Doer:

- Writer: You will write a set of instructions on how to build one of three simple structures. You will be able to see how the Doer interprets your instructions, and then you will be given an opportunity to "debug" your instructions to help the Doer produce a better output (or build the right structure!). You need to describe the actions needed to build one of three possible structures and the different courses of action that the Doers must take to figure out what to do.
- Doer: You will use the written instructions from your partner to build a structure. You will be given a set of materials to create one of the three structures—but you won't know which one! The Writer's instructions will help you not only build the structure but also figure out which one you *can* build with the materials you have been given.

You can only communicate in writing—NO talking is allowed!

Directions:

1. The Doers leave the room and get to do one of the activities on RM 1.6.
2. The Writers will be shown the same three structures.
3. The Writers will take 15 minutes to write down instructions that should describe to the Doer how to build one of the three structures. The instructions should also help the Doer to figure out which structure the Doer has the materials to build (the Doer won't be told this; the Doer will have to figure it out based on your instructions).

   For example, if the structures were an airplane, a tree, and a boat, your instructions might include details like this: "If your materials include wheels, you are building structure 1; if your materials include leaves, you are building structure 2; if your materials include sails, you are building structure 3."
4. After the 15 minutes is up, each Doer will be given a set of materials and 10 minutes to create *one* of the structures. Doers will be given more parts than they need, and they won't be told which structure they have to build. Doers will have to use their partner's instructions to figure it out.
5. Writers: While the Doers are building, think about what other instructions you can give that would help them. You will do that next.
6. After 10 minutes is up, each Writer has 5 minutes to give the Doer additional written instructions to help the Doer "debug" the structure.
7. The Doers have another five minutes to try to "debug" their structure based on

the new instructions the Writer has given them.

Note to Writers *and* Doers: Try to look at the structures from different angles. Think about the qualities of the individual components themselves, the spatial relationships between the components, and the directions in which the components are facing. Consider the relationships of the components based on where they are in relation to the viewer as well as where they are in relation to each other.

8. Throughout this activity everyone must follow these rules:
   - Writers cannot touch any of the structures.
   - Writers may use letters, numbers, and any symbols, but no drawings in their written instructions.
   - Writers must define any abbreviations used.
   - Writers do not need to write in complete sentences.
   - Again, the only form of communication between the Writer and the Doer is the Writer's instructions (i.e., no talking).
9. Teams will be scored based on the overall number of correct connections. If necessary, the tiebreaker is determined by which Doer completed the structure faster.

RM 1.6: What to Do

While you wait for the Writers . . . try one of the following activities.

OPTION 1: Group Juggle

In this activity, the group passes objects (such as Koosh balls, small stuffed animals, or bean bags) through the air, and each student says the name of the person to whom they are throwing the object. Your leader may add objects, tell everyone to pass objects faster, and/or not allow any talking to increase the difficulty. (Use underhand throws only!)

1. Stand in a circle with your hands out to catch the object. When you catch the object, put your hands down so you can tell who still needs to receive it once. Your leader will start it and also catch it last.
2. The leader starts by calling a person's name, making eye contact, and throwing him or her the object. That person then catches the object and says, "Thank you, ______." That person throws to someone else by calling his or her name and making eye contact. Again, the person who caught the object says "Thank you, ______." This activity will help everyone learn one another's names.

OPTION 2: Making Chocolate Milk

In this activity, the group goes through an exercise of describing the process of making something. You will have the following four objects (or representative objects that your group can use as it works through writing the instructions):

- A gallon of milk
- A squeeze bottle of chocolate syrup
- A glass
- A tablespoon

As a group, write instructions for making a glass of chocolate milk. This activity mimics what the Writers are in the other room doing at that very moment (writing instructions for you)!

RM 2.1: Order, Order!

Figuring out where to send a robot usually requires some number crunching, such as writing equations that will calculate things like distance or velocity. In order to do this, you need to understand what operators are. Some common operators in math and in programming are listed below.

Math Operators:

+ plus or addition

++ increase the value of a number by one

– minus or subtraction

– – decrease the value of a number by one

* multiplication

/ division

Logic Operators:

> greater than

>= greater than or equal to

< less than

<= less than or equal to

Now try your hand at some math problems that use operators. You'll need to use the correct order of operations to find the right answer, so remember GEMA!

