

ZER 
RO  **B**  **TICS**
HS TOURNAMENT 2014



**Zero Robotics ISS Programming Challenge
High-School Tournament 2014:**



Conducting Optical Research on Nearby Asteroids (CORONA) GAME MANUAL V1.0

2014-Sep-06

To: Zero Robotics Teams
Re: CORONA program

Attention to all teams:

NASA has observed a recent spike in the number of asteroids knocked loose from the asteroid belt and into close proximity to Earth. Using highly sophisticated algorithms, they have used the mass of these space rocks to predict the composition of these asteroids with great accuracy. However, there exists amongst these rocks a small percentage that are of comparable mass percentages but do not fit the model. Eager geologists believe scientists may have discovered a new element and immediately rally for further study.

NASA calls upon its fellow scientists at MIT for a plan of action. With the help of these researchers, the joint CORONA program was born. CORONA (Conducting Optical Research On Nearby Asteroids) requires the use of MIT's most recent project, SPHERES, to capture visuals of the closest asteroids to Earth's atmosphere.

MIT researchers are fairly confident that they have stumbled upon something novel in these asteroids. However, some believe that there may actually be ice on these rocks and not a new element at all. The contrasting opinions lead to the creation of two separate SPHERES teams. Both teams will be using their respective satellites to take up-close photos of the asteroids at specific points of interest, as determined by NASA.

Soon after launch, however, NASA catches sight of incoming solar flares at the exact location where the SPHERES are intending to intercept the asteroids. The satellites risk imminent mechanical issues and loss of their stored photos if they impact with the solar flare while turned on. The MIT teams have to decide whether they want to use the shadow zone of the asteroid for protection or temporarily power down when a solar flare is near.

The choice is a tough one, but one that must be made. Each team is counting on its satellite to find visual proof of why either water or a new element exists on these space rocks. Proof of either would be invaluable, but conclusive evidence is essential.

As a SPHERES expert, your skills will be in high demand. GOOD LUCK!

A handwritten signature in blue ink, appearing to read 'ASA', with a stylized flourish extending from the end.

Alvar Saenz-Otero
MIT SPHERES Lead Scientist

Contents

1	GAME OVERVIEW	4
1.1	Game Layout	4
1.2	Satellite	5
1.2.1	ZR User API	5
1.2.2	Time	6
1.2.3	Fuel	6
1.2.4	Inter-satellite Communications	6
1.2.5	Code Size	6
1.3	Initial Position	6
1.4	Game play	7
1.4.1	Picture Taking	7
1.4.2	Memory Upgrades Packs	9
1.4.3	Solar Flares	10
1.4.4	Uploads	11
1.4.5	Collision	12
1.4.6	End of Game	12
1.5	Item Collection	12
1.6	Out of Bounds	14
2	SCORING	14
3	TOURNAMENT	14
3.1	The Leaderboard	15
3.1.1	Rating	15
3.1.2	Playing Matches	15
3.1.3	Competition Submissions	16
3.2	2D Practice Simulation Competition	16
3.3	3D Simulation Competition	16
3.4	Alliance Formation Event	16
3.5	Semifinal Simulation Competition	16
3.6	ISS Final Competition	16
3.6.1	Overview and Objectives	16
3.6.2	Competition Format	16
3.6.3	Definitions	16
3.6.4	Scoring Matches	16
4	SEASON RULES	16
4.1	Tournament Rules	16
4.2	Ethics Code	17
5	COSMOSPHERES API REFERENCE	18
6	LISTS OF FIGURES AND TABLES	19
6.1	List of Figures	19
6.2	List of Tables	19
7	REVISION HISTORY	19

1 Game Overview

Matches will be played between two SPHERES satellites, controlled by code written by two different teams/alliances. Satellites start the game facing the asteroid. Each team will attempt to take pictures of special points of interest (POI). These points of interest change location once every minute. The SPHERES can hold up to two pictures in its memory at a time. During the game, teams have the option of picking up a memory upgrade pack which allows players to store extra pictures before needing to upload. There are two zones around the asteroid which determine the amount of points each picture is worth. Points are received only after uploading the pictures stored in each satellite's memory. Throughout the game, there are solar flares that can corrupt the pictures in memory, and if precautions are not taken, can also damage the satellite. Satellites are completely safe only in the shadow zone.

1.1 Game Layout

The Zero Robotics High School Tournament 2014 begins with competitions in simulation. The competition mimics the operational volume available aboard the International Space Station, where the 3D finals will be conducted in January 2014. For the 2D game, the arena is a plane with only X and Y cardinal dimensions. The game arena encompasses the complete area where the SPHERES satellites can operate. However, the game is played in a smaller area called the Interaction Zone. If players leave the Interaction Zone, they may still be within the arena operational area, but they will be considered out of bounds.

The dimensions of the *Interaction Zone* are:

Table 1 Interaction Zone Dimensions

	2D	3D	Alliance
X [m]	[-0.64 : +0.64]	[TBD]	[TBD]
Y [m]	[-0.80 : +0.80]	[TBD]	[TBD]
Z[m]	0.0	[TBD]	[TBD]

Within the interaction zone there are three zones: Danger, Inner, and Outer zones.

When satellites enter the danger zone thrusters are automatically turned on to move them out of the danger zone. Inner and Outer zones only effect the amount of points a picture is worth.

The Shadow zone represents the area safe from Solar Flares.