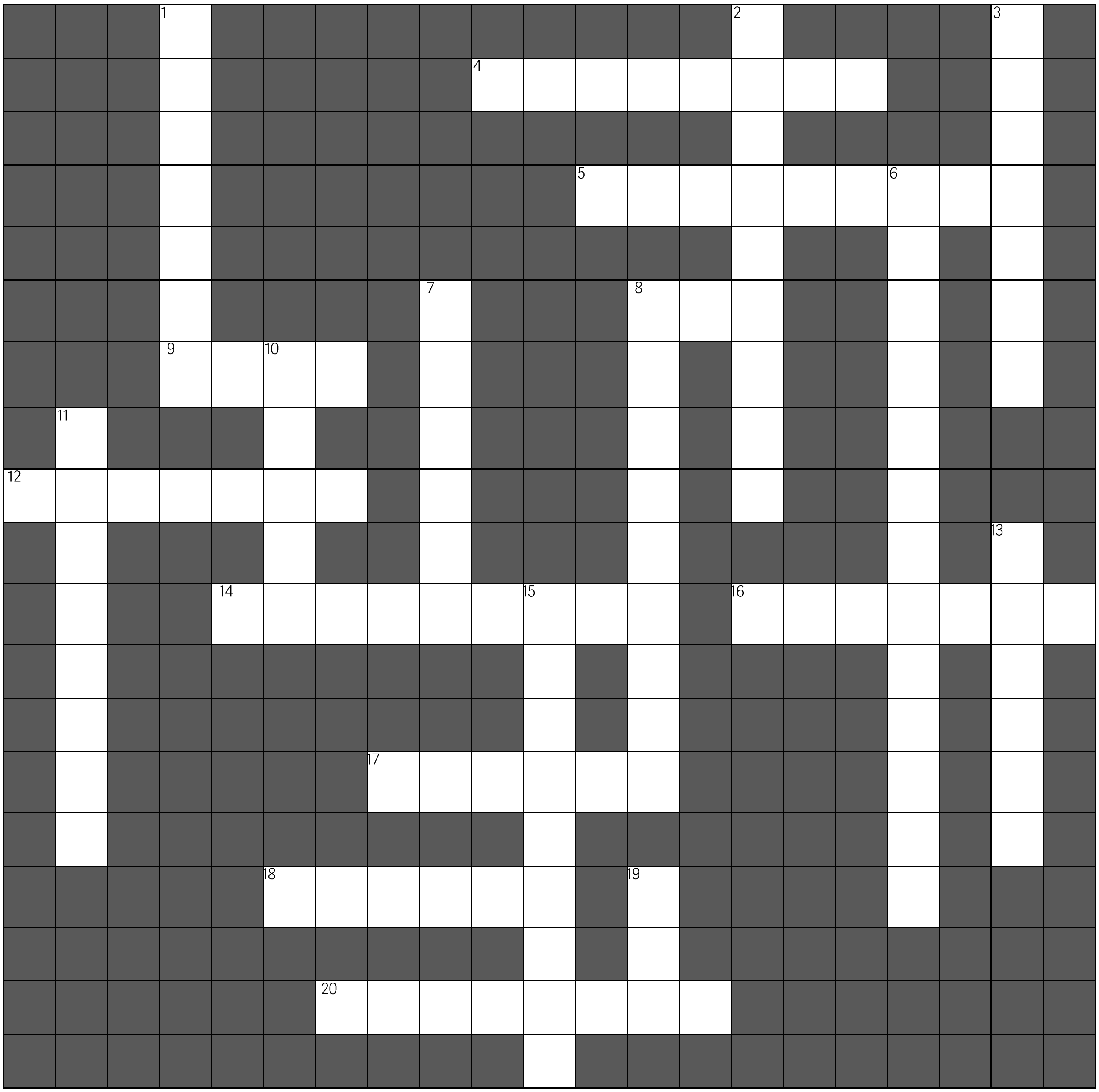
$4 \times (7 + 2)^2 - 8 \div 2$

$7 + (8 - 7 \times 2)^2 \times 3 + 6 \div 2$

$40 \div 5 \times 3 + 7 \times 2^3$

RM 2.2: Order of Operations Crossword

Solve the problems on page 2 using the correct order of operations. Then use the number of your solution to identify the correct word from the Word Bank (also on the next page) to enter into the crossword puzzle.



Word Bank		
–23 international	30 function	–22 IDE
18 magnitude	12 program	46 astronaut
1 computer	4 normal	–12 motion
–25 resultant	11 Newton	13 direction
9 satellite	–13 space	25 station
41 friction	–20 degrees	43 freedom
–2 SPHERES	10 MIT	14 mass

Across

4. $7 \times (2 + 3) - 5$

5. $13 - 2(4 - 2)$

8. $6 \times 2 - 5 + 12 \div 4$

9. $3 \times (16 - 12) + 2$

12. $2 + 5^2 - 15$

14. $2 + 3 \div 3 - 7 \times 2^2$

16. $7 + 10^2 \div 5 - 8 \div 2^2$

17. $(2 + 4)^2 - 14 \div 2 - 7^2$

18. $(17 - 15)^4 - 20 \div (1 + 3)$

20. $50 \div 5^{(4 \div 2)} - (2 \times 2 - 3)$

Down

1. $4 \times 7 + 15$

2. $16 - 12 \div 4$

3. $4 - 3 \times 2$

6. $20 - 15 \times (2 + 3) - 4(6 - 14)$

7. $(2 + 2 - 2 \times (2 + 2) + 2)^2$

8. $15 \div 5 + 15$

10. $4 + 7(3 - 5) - (12 \div 4)$

11. $17 + 2^3 \times 4 - 8$

13. $6 \div 2 + 7^2 - 4^3$

15. $20 \div 5 + 3 \times 4^2 - 3 \times 4 \div 2$

19. $30 \div 6 + 7^{(2-1)} - 17 \times 2$

RM 2.3: Is It a Vector?

Look at the table below and determine whether each measurement is a scalar or a vector by figuring out if it has magnitude and direction, then writing yes or no in the appropriate boxes.

Measurement	Magnitude?	Direction?	Vector?
South			
30 meters northwest			
20 feet			
8 miles west + 10 miles south			
7 cm up			
East of the river			
15 km down the street			

RM 2.4: Vector Hunt Answers

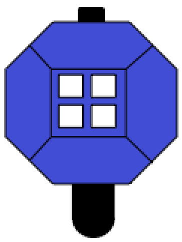
Question: What is the Blue SPHERES' favorite thing to do?

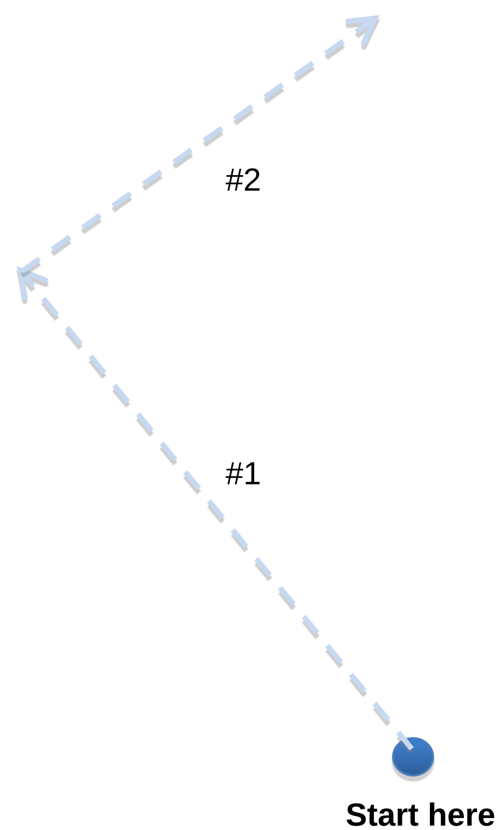
Your Challenge: Conduct the Vector Hunt and write your answers in the table below. Then draw out the vectors on page 2 to find the answer!

#	Magnitude (cm)	Direction (degrees counterclockwise or clockwise)	Rough Sketch (magnitude and direction do not need to be accurate)
1			
2			
3			
4			
5			
6			
7			
8			
9			

- Use a ruler and protractor to copy each vector (with correct magnitude and direction) found on the Vector Hunt below.
- Copy each vector in order (from 1 to 9), placing the tail of each vector at the arrow of the previous vector. (The first two have been lightly drawn in for you as examples—see if you agree.)

Look at the vector path you created to find the answer to the challenge question!

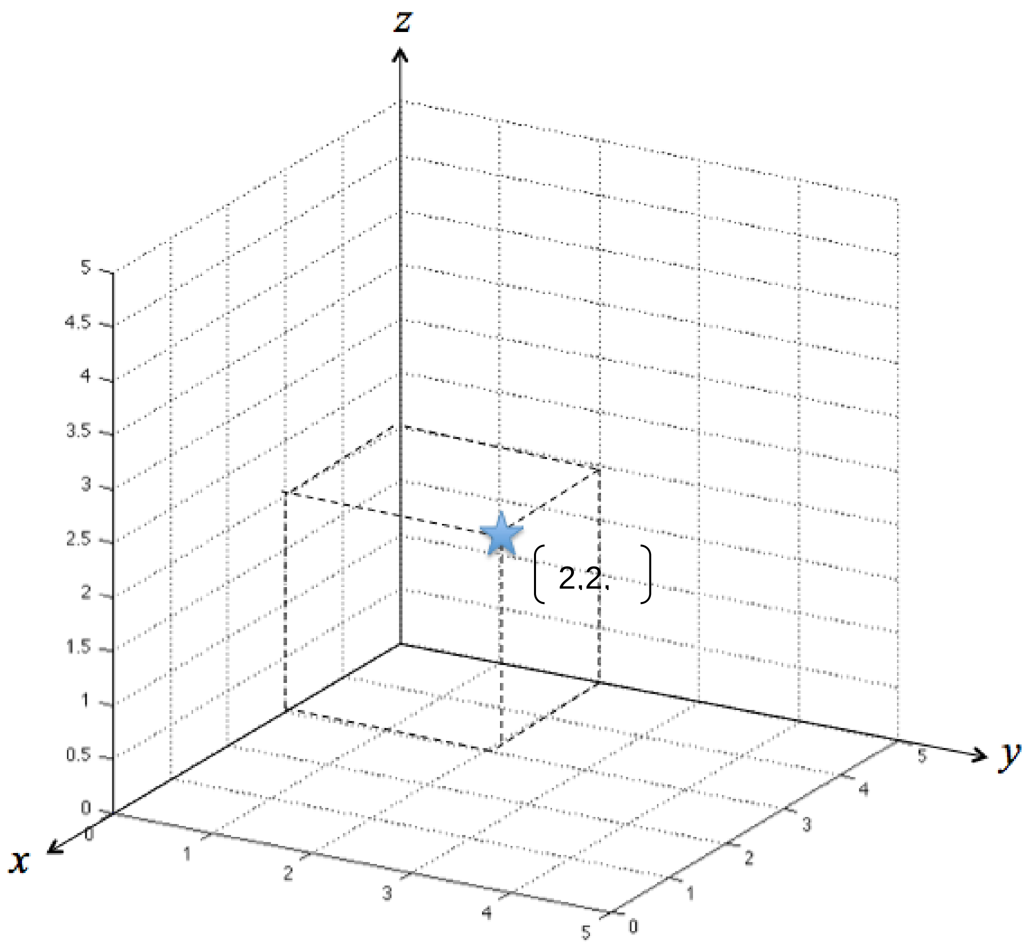




RM 2.5: The Plot Thickens

The table below gives coordinates for different points in 3-D space. The first point has already been plotted for you—note the scale of the *z*-axis is different from that of the *x*- and *y*-axes. Try to add the remaining points on the given 3-D graph.

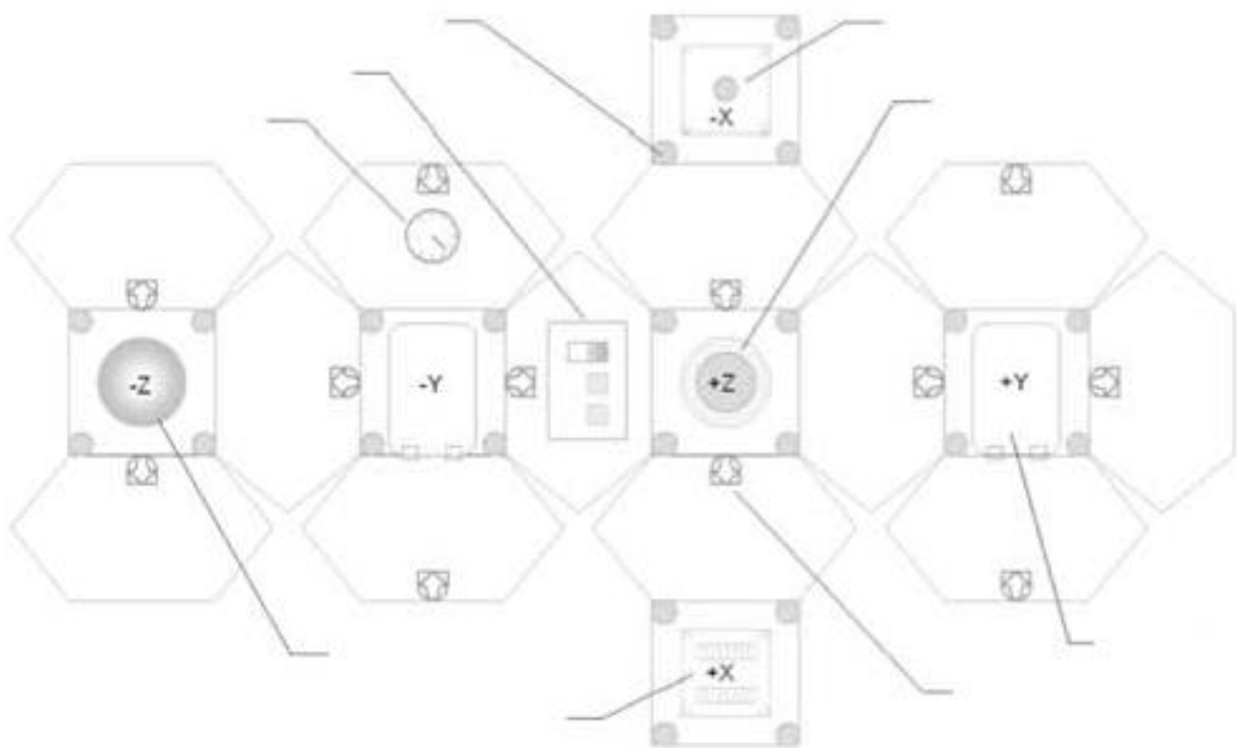
Point	(*x*, *y*, and *z* coordinates)
A	(2,2,2)
B	(0,0,1)
C	(5,5,2)
D	(1,1.5,4)
E	(3.5,4,3.5)
F	(4,4.5,4)



RM 3.1: Name the Parts

Below is a list of all the parts on the outside of the SPHERES satellite. See if you can match each part with one of the labels on the diagram.

- Pressure gauge
- Switch panel
- Ultrasound sensor
- Onboard beacon
- Pressure regulator knob
- Battery door
- Thruster
- Expansion port
- Carbon dioxide propellant tank



RM 3.2: A Question of Kinematics

Speed and velocity both measure how fast a point or object is moving. Speed, however, has no direction while velocity does. So, what is the velocity of the objects you are shown? Fill out the following table to show what you know!

Vehicle	Direction	Speed	Velocity
Helicopter			
Airplane			
Eagle			
Runner			
Rocket			
Satellite			

Now, imagine that two friends, Jane and Paul, want to race their rocket car down a very long straight street. The rocket car starts parked at mile marker "0" at 8 a.m. and fires its rocket engine *once*. The rocket fire creates a force that causes the car to accelerate. Show what you think will happen to the rocket car by filling in the following table:

	Start	Finish
Position		
Speed		
Acceleration		
Time		

What if Jane and Paul were racing their rocket car *in space*?!?! If they start their race outside the International Space Station, what will happen to the rocket car after it fires its rocket engine?

	Start	Finish
Position		
Speed		
Acceleration		
Time		

Appendices

Appendix 1: Ice-Breaker Activities

The following are some activities that you can use either as "ice breakers" or to break up programming and other non-active components of the program.