Table 2 Zone Radii Positions

	2D	3D	Alliance
Danger min radius	.2	TBD	TBD
Danger max radius/Inner min radius	.25	TBD	TBD
Inner max radius/Outer min radius	.375	TBD	TBD
Outer max radius	.5	TBD	TBD

Table 3 Shadow Zone Dimensions

	2D	3D	Alliance
X [m]	[0.0 : +0.64]	[TBD]	[TBD]
Y [m]	[-0.2 : +0.2]	[TBD]	[TBD]
Z[m]	0.0	[TBD]	[TBD]

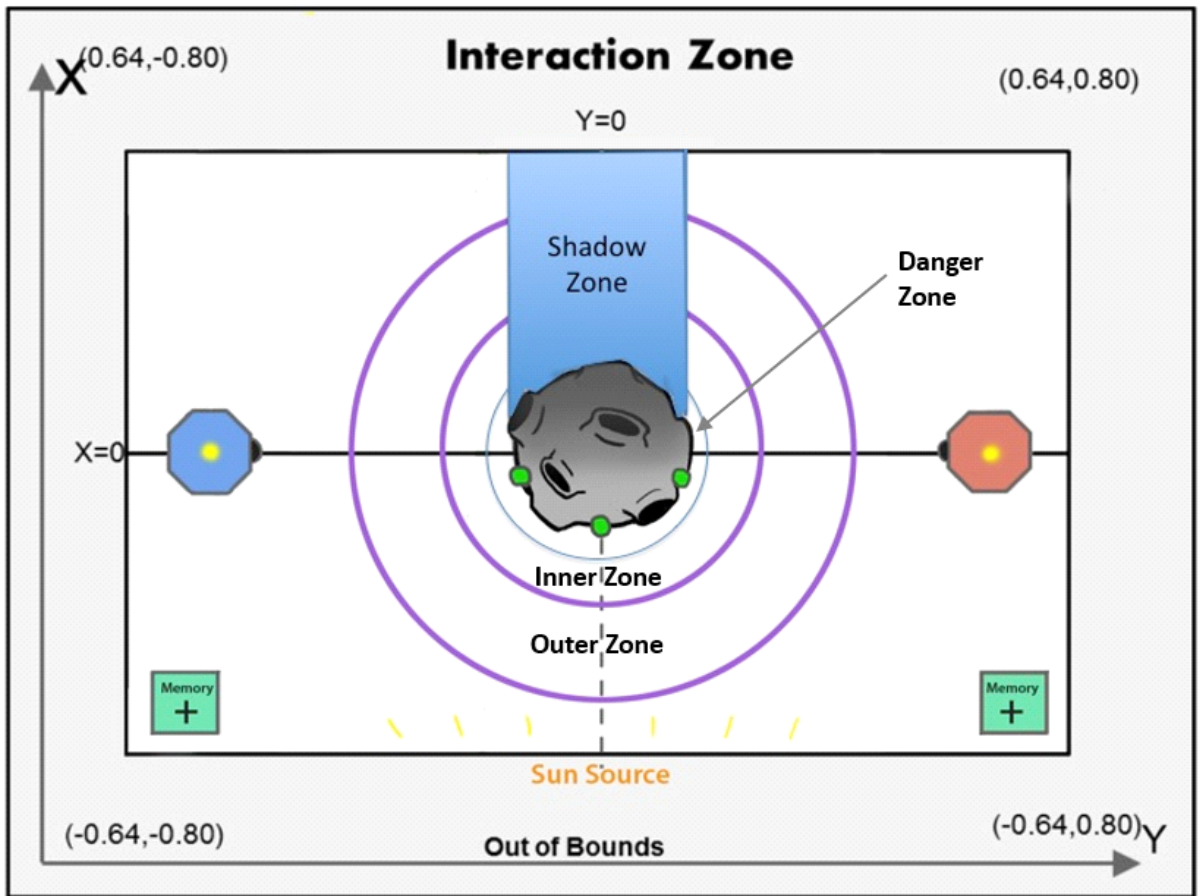


Diagram not to scale

Figure 1 Game Overview

1.2 Satellite

Each team will write the software to command a SPHERES satellite to move in order to complete the game tasks. A SPHERES satellite can move in all directions using its twelve thrusters. The actual SPHERES satellite, like any other spacecraft, has a fuel source (in this case liquid carbon dioxide) and a power source (in this case AA battery packs). These resources are limited and must be used wisely. Therefore, the players of Zero Robotics are limited in the use of real fuel and batteries by virtual limits within the game. This section describes the limits to which players must adhere to wisely use real SPHERES resources.

1.2.1 ZR User API

The non-game-specific functions used to control the SPHERES satellite in Zero Robotics can be found in a document titled “ZR User API” on the Tutorials webpage under “Other Resources”. There is also a link to this document at the end of section 5. Game specific functions, along with a link to the standard ZR User API functions, are also provided in section 5 of this document.

1.2.2 Time

Players have 240s to take and upload as many photos as possible. After 240s scores will be final and compared.

1.2.3 Fuel

Each player is assigned a virtual fuel allocation (Table 3), which is the total sum of fuel used in seconds of individual thruster firing. Once the allocation is consumed, the satellite will not be able to respond to SPHERES control commands. It will fire thrusters only to avoid leaving the Interaction Zone or colliding with the other satellite.

Table 4 Fuel Allocation

	2D	3D	Alliance
Fuel Allocation [s]	120s	TBD	TBD

The virtual fuel allocation is consumed any time the thrusters are fired. Potential reasons include:

- Motion initiated by player
- Motion initiated by the SPHERES controller to avoid leaving the Interaction Zone (see section 1.6)
- Motion initiated by the SPHERES controller to avoid a collision with the other satellite

1.2.4 Inter-satellite Communications

The satellites have the ability to communicate with each other using binary messages. The API functions `sendMessage` and `receiveMessage` may be used to send data between the satellites. The bandwidth available to the satellites is as follows: (adding function for cooperation)

Table 5 Inter-satellite communications bandwidth

	2D	3D	Alliance
Message size	Unsigned char	TBD	TBD

Note: bandwidth allocation may either increase or decrease as the tournament progresses.