- Circle Games
  - Each person says his or her name followed by an adjective that best describes him- or herself.
  - As a second round, the group repeats together with the person the name and adjective.
  - Each person says his or her own name followed by a gesture and sound.
- Machine of Rhythms (good outdoor activity)
  - One person starts a rhythmical gesture and sound; one by one, students add to this gesture and sound, trying to create a unified machine.
  - Students should be close to each other in order to better portray the machine.
  - Once students all have their own gesture and sound, ask them to do it faster, louder, slower, or softer in any variation you desire.
- The Driver Is Crazy (good outdoor activity)
  - Split the group into two teams and have each team form a line.
  - The first person in each line is the driver; at your signal, each driver will drive in a crazy way while all the others on the line behind the driver need to hold tight to the person ahead (at waist level) so as not to get loose.
  - Signal the drivers to begin.
  - Change the driver until each one has a turn.
  - Mix up the two teams and go for a second round.
- Collaborative Storytelling
  - Each student says one word for a story created on the spot.
  - *Variation:* Each student says a sentence for a story being created by the whole group.
- The Boss
  - Have students stand in a circle.
  - One student volunteers to leave the room.
  - Choose one student to act as the "boss." The boss will make a movement or sound that everyone should follow and will change the movement or sound often.
  - When the volunteer comes back, everybody is doing the same movement or sound. The volunteer needs to find out who is the boss.

- Sticky Paper
  - One student stands in the center of the group; the others might be touching that student or one another, but there should be a sheet of paper between the touching parts of the bodies.
  - The person in the center must move and everyone else must move with her or him so that the papers do not drop.
- Keep It Up
  - Give each student one balloon.
  - Each student should throw the balloon up in the air and then keep it from falling onto the floor, *without using his or her hands*.
  - If a student's balloon falls, you should take the balloon and that student will help another student keep his or her balloon from falling.
- Walking On . . . (good outdoor activity; especially good for younger students)
  - Direct students to walk around the room on the kind of surface chosen by you (ocean, hot sand, sticky mud, stones, and so on).
  - Call out a change in the kind of surface every 30 seconds to 1 minute.
- Grandmother's Footsteps (good outdoor activity; especially good for younger students)
  - Have one student face a wall or other object and the others stand some distance away facing the same direction.
  - As the student facing the wall counts to three, everyone else should start moving toward the counter.
  - At three, the counter will turn around and face the walkers. Whoever she catches in movement when she turns needs to start over at the beginning.
  - This goes on until someone manages to touch the student facing the wall.
- Lollipop List
  - Pass out dum-dum lollipops to the group.
  - For every letter that appears in their flavor, each student has to share something about himself or herself with the group.
- Mumble Jumble
  - Before the activity, cut up a few pictures into puzzle pieces and place them in a bag.
  - Have each student grab a piece of a puzzle from the bag.
  - The students will keep their puzzle pieces to themselves until you say "GO!"
  - Students need to try to locate the other members of the group with the pieces to form the appropriate pictures. Whichever team does it first, wins.
- Ice Cubes (particularly good on a hot day)
  - Divide the group into equal teams of three or more players.
  - At a starting signal, each team needs to pick up ONE ice cube and tries to

melt it as quickly as possible.
  - Players can rub it between their hands or against their clothes, but they may NOT put it in their mouths or on the ground.
  - While they are trying to melt the ice cube, it should be passed around the team frequently—no player should keep it for more than a few seconds at a time.
  - The first team to melt its ice cube wins.
- Dead Fish (this is a great game for those low-key early mornings or after lunch when a group activity is still necessary; it's also useful for calming down a high-energy group)
  - Have everyone get into a comfortable position that they can sustain for a long period of time.
  - Once everybody has established a position, count down from ten to zero. At zero, the game will begin.
  - Once the game has begun, nobody is allowed to talk or move, with the exception of the eyes and chest for breathing.
  - If *you* notice anyone talk or move, remove the person from the game.
  - Anyone removed from playing may persuade others to talk or move, but they may not physically touch those players still in the game or indicate to you that anyone has moved.
  - The last person remaining becomes the leader of the next round.
- M & M Swap (good outdoor activity)
  - Have the group stand up and form a circle.
  - Give each student a plastic spoon and place four or five M&Ms in the first student's spoon.
  - Have everyone put their spoons into their mouths, holding the handle between their teeth with the scoop of the spoon facing up.
  - The group must pass the M&Ms from the first person to the last, without using their hands. The object is to get ALL of the M&Ms to the end, without dropping any.
- Venn of You
  - Divide the group into teams of three or four.
  - Give each team a large sheet of butcher paper and a different colored marker for each person.
  - Have each team draw a Venn diagram with an oval for each student in the team.
  - The teams should discuss what their similarities and differences are.
  - After the discussion, they fill in the diagram showing their similarities and differences.

    - If a team has a hard time getting started, give them some guidance by asking questions such as, "What is your favorite music?", "When is your birthday?", "What sports do you like?", or "Where were you born?

- I Am What I Wear
    - Ask students to introduce themselves by saying their names and revealing what one (visible) article of clothing reveals about them.
    - For example, "I'm Robin and these sneakers reveal how boring I am because I wear them every single day," or "I'm Max and this shirt reveals how organized I am because I knew I would be wearing it four days ago."

Appendix 2: Materials Needed

Summary Materials List

Generally Available Supplies

- Scissors
- Tape
- Glue
- Markers, pens, pencils
- Construction paper
- Rulers
- Protractors

Less Common Supplies

- Flip chart paper or newsprint
- Heavy paper 8.5 x 11 (card stock - bright colors for SPHERES models - print in advance?)
- Stickers or awards
- Poster board (for ISS Finals posters)
- Video equipment or cell phone with video capability (optional)

"What's the Logic" activity

- Sandwich making supplies

Coordinate activity

- Plastic drinking straws
- Donuts or Play Doh

Bottle Rocket supplies

- 2 liter soda bottle – one per each pair of students
- Bottle launcher (such as: https://www.amazon.com/Aquapod-Bottle-Launcher-Fluorescent-Orange/dp/B003Y5DOJC/ref=as_at?creativeASIN=B003Y5DOJC&linkCode=w61&imprToken=Lzgm5ppys1A8LOrPovJ3PA&slotNum=0&tag=vat19oos-20)

- Bicycle air pump with pressure gauge
- Duct tape (other types of tape will not work)
- Bottle Rockets Part 2 (in Week 5): Eggs (optional)

Legos *or* building kits, such as one of the following:

- Building kits with connectors (https://brackitz.com)
- Marble Runs (http://www.mindware.com/p/Marble-Run-103-Piece-Set/25078)
- Keva structures (http://www.kevaplanks.com/keva-maple-200-2-1-2)

Thruster Balloon supplies

- Balloons
- Plastic drinking straws
- String

Model of the ISS (optional activity)

- Fishing line
- Wood skewers or ice pops
- Heavy paper 8.5 x 11

Dry Ice Activity (This is an optional activity for Field Day. If you choose to do the Dry Ice activity at Field Day, the following are the necessary materials.)

- 10-lb block dry ice
- Cooler
- Poster paper, approx. two 1-ft pieces of 2 x 4 lumber
- Ramp (can be a table with one end raised)
- Pair of insulated gloves
- Pair of safety glasses
- Hammer
- Fan

Full materials list for:

Materials Needed for Week 1

For the Love of Space!

For each student:

- Program Introduction Letter (**Family Letter Template** on page 150 is a resource to help you craft letters to students' families during the program)

For the group:

- Computer with Internet connection and projection equipment or TV and DVD

The Sky Is Falling!

For the group:

- Three prepared sheets of newsprint (see "Before You Teach" on page 10) plus extra sheets of newsprint
- Markers

Optional:

- Computer with Internet connection and projection equipment or TV and DVD
- *Optional:* Clips from *Armageddon* and/or *Deep Impact*

What in the Universe?!?

For the group:

- Computer with Internet connection and projection equipment
- Some short video clips and/or other multimedia highlighting space and space exploration, such as the Mars rover landing, the ISS, and the moon landing, selecting from the following resources:

- NASA Johnson Style (https://www.youtube.com/watch?v=2Sar5WT76kE)
- Space Shuttle Launch (http://www.wimp.com/spaceshuttle/) (3:52)
- Challenges of Getting to Mars: Curiosity's Seven Minutes of Terror (http://www.youtube.com/watch?v=Ki_Af_o9Q9s) (5:08)
- Engineering.com's top 5 Historical NASA video clips (24:99) (https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/15448/Top-5-Historical-NASA-Videos.aspx)
- Mars in a Minute: How Do Rovers Drive on Mars? animation (http://www.jpl.nasa.gov/video/?all_videos&id=1176#fragment-5) (1:00)

Moving Off Earth

For each students:

- Computer with Internet connection and headphones (optional)

For each team:

- Scissors (at least one pair)
- Tape (regular clear tape, masking or duct tape)
- Glue (optional)

For the group:

- Computer with Internet connection and projection equipment to show videos and ISS Facts PowerPoint presentation
- Visual Resources to explore the ISS, chosen from the following resources (available in the student materials)
  - Further Up Yonder: Time Lapse of the Earth from the ISS (http://www.guardian.co.uk/sciece/video/2012/nov/29/earth-international-space-station-tinelapse-video) (2:28)
  - Everything About Mission Control (7:40) (https://www.youtube.com/watch?v=Ne57B7QI4gk)
  - See How the ISS Was Assembled (2:13) (https://www.youtube.com/watch?v=h8kOAroNNAo)
  - Virtual Tour of ISS (http://esamultimedia.esa.int/multimedia/virtual-tour-iss/)
    Note: *Show students how to click on arrows to move around station and click on white triangles inside blue circles to watch narrated videos of points of interest and then let them explore.*

    - A Narrated Tour of the ISS (https://www.nasa.gov/mission_pages/station/main/suni_iss_tour.html)
      Note: *"Station Tour: Harmony, Tranquility, Unity" (8:41) shows you life on the ISS including where and how the astronauts sleep, what they eat and the bathroom facilities.*

- Heavy 8½" x 11" paper (to print ISS paper model templates from NASA
- Nylon fishing line
- Wood skewers or ice pop sticks (for one team)

Welcome to Your Launchpad

For each student:

- Computer with Internet connection
- Email address

For the group:

- Computer with Internet connection and projection equipment to show the PowerPoint presentation for the *Create an Account* tutorial
- Group list for students to record their username and password

Bottle Rockets, Part 1

For each student:

- Copy of **RM 1.1: Make Your Own Rocket!**

For each team of students:

- At least 1 2-Liter soda bottle (Coca-Cola Company brands work best)
- Construction paper and/or cardstock
- Duct tape to attach nose cone/fins to bottle (do not substitute with other tape)
- Scissors
- Computer with Internet connection

For group:

- Bottle launcher—as noted on page 4 of this guide, you may want to either purchase a bottle rocket launcher or build your own.
- Launch retainer pin
- Air pump
- A few metal stakes

- Ample supply of water

> ***NOTE:*** This activity can be dangerous—never stand over the bottle once it is in the launcher, and do not allow students to operate the pump without close supervision.
>
> If you are uncomfortable doing this activity, you could use one of the following instead:
> - 3 . . . 2 . . . 1 . . . PUFF! rocket activity from NASA
> - Foam Rocket activity from NASA

You Have Nothing to SPHERE . . .

For each student:

- Copy of **Make Your Own SPHERES (RM 1.2)**
- Piece of brightly-colored cardstock with image of SPHERES printed on it (image is available in the student materials)

For the group:

- Computer with Internet connection and projection equipment to show video and Getting to Know SPHERES PowerPoint presentation

Try Your Hand at Programming

For each student:

- Computer with Internet connection

For the group:

- Table on newsprint or chart paper, prepared according to the directions in Before You Teach on page 10
- Computer with Internet connection and projection equipment to show the Getting to Know the ZR IDE and Mystery Coordinate Grid PowerPoint presentations

Optional Activity: Coordinate Hunt

For each student:

- Copy of **Coordinate Clue Solutions (RM 1.3)**

For the group:

- Copy of Coordinate Hunt Pieces, prepared according to the directions in Before You Teach on pages 10 and 11

- Computer with Internet connection and projection equipment to show the Coordinate Hunt PowerPoint presentation

Optional Activity: Battleship

For each pair of students:

- Either the board game "Battleship" or some graphing paper

Get in the Game

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to show the ZR Game Introduction video and the Game Overview PowerPoint presentation

What's the Logic?

For each student:

- Copies of RM 1.4: Making a Sandwich

For the group:

- 1 jar of peanut butter or other spread (see Note below)
- 1 jar of jelly
- Additional containers to distribute jelly and peanut butter/spread across teams of 4 students
- Ball or similar object to represent the satellite
- Computer with Internet connection and projection equipment

For each team:

- 2 slices of nut-free bread
- 2 butter or plastic knives
- Lined writing paper
- Pencils
- *Optional:* A small collection of blocks or similar objects

* NOTE: Due to food allergies, it is highly recommended that alternate spreads be used in place of peanut butter. Some options include soy butter, spreadable butter or margarine, or whipped cream cheese.

Getting into CS

For the group:

- Computer with Internet connection and projection equipment to show the Getting Into CS PowerPoint presentation

Getting Your SPHERE Lined Up

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to show the Sample Arrays and Introduction to Arrays and the setPositionTarget Function PowerPoint presentations

Logic Competition

For each student:

- Copy of RM 1.5: Writers-Doers: What You Need to Know

For each pair of students:

- Copy of RM 1.6: What to Do
- Paper
- Pen or pencil
- A set of materials to build one of the three structures

For group:

- Legos or building kits such as the following:
    - Building kits with connectors (https://brackitz.com)
    - Marble runs (http://www.mindware.com/p/Marble-run-103-Piece-Set/25078)
    - Keva structures (http://www.kevaplanks.com/keva-maple-200-2-1-2))

Learning All the Right Moves

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to show:
    - Rotation in 2-D PowerPoint presentation
    - *Optional:* PowerPoint presentations for the *More Simple Arrays and the setAttitudeTarget Function* and *More Simple Arrays—Another Way to Initialize Variables* tutorials

Materials Needed for Week 2

Field Day

For the group:

- Materials described in the guide for the Acting Out the Game activity

What's the Right Order?

For each student:

- Copy of RM 2.1: Order, Order!
- Copy of RM 2.2: Order of Operations Crossword

As If!

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to show The Conditionals: The Basic of "If-Then" PowerPoint presentation

What's in a Vector?

For each student:

- Copy of RM 2.3: Is It a Vector?