1.2.5 Code Size

A SPHERES satellite can fit a limited amount of code in its memory. Each project has a specific code size allocation. When you compile your project with a code size estimate, the compiler will provide the percentage of the code size allocation that your project is using. Formal competition submissions require that your code size be 100% or less of the total allocation.

1.3 Initial Position

Each satellite starts on the X axis on opposite sides of the asteroid.

The SPHERES satellites are deployed at:

Table 6 SPHERES Satellite Deployment Locations

	2D	3D	Alliance
Red			
X [m]	0.0	TBD	TBD
Y [m]	-0.6	TBD	TBD
Z[m]	0.0	TBD	TBD
Blue			
X [m]	0.0	TBD	TBD
Y [m]	0.6	TBD	TBD
Z[m]	0.0	TBD	TBD

PlayerID

Users will identify themselves as “playerID = 0” and opponents as “playerID = 1” for all games, whether or not they are the red SPHERES satellite or the blue one.

1.4 Game play

For all the games the interaction zone has 2D games, there are 4 possible points of interest. The 4 points of interest will be randomly placed on the surface of the asteroid on the half closest to the sun and every 60 seconds a random 2 of these points will be visible.

Table 7 Asteroid and SPHERES Measurements

	2D	3D	Alliance
Asteroid radius (m)	0.2	TBD	TBD
SPHERE radius (m)	0.11	TBD	TBD

1.4.1 Picture Taking

For taking pictures in 2D, there are two possible orbits from which the satellite can take pictures: a low orbit in the Inner Zone and a high orbit in the Outer Zone (dimensions listed above in Table 2). In order for a picture to be valid there are three qualifications:

- The satellite must be in the Inner or Outer Zone
- The angle between the line connecting the center of the sphere and the POI, and the radius of the asteroid must be smaller than the Max Angle for the zone the satellite is in
- The satellite must be facing the POI (with a tolerance of .05).

Table 8 Max Angle of Zones (in radians)

	2D	3D	Alliance
Inner Zone	0.8	TBD	TBD
Outer Zone	0.4	TBD	TBD

For the satellite to be considered in orbit the center point of the satellite must be within the bounds of the respective Zone (Figure 2). Scoring will be determined by which orbit the satellite is in; more points will be awarded for pictures taken in the high orbit. The POI must be within field of view for the pictures to score points. The field of view is different for the two orbits (Figure 2).