- Copy of RM 2.4: Vector Hunt Answers
- Optional: Computer with Internet connection
- Paper
- Pen

For each pair of students:

- Ruler
- Protractor

For the group:

- Computer with Internet connection and projector to show the Directions with No Direction

Seeing in 3-D—How to Visualize Space!

For each student:

- Copy of RM 2.5: The Plot Thickens
- Computer with Internet connection

For the group:

- Computer with Internet connection and projector to show the Dimensions PowerPoint presentation

More Conditionals

For each student:

- Computer with Internet connection

What Else?

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to show the What If: An Introduction to Conditional Statements PowerPoint presentation

How to Make SPHERES Move

For the group:

- *Optional*: 2 chairs with wheels

For each pair of students:

- 2 balloons
- Straw
- Piece of string (10–15 ft., depending on the size of the room)
- Tape
- Newsprint or chart paper to record students' ideas

Optional: What Friction?

Note: A long flat surface, such as a table, is needed for this activity.

For each student:

- Marker

For the group:

- Roll of poster paper
- 4 heavy objects
- Materials for the obstacle courses
- Dry ice ramp (provided)
- One pair of insulated gloves with long sleeves (welding or winter gloves work well)
- Safety glasses
- 10-lb block dry ice
- Cooler (for dry ice)
- Hammer
- Large storage tub or laundry basket
- *Optional*: 2 1-ft pieces of 2 × 4 lumber
- *Optional*: Fan

Breaking Down Programming

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projector to show the Programming Functions PowerPoint presentation

Let the Games Begin!

For each student:

- Computer with Internet connection

For each team:

- Newsprint
- Markers

For the group:

- Computer with Internet connection and projection equipment to display *Introduction to Game Mode* tutorial and **Let the Games Begin** PowerPoint presentations

Materials Needed for Week 3

SPHERES: Here, There, Everywhere!

For each student:

- Copy of RM 3.1: Name the Parts

For group:

- Computer with Internet connection and projection equipment to show the **All About SPHERES** PowerPoint presentation

Thrown for a Loop!

For each student:

- Computer with Internet connection

It's Game Time!

For each student:

- Computer with Internet connection

For each team:

- Planning notes from Week 2
- Newsprint
- Markers

Reaching Warp Speed

For each students:

- Computer with Internet connection
- Copy of RM 3.2: A Question of Kinematics

For group:

- Computer with Internet connection and projection equipment to show video and the Mapping with Speed PowerPoint presentation
- Table on newsprint or board, prepared in Before You Teach on page 77

If the SPHERE Is Here, Then . . .

For each student:

- Computer with Internet connection

Keeping Track of Things

For each student:

- Computer with Internet connection

Game On

For group:

- Computer with Internet connection and projection equipment to show the Intramural Competition
- Stickers or awards
- Newsprint or board to record results of the Intramural Competition

Take a Break!

For each student:

- Computer with Internet connection

For group:

- Computer with Internet connection and projection equipment
- Additional materials needed depend on any activities selected.

Materials Needed for Week 4

Instant Replay

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment to replay simulations from practice Regional Competition
- Newsprint or board and markers

Your Move

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment
- Newsprint or board and markers to record group strategy

Programming In Space Is STEMulating!

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment
- Other materials needed depending on activities selected

Getting to the Finals

For each student:

- Computer with Internet connection

Materials Needed for Week 5

The Announcement

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Showcase Your Team

For each student:

- Handout to send home to families providing the date, time, and location of the ISS Finals Event

For the group:

- Poster boards
- Markers
- Video equipment or cell phones
- Computer with Internet connection and projection equipment

Bottle Rockets, Part 2

For each team:

- Bottle rockets from Week 1
- Computer with Internet connection
- Cloth, wood, duct tape, and a variety of other materials students might use for their designs
- *Optional*: 1–2 eggs

How Bright Can Your Future Be?

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

What Is NASA Up To?

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Other Ways to Get to Space

For each student:

- Computer with Internet connection

Smooth Moves

For group:

- Computer with Internet connection and projection equipment to show the Smooth Moves PowerPoint presentation

How Many Degrees of Freedom Do You Have?

For group:

- Newsprint with "NASA" written on it, prepared in Before You Teach

Searching for Whatever Is Out There

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Where Can You Go From Here?

For each student:

- Computer with Internet connection

For the group:

- Computer with Internet connection and projection equipment

Popcorn Time!

For the group:

- Computer with Internet connection and projection equipment or television and DVD player

Appendix 3: Alignment to Curriculum Standards

The Common Core State Standards (CCSS) and Next Generation Science Standards (NGSS) provide a clear, coherent, and consistent framework and related benchmarks for the skills and knowledge that students need to succeed in college and careers. The ZR Middle School Program curriculum provides a rigorous and engaging informal education curriculum that will allow your students to achieve these new standards.

The NGSS reflect a number of key shifts in science education, including a focus on deeper understanding and application of content, the real world interconnections in science, and the integration of science and engineering. The NGSS recognize that K–12 science education cannot teach the breadth of our ever-increasing scientific knowledge and thus the goal must be to prepare students with enough core knowledge and the skills to evaluate and select information so that they can add to their knowledge base on their own. The NGSS coordinate with the CCSS literacy standards, emphasizing the importance of applying literacy skills to science.

The CCSS includes benchmarks for Literacy in History/Social Studies, Science, and Technical Subjects. These literacy standards focus on developing and using literacy skills across the disciplines and highlight the importance of the application of knowledge through higher-order skills such as critical thinking and problem-solving.

The following tables list the NGSS and the CCSS in Science and Technical Subjects that are applicable to this curriculum.

Next Generation Science Standards
Science and Engineering Practices
Developing and using models
Planning and carrying out investigations
Analyzing and interpreting data
Using mathematics and computational thinking
Constructing explanations (for science) and designing solutions (for engineering)
Obtaining, evaluating, and communicating information
Crosscutting Concepts
Cause and effect
Scale, proportion, and quantity
Systems and system models
Structure and function
Disciplinary Core Ideas—Middle School (6–8)
Physical Sciences: MS-PS2 Motion and Stability: Forces and Interactions

MS-PS2-1. Apply Newton's Third Law to design a solution to a problem involving the motion of two colliding objects.

MS-PS2-2. Plan an investigation to provide evidence that the change in an object's motion depends on the sum of the forces on the object and the mass of the object.

Engineering, Technology, and Applications of Science

MS-ETS1-1. Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.

MS-ETS1-2. Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

MS-ETS1-3. Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

Disciplinary Core Ideas—High School (9–12)

Physical Sciences: HS-PS2 Motion and Stability: Forces and Interactions

HS-PS2-1. Analyze data to support the claim that Newton's second law of motion describes the mathematical relationship among the net force on a macroscopic object, its mass, and its acceleration.

Common Core State Standards for Literacy in Science, and Technical Subjects

Reading Standards for Literacy in Science and Technical Subjects

Key Ideas and Details

CCSS.ELA-Literacy.RST.6-8.3 Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks

Craft and Structure

CCSS.ELA-Literacy.RST.6-8.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to *grades 6–8 texts and topics*.

Integration of Knowledge and Ideas

CCSS.ELA-Literacy.RST.6-8.7 Integrate quantitative or technical information expressed in words in a text with a version of that information expressed visually (e.g., in a flowchart, diagram, model, graph, or table).

Writing Standards for Literacy in Science and Technical Subjects

Text Types and Purposes

CCSS.ELA-Literacy.WHST.6-8.2 Write informative/explanatory texts, including the narration of historical events, scientific procedures/ experiments, or technical processes.

Research to Build and Present Knowledge

CCSS.ELA-Literacy.WHST.6-8.7 Conduct short research projects to answer a question (including a self-generated question), drawing on several sources and generating additional related, focused questions that allow for multiple avenues of exploration.
CCSS.ELA-Literacy.WHST.6-8.9 Draw evidence from informational texts to support analysis reflection, and research.

Appendix 4: Coordinate Hunt Solutions

Solutions for the coordinate clues for Teams 1, 2, 3, and 4 are provided in separate tables below. A partially-completed student solution page (for RM 1.3: Coordinate Clue Solutions that students fill in as they complete the Coordinate Hunt) is also provided on pages 28A and 29A.

Team 1 Solutions

Clue #	Coordinates	Problem and Solution
1	Team 1 Initial clue	The x and y coordinates add up to 0. The x coordinate is exactly 2 units smaller than the y coordinate. **Go to** (−1,1)
2	(−1,1)	x coordinate: $4 - 6 \div 2$ y coordinate: x coordinate on current clue $+ 5$ **Go to** (1,4)
3	(1,4)	x coordinate: $2x + 2 = -6$ y coordinate: $3y + 8 = 17$ **Go to** (−4,3)
4	(−4,3)	If $y = 6$, solve for x: x coordinate: $y + 3x = 18$ If $x = 6$, solve for y: y coordinate: $3y + 2x = 3$ **Go to** (4,–3)
5	(4,–3)	Reflect over $y = x$ line (switch current x and y coordinates) **Go to** (−3,4)
6	(−3,4)	x coordinate: $\frac{2+7}{3} - 2$ y coordinate: $\frac{2-7}{5}$ **Go to** (1,–1)

Clue #	Coordinates	Problem and Solution
7	(1,–1)	x coordinate (solve for x): $\frac{-150}{375} = \frac{x}{10}$ y coordinate: If you multiply any number by y, you get y. What is y? **Go to** (–4,0)
8	(–4,0)	Graph of $y = \frac{-x}{3} + 1$ x coordinate: the x intercept y coordinate: the y intercept **Go to** (3,1)
9	(3,1)	In the third quadrant. The product of x and y coordinates (both integers) is 1. **Go to** (–1,–1)
10	(–1,–1)	x coordinate: $\frac{9}{8} \div \frac{1}{3} + \frac{5}{8}$ y coordinate: $\frac{1}{3} + \frac{1}{2} - \frac{17}{6}$ **Go to** (4,–2)
11	(4,–2)	Great! Now go to the origin for you final clue! **Go to** (0,0)
12	(0,0)	**How to crack the code:** 1=a, 2=b, 3=c . . . 25=y, 26=z Answer: You need to press the power button!

TEAM 2 SOLUTIONS

Clue #	Coordinates	Problem and Solution
1	Team 2 Initial clue	The x and y coordinates add up to 0. The x coordinate is exactly 4 units smaller than the y coordinate. **Go to** (–2,2)
2	(–2,2)	x coordinate: 4 – 6 ÷2 y coordinate: x coordinate on current clue + 4 **Go to** (1,2)
3	(1,2)	x coordinate: $2x + 1 = -3$ y coordinate: $3y + 8 = 17$ **Go to** (–2,3)
4	(–2,3)	If $y = 6,$ solve for x: x coordinate: $y + 3x = 12$ If x=6, solve for y: y coordinate: $3y + 2x = 3$ **Go to** (2,–3)
5	(2,–3)	Reflect over $y = x$ line (switch current x and y coordinates) **Go to** (–3,2)
6	(–3,2)	x coordinate: $\frac{2+7}{3} - 1$ y coordinate: $\frac{2-7}{5} - 1$ **Go to** (2,–2)
7	(2,–2)	x coordinate: $\frac{-150}{375} = \frac{x}{5}$ y coordinate: If you multiply any number by y, you get y. What is y? **Go to** $(-2,0)$
8	(–2,0)	Graph of $y = \frac{-2x}{3} + 2$ x coordinate: the x intercept y coordinate: the y intercept **Go to** (3,2)

Clue #	Coordinates	Problem and Solution
9	(3,2)	In the third quadrant. y coordinate > -2 The product of x and y coordinates (both integers) is 2. **Go to** (–2,–1)
10	(–2,–1)	x coordinate: $\frac{9}{8} \div \frac{1}{3} + \frac{5}{8}$ y coordinate: $\frac{1}{3} + \frac{1}{2} - \frac{11}{6}$ **Go to** (4,–1)
11	(4,–1)	Great! Now go to the origin for you final clue! **Go to** (0,0)
12	(0,0)	**How to crack the code:** 1=a, 2=b, 3=c . . . 25=y, 26=z Answer: You need to press the power button!

TEAM 3 SOLUTIONS

Clue #	Coordinates	Problem and Solution
1	Team 3 Initial clue	The x and y coordinates add up to 0. The x coordinate is exactly 6 units smaller than the y coordinate. **Go to** (–3,3)
2	(–3,3)	x coordinate: $4 - 6 \div 2$ y coordinate: x coordinate on current clue $+ 1$ **Go to** (1,–2)
3	(1,–2)	x coordinate: $2x + 2 = 6$ y coordinate: $3y + 8 = 17$ **Go to** (2,3)
4	(2,3)	If $y = 6$, solve for x: x coordinate: $y + 3x = 0$ If x = 6, solve for y: y coordinate: $3y + 2x = 3$ **Go to** (–2,–3)
5	(–2,–3)	Reflect over $y = x$ line (switch current x and y coordinates) **Go to** (–3,–2)
6	(–3,–2)	x coordinate: $\frac{2+7}{3}$ y coordinate: $\frac{2-7}{5} - 2$ **Go to** (3,–3)
7	(3,–3)	x coordinate: $\frac{150}{375} = \frac{x}{5}$ y coordinate: If you multiply any number by y, you get y. What is y? **Go to** (2,0)
8	(2,0)	Graph of $y = -x + 3$ x coordinate: the x intercept y coordinate: the y intercept **Go to** (3,3)

Clue #	Coordinates	Problem and Solution
9	(3,3)	In the third quadrant. y coordinate > -2 The product of x and y coordinates (both integers) is 3. **Go to** (–3,–1)
10	(–3,–1)	x coordinate: $\frac{9}{8} \div \frac{1}{3} - \frac{3}{8}$ y coordinate: $\frac{1}{3} + \frac{1}{2} - \frac{17}{6}$ **Go to** (3,–2)
11	(3,–2)	Great! Now go to the origin for you final clue! **Go to** (0,0)
12	(0,0)	**How to crack the code:** 1=a, 2=b, 3=c . . . 25=y, 26=z Answer: You need to press the power button!