Table 9 Picture values in Inner and Outer Zones

	2D	3D	Alliance
Inner Zone	2	TBD	TBD
Outer Zone	3	TBD	TBD

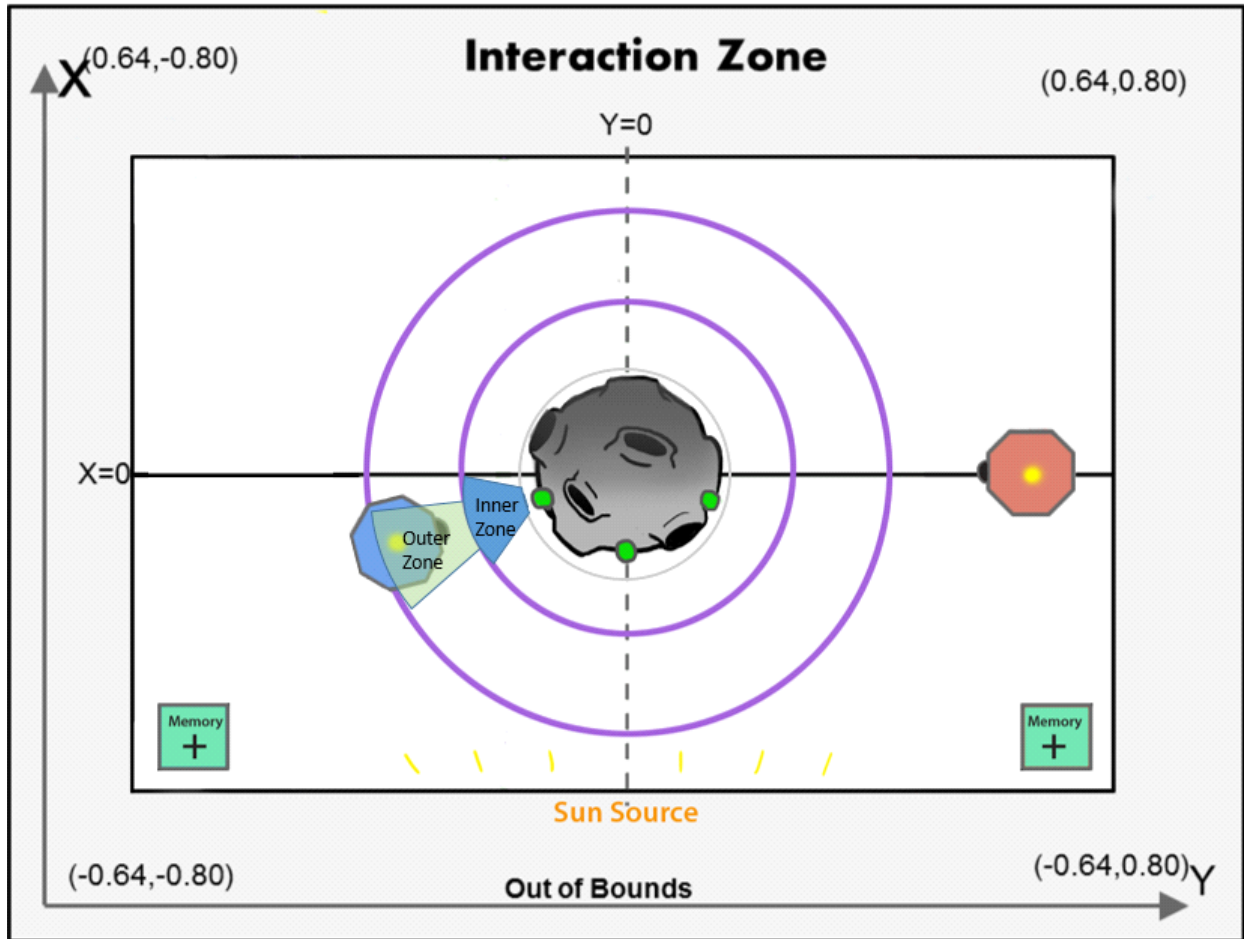


Diagram not to scale

Figure 2 Picture Taking Zones

The satellite can only store 2 photos at a time unless it has obtained an upgrade from a memory pack (see 1.4.2). The camera will be disabled once the photo storage is full until the pictures are uploaded.

Pictures of a single point of interest cannot be taken from the same orbit within each 60s window. In 2D mode, there are four possible pictures to take in each 60s window, a picture of each of the points can be taken from each orbit once. These limitations are reset after each 60s window.

In order to take a picture, the point of interest must be within your field of view. This means that that your opponent cannot be blocking your field of view (Figure 3).

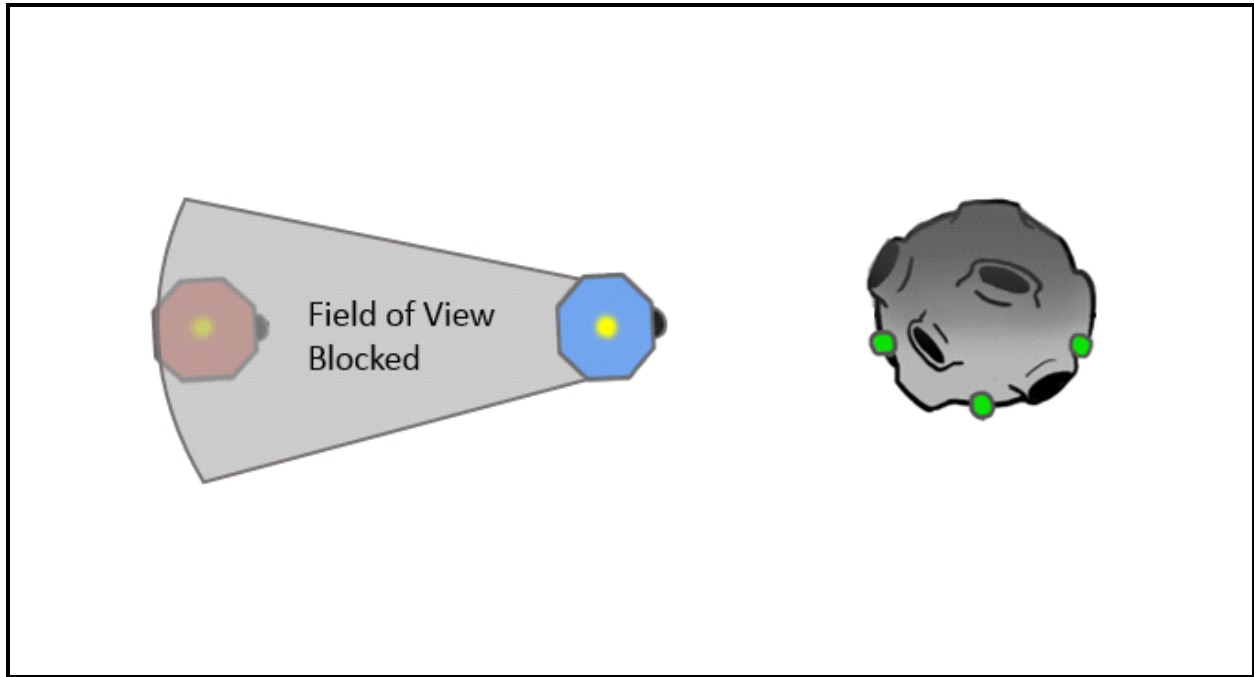


Diagram not to scale

Figure 3 Blocking Opponent’s Field of View

To take a pictures:

- Camera must pointing at POI
- POI must be in field of view (an uninterrupted line can be made from camera to POI)
- Game function `void takePicture()` must be called

1.4.2 Memory Upgrade Packs

Memory Upgrade packs allow players to store more pictures on their satellite. Memory Upgrade packs are located on the corners of the sunny side of the interaction zone. Memory packs will remain in play throughout game play. Once a memory pack has been taken by a satellite, it is then in the possession of that satellite and may not be picked up by the other satellite. A satellite can hold up to 2 memory packs. Each pack gives the satellite enough memory to hold 1 more photo.

Table 10 Memory Upgrade Pack Locations

	2D	3D	Alliance
Memory Pack 1(ID = 0)			
X [m]	-0.5	TBD	TBD
Y [m]	-0.6	TBD	TBD
Z[m]	0.0	TBD	TBD
Memory Pack 2(ID = 1)			

X [m]	-0.5	TBD	TBD
Y [m]	0.6	TBD	TBD
Z[m]	0.0	TBD	TBD

To collect Memory Upgrade packs see 1.5 ITEM COLLECTION.

1.4.3 Solar Flares

Solar Flares pose a danger to satellites. Solar Flares occur randomly every 70s starting 30 seconds after the start of the game. Each Solar Flare lasts 3 seconds. All satellites exposed to the solar radiation for any period of time lose the photos stored in their memory. Satellites exposed to the solar radiation will lose 1 point every second they are exposed to the solar radiation, unless the satellites are powered off prior to exposure with the solar radiation.

While the satellite is powered off, players:

- Reduce Damage (lose half the points of satellites powered on)
- Lose any pictures currently stored on satellite
- Must wait 5 seconds for their instruments to turn on and warm up
- Cannot take pictures
- Cannot stop their satellites from drifting (collisions may occur)

To power off:

- Game function `void turnOff()` must be called

The other way to prevent damage is to seek shelter in the shadow zone. Players are protected from solar radiation in the shadow zone, preserving pictures and points. To be deemed safe, the center point of the satellite must be in the shadow zone.

Players will have a 30 second warning before a solar flare arrives.

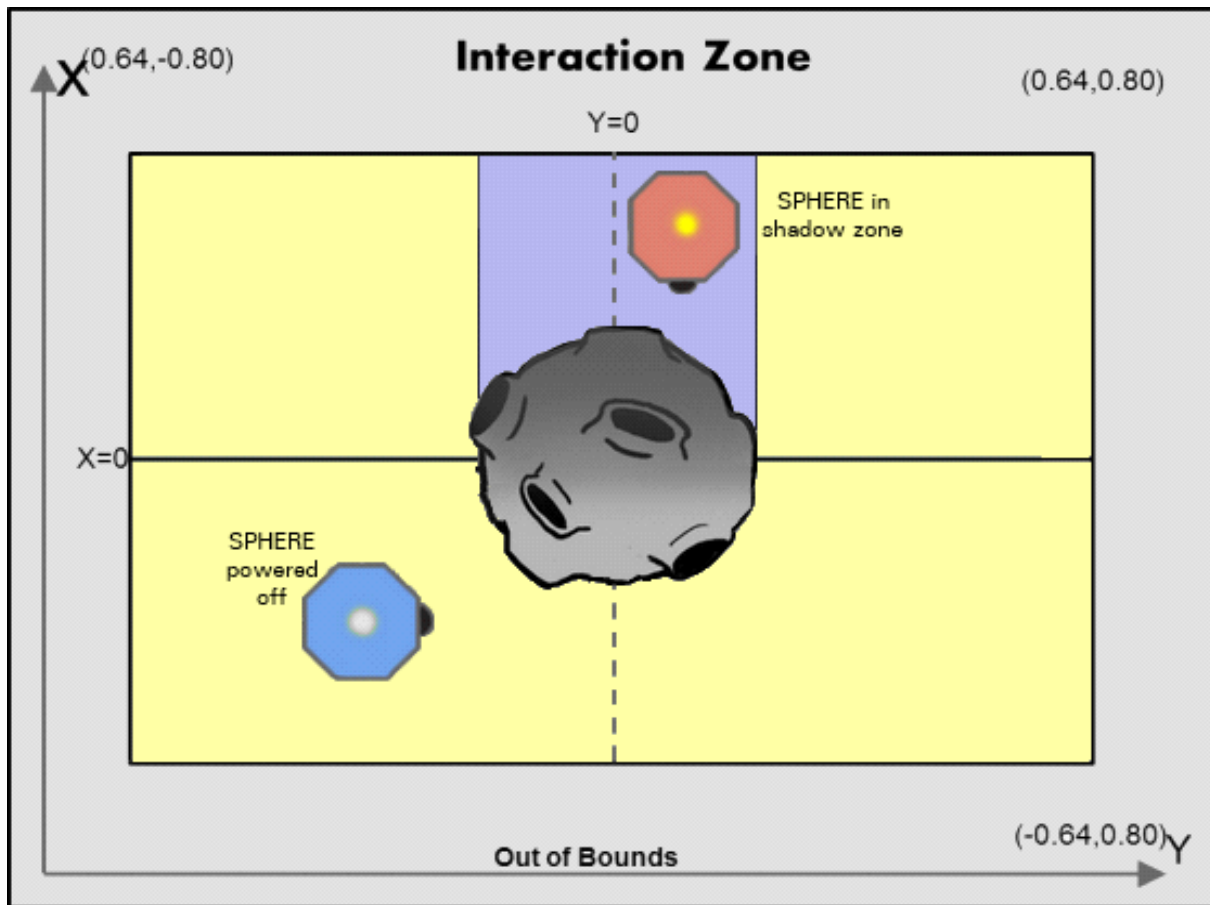


Diagram not to scale

Figure 4 Solar Flare

1.4.4 Upload

Once the satellite's memory is full, the pictures must be uploaded. Pictures must be uploaded to score full points for taking the picture. Without uploading, only 0.01 points will be awarded for successfully taking a photo.

During the upload process the camera will be disabled for 3 seconds.

There is a Discover Bonus for the first satellite to upload a picture of a specific point of interest within a cycle. The Discover Bonus is +0.5. The Discover Bonus resets with each 60s POI rotation cycle.

To upload photos:

- The center of the SPHERE must be either outside of the Danger Zone and both Picture Taking, or in the Shadow Zone
- The game function `void uploadPicture()` must be called

1.4.5 Collisions

While it is not possible to collide with the other satellite, collision with the asteroid is possible. If the center point of your satellite enters the Danger Zone, it is considered a collision with the asteroid.

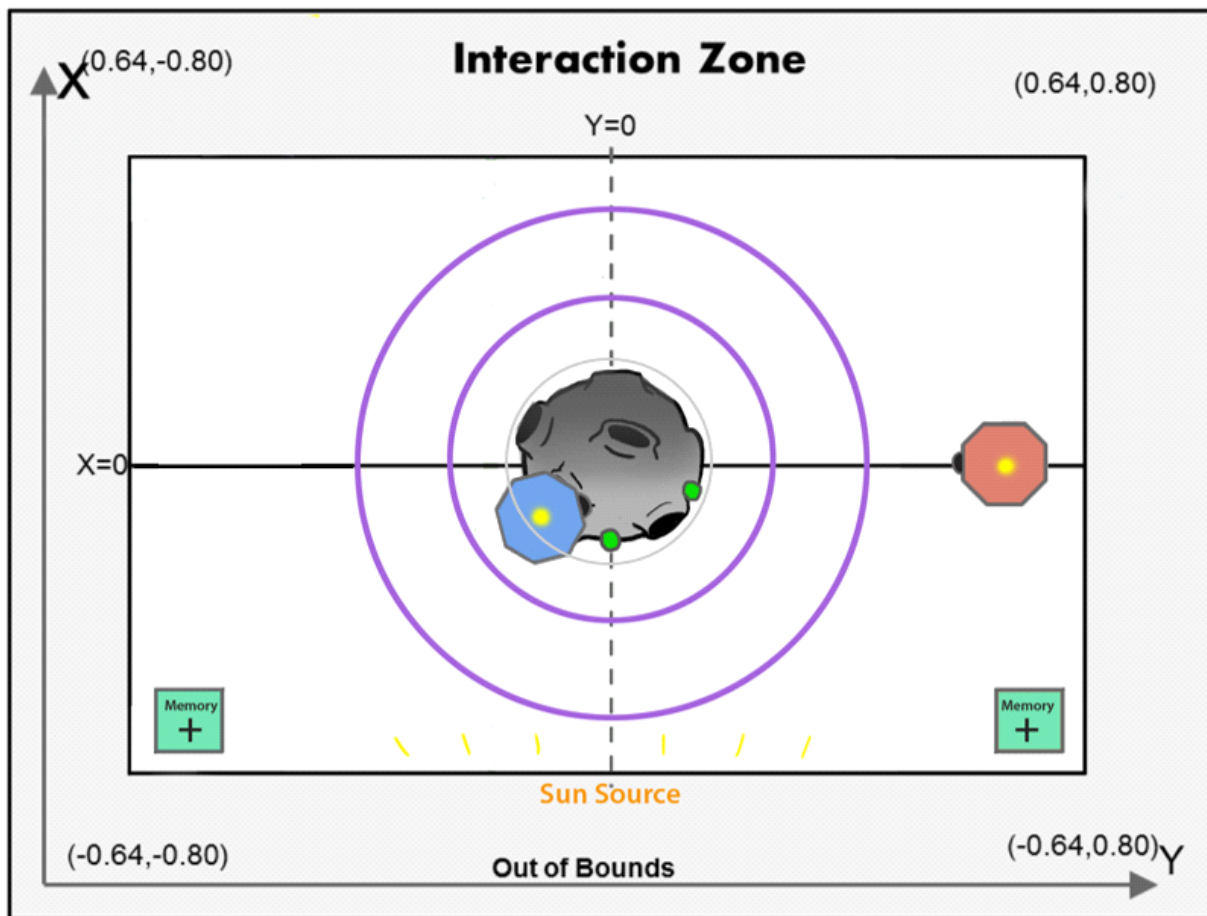


Diagram not to scale

Figure 5 Asteroid Collision

If the satellite crashes into the asteroid:

- The satellite will come to a full stop
- The satellite can only move away from the asteroid
- There will be a fuel penalty equal to 2 thruster seconds of fuel per second in the asteroid

1.4.6 End of Game

The game ends when time runs out.

1.5 Item Collection

To increase the memory capacity of the satellites, teams have the opportunity to collect memory upgrade packs found in the 2 corners of the interaction zone closest to the sun.

Table 10 Memory Upgrade Pack Locations (repeated)

	2D	3D	Alliance
Memory Pack 1			
X [m]	-0.5	TBD	TBD
Y [m]	-0.6	TBD	TBD
Z[m]	0.0	TBD	TBD
Memory Pack 2			
X [m]	-0.5	TBD	TBD
Y [m]	0.6	TBD	TBD
Z[m]	0.0	TBD	TBD

In order to pick up an item, you need to perform a spinning maneuver (see Figure 7). The steps to collect the Memory Upgrade packs are:

- Position the satellite within 0.05m of the item's center.
 - The satellite's velocity must be less than 0.01m/s.
 - The satellite's angular velocity must start at less than 2.3°/s.
- Rotate the satellite >90° along any axis. Do not attempt to rotate faster than 80°/s.

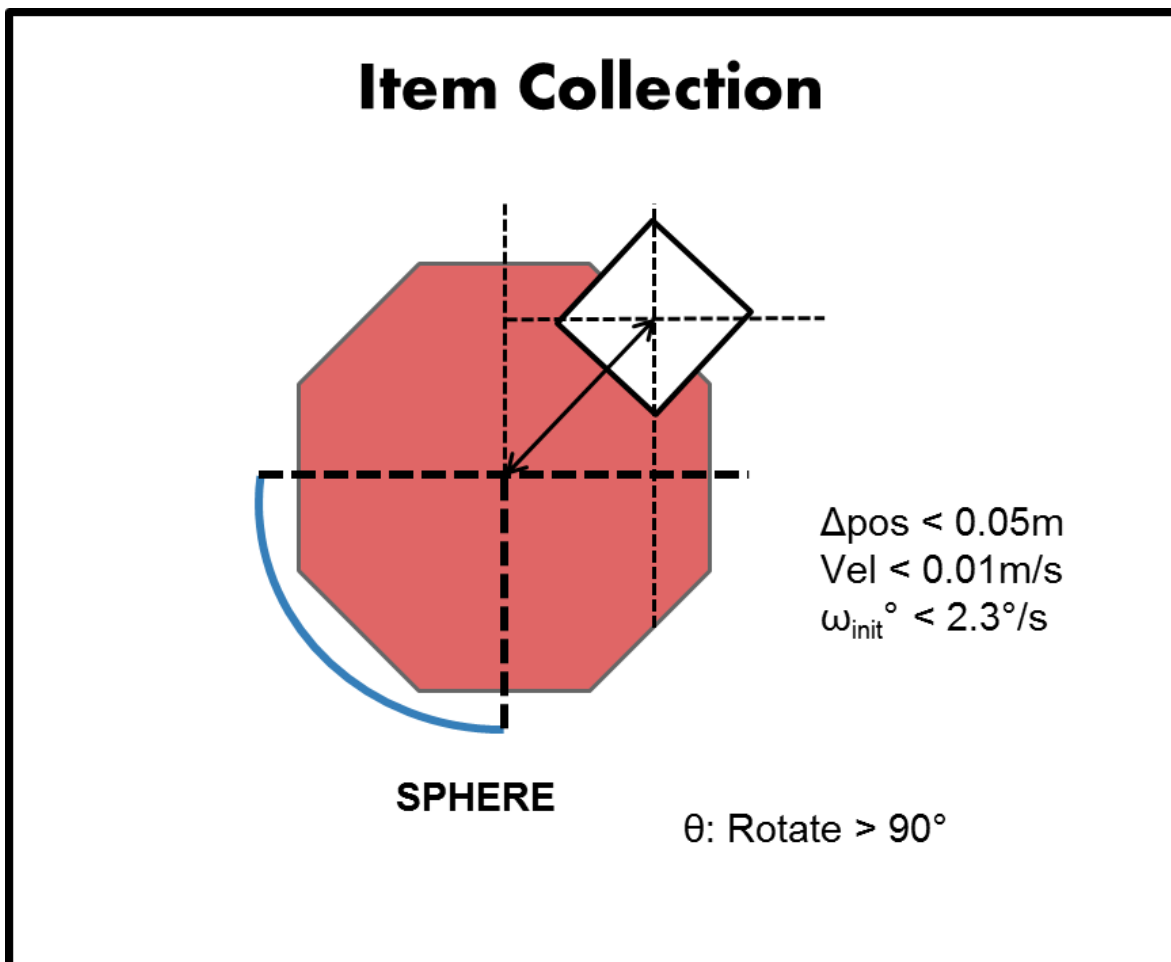


Figure 6 Maneuver to Collect Item

1.6 Out of Bounds

You must remain within the boundaries of the Interaction Zone to avoid a fuel penalty.

If you exit out of bounds, the SPHERES controller will override your commands and force the satellite to stop its motion in the direction that would continue to push it out of bounds (other directions are not affected). The fuel used to stop this motion will be charged to your fuel usage. There is an additional fixed penalty of **2 thruster-seconds** (4% of total initial fuel) for each second spent out of bounds.

2 Scoring

Your score is based on the pictures you take and upload minus the damage obtained from solar flares and collisions. Pictures are worth different point values depending on which orbit they were taken from. The score will be totaled from the number and location of the photos taken. The seconds exposed to the solar radiation will subtract points from your score.

The scoring calculation is as follows:

Table 11 Point Values

	2D	3D	Alliance
Non Uploaded photos	0.01	TBD	TBD
Uploaded: Taken in Inner Zone	2	TBD	TBD
Uploaded: Taken in Outer Zone	3	TBD	TBD
Exposed to Solar radiation (per sec)	-1	TBD	TBD

3 Tournament

A Zero Robotics tournament consists of several phases called *competitions*. The following table lists the key deadlines for the 2013 tournament season:

Table 12 Tournament Key Dates

Date (2014)	Event
Sep 6 (Sat)	Kick-off webcast
Sept 26 (Fri)	2D Simulation Competition Deadline (end of practice)
Oct 31 (Fri)	3D Simulation Competition Deadline
Nov 8 (Sat)	Alliance Formation Event
Dec 5 (Fri)	Alliance Simulation Submission Deadline
Dec 19 (Fri)	ISS Code Submission Deadline
mid-Jan TBA	ISS Final Competition

The rankings in each competition are determined by a *Leaderboard*, described below. The 2D Simulation Competition is *not* an elimination round; everyone who submits code to the 2D Simulation Competition advances to the 3D Competition. At the end of the 3D Simulation Competition an elimination round takes place, with the top 54 US / 36 EU ranking teams moving to the Alliance phase of the Tournament. The top 54 US / 36 EU teams will then form alliances of three (3) teams each: 18 US and 12 EU alliances will be formed. The alliances will compete against each other, with the top 9 US and 6 EU alliances moving to the ISS Final Competition.

3.1 The Leaderboard

This year's tournament uses a continuously updated ranking system called *The Leaderboard*. The Leaderboard uses a system similar to the Elo rating system for chess players called Whole History Ranking, as well as ideas from the *TrueSkill*® rating system used for the Xbox Live gaming platform. The Leaderboard tracks all matches a team has played against other players in the course of the competition and creates a rating based on the outcomes. At the end of each competition phase, the final standings on the Leaderboard will determine which teams advance to the next phase.

3.1.1 Rating

A team's standing in a competition is determined by a value called rating. Teams with higher ratings are have higher standing. In any match between two teams A and B, the probability of a match outcome is estimated as a function of the ratings of the two teams:

$$P_{A\text{wins}} = \frac{e^{R_A}}{e^{R_A} + e^{R_B}}$$

where R_A is the rating of Team A, and R_B is the rating of Team B. Ratings are determined using Bayes rule and a deviation over time in probability of rank.

Players of nearly equivalent ratings have roughly equal probability of winning a match, while differences in rating result in one of the teams being more heavily favored. After a series of matches are played, the ratings are recalculated to account for new outcomes. Under Whole History Ranking, all ratings for all time periods are recalculated using the results of new matches. Therefore, ratings from past dates may change. More recent updates, however, are more accurate portayals of ratings at past times.

There are several factors that affect a team's rating:

- **Match Outcomes:** A team that consistently wins matches will usually have a higher rating.
- **Opponent Rank** Winning against a higher ranked team will usually improve a team's rating.
- **Other Match Outcomes** The Leaderboard takes into account all matches played by all teams. Even if two teams do not have a direct encounter, their match outcomes will have an effect as they filter through third parties.

Every day during the competition, teams will play the 10 teams above and the 10 teams below them on the leaderboard, rather than all other teams. The Leaderboard calculates ratings daily from the beginning of a competition until the submission deadline. All matches during the competition period count toward the rating in the competition. The Leaderboard ratings are cleared at the start of each competition.

The Leaderboard will throw out 0-0 ties, and they will not affect ratings.

US and EU teams compete against each other on a single Leaderboard. Once each competition phase ends, the Leaderboard rankings are used to create separate ranking lists for US and EU teams.

3.1.2 Playing Matches

Twenty (20)Matches are played daily. To play, the team must have submitted code in the competition. The most recently submitted code will be used for all matches. Ten (10) of the twenty (20) matches will be as the Blue sphere, the other ten as the Red sphere.

3.1.3 Competition Submissions

Once a competition is open, teams may make as many code submissions as they want, understanding that all matches played with those submissions will affect their ranking. There is no cap on the number of submissions a team may make in a day, but you cannot make two submissions less than 15 minutes apart. Submissions for the competitions will close at 11:59 PM EDT on the deadlines shown in Table 15. Remember that although the 2D competition is a practice round, teams are required to make at least one submission before the deadline to be entered in the 3D round.

3.2 *2D Practice Simulation Competition*

All teams that complete a valid registration are eligible to participate in the 2D practice simulation competition.

3.3 *3D Simulation Competition*

TBD

3.4 *Alliance Formation Event*

TBD

3.5 *Semifinal Simulation Competition*

TBD

3.6 *ISS Final Competition*

TBD

3.6.1 Definitions

TBD

3.6.2 Scoring Matches

TBD

4 Season Rules

4.1 *Tournament Rules*

All participants in the Zero Robotics High School Tournament 2014 must abide by these tournament rules:

- The Zero Robotics team (MIT / Top Coder / Aurora) can use/reproduce/publish any submitted code.
- In the event of a contradiction between the intent of the game and the behavior of the game, MIT will clarify the rule and change the manual or code accordingly to keep the intent.
- Teams are expected to report all bugs as soon as they are found.
 - A “bug” is defined as a contradiction between the intent of the game and behavior of the game.
 - The intent of the game shall override the behavior of any bugs up to code freeze.

- Teams should report bugs through the online support tools. ZR reserves the right to post any bug reports to the public forums (If necessary, ZR will work with the submitting team to ensure that no team strategies are revealed).
- Code and manual freeze will be in effect 3 days before the submission deadline of a competition.
 - Within the code freeze period the code shall override all other materials, including the manual and intent.
 - There will be no bug fixes during the code freeze period. All bug fixes must take place before the code freeze or after the competition.
 - The code is finalized at the ISS Final Competition freeze (unless there is a critical issue which will affect the final tournament, including lessons learned from ground hardware testing and simulation.)
- Game challenge additions and announcement of TBA values in the game manual may be based on lessons learned from earlier parts of the tournament.

4.2 Ethics Code

- The ZR team will work diligently upon report of any unethical situation, on a case by case basis.
- Teams are strongly encouraged to report bugs as soon as they are found; intentional abuse of an unreported bug may be considered as unethical behavior.
- Teams shall not intentionally manipulate the scoring methods to change rankings.
- Teams shall not attempt to gain access to restricted ZR information.
- We encourage the use of public forums and allow the use of private methods for communication.
- Vulgar or offensive language, harassment of other users, and intentional annoyances are not permitted on the Zero Robotics website.
- Code submitted to a competition must be written only by students.

5 CosmoSPHERES API Reference

The functions in this section are called as members of the game object, that is, `game.functionName(arguments)`:

Name	Description
<code>void getNextFlare()</code>	Returns number of seconds until next flare
<code>int getMemoryFilled()</code>	Returns number of pictures currently saved in camera
<code>int getMemorySize()</code>	Returns current limit of picture storage
<code>void takePic(float zrState[12], float poiPos[3])</code>	Takes picture and stores in satellite memory. Camera disabled if picture fills last memory slot
<code>void uploadPic()</code>	Uploads pictures from satellite, disables camera for 3 seconds
<code>int numActivePOIs()</code>	Always returns 2 in 2D
<code>void getPOILoc(float pos[3], int id)</code>	Returns location of each visible POI
<code>float getScore()</code>	Returns player's score
<code>float getOtherScore()</code>	Returns opponent's score
<code>void turnoff()</code>	Turns off satellite to protect from solar flare. Satellite will drift.
<code>void turnOn()</code>	Begins process to turn satellite on. Takes 3 seconds. (unless game just beginning)
<code>float getFuelRemaining()</code>	Returns remaining fuel
<code>bool hasMemoryPack(int playerId, int packID)</code>	Returns true if specified player has specified memory pack
<code>**bool alignLine()</code>	Returns true if the SPHERE can take a good picture of the POI (i.e. within the picture taking zone)
<code>void sendMessage(unsigned char inputMsg)</code>	Sends inputMsg to other satellite
<code>unsigned char receiveMessage()</code>	Returns the most recent message sent by other satellite

A guide to the standard Zero Robotics API functions is available here:

http://www.zerorobotics.org/documents/10429/374963/ZR_user_API.pdf