TEAM 4 SOLUTIONS

Clue #	Coordinates	Problem and Solution
1	Team 4 Initial clue	The x and y coordinates add up to 0. The x coordinate is exactly 8 units smaller than the y coordinate. **Go to** (–4,4)
2	(–4,4)	x coordinate: $4 - 6 \div 2$ y coordinate: x coordinate on current clue **Go to** (1,–4)
3	(1,–4)	x coordinate: $2x + 2 = 10$ y coordinate: $3y + 8 = 17$ **Go to** (4,3)
4	(4,3)	If $y = 6$, solve for x: x coordinate: $y + 3x = -6$ If x = 6, solve for y: y coordinate: $3y + 2x = 3$ **Go to** (–4,–3)
5	(–4,–3)	Reflect over $y = x$ line (switch current x and y coordinates) **Go to** (–3,–4)
6	(–3,–4)	x coordinate: $\frac{2+7}{3} + 1$ y coordinate: $\frac{2-7}{5} - 3$ **Go to** (4,–4)
7	(4,–4)	x coordinate: $\frac{150}{375} = \frac{x}{10}$ y coordinate: If you multiply any number by y, you get y. What is y? **Go to** (4,0)

Clue #	Coordinates	Problem and Solution
8	(4,0)	Graph of $y = -\frac{-4x}{3} + 4$ x coordinate: the x intercept y coordinate: the y intercept **Go to** (3,4)
9	(3,4)	In the third quadrant. y coordinate > –2 The product of x and y coordinates (both integers) is 4. **Go to** (–4,–1)
10	(–4,–1)	x coordinate: $\frac{9}{8} \div \frac{1}{3} - \frac{3}{8}$ y coordinate: $\frac{1}{3} + \frac{1}{2} - \frac{11}{6}$ **Go to** (3,–1)
11	(3,–1)	Great! Now go to the origin for you final clue! **Go to** (0,0)
12	(0,0)	**How to crack the code:** 1=a, 2=b, 3=c . . . 25=y, 26=z Answer: You need to press the power button!

Example Team Coordinate Clue Solutions

The first few solutions for the Team 1 coordinate clues are provided as an example of how students should fill in **Coordinate Clues Solutions (RM 1.3)**.

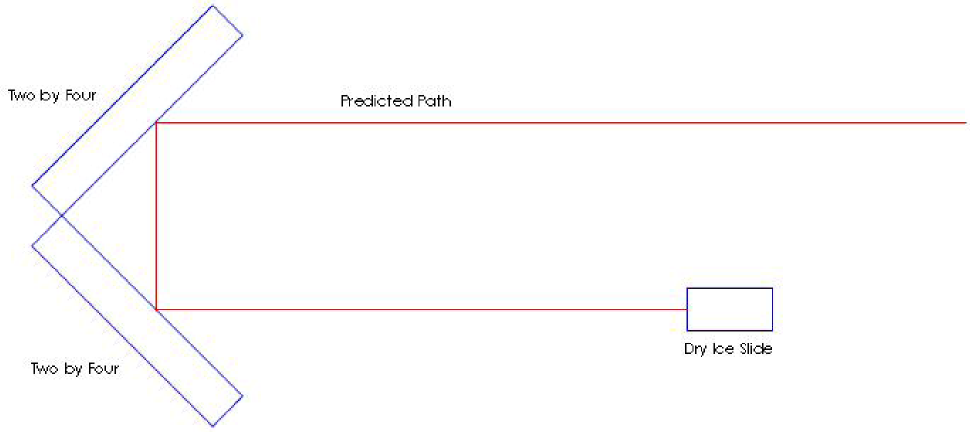
Clue #	Coordinates	Solution	Work Space
1	Initial clue	x coordinate = –1 y coordinate = 1 Go to: (–1,1)	
2	(–1,1)	x coordinate = 1 y coordinate = 4 Go to: (1,4)	
3	(1,4)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
4	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
5	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
6	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
7	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
8	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	
9	(__,__)	x coordinate = ___ y coordinate = ___ Go to: (__,__)	

Clue #	Coordinates	Solution	Work Space
10	(__,__)	x coordinate = ___ y coordinate = ___ **Go to:** (__,__)	
11	(__,__)	x coordinate = ___ y coordinate = ___ **Go to:** (__,__)	
12	(__,__)		

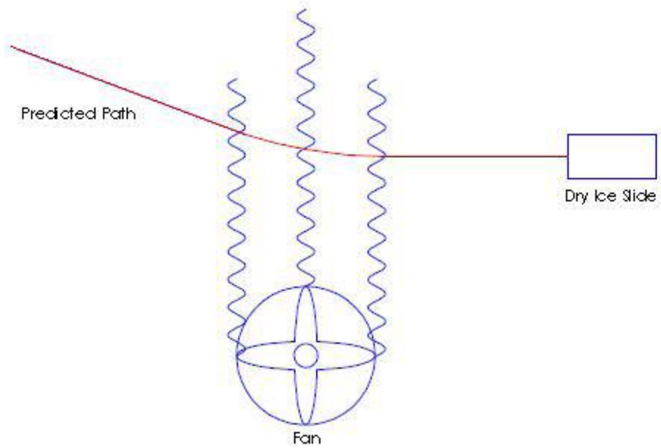
Write answer here: ____

Appendix 5: Obstacle Courses

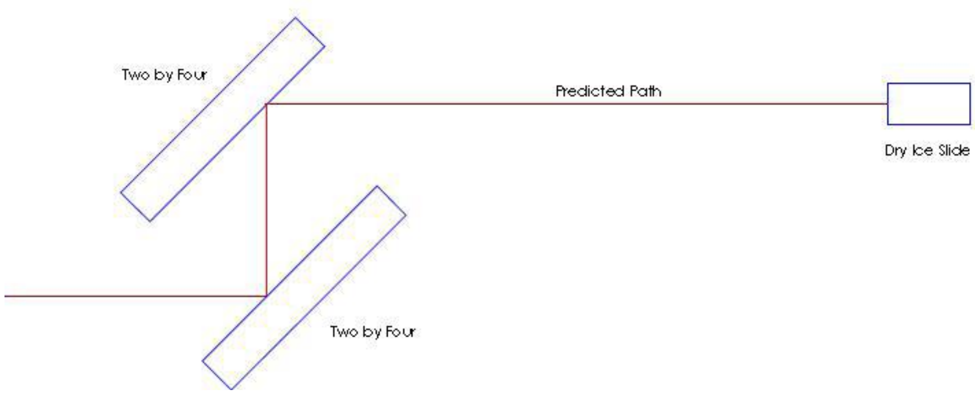
1. An obstacle demonstrating reflections off a barrier at 45 degrees



2. An obstacle demonstrating a particle moving under a constant force



3. An obstacle demonstrating reflections in a different manner



4. An obstacle demonstrating reflections and a constant force. Note that the outcome

of this obstacle is highly dependent on the strength of the fan used.

Appendix 6: Vector Hunt Challenge Question Solution

The solution to the challenge question "What is the Blue SPHERES' favorite thing to do?" is as follows: