AsteroidBee 2026 — Game Rules Reference

For teachers. Everything on this page was read directly from the game's source code (`Main/Games/ZRMS2026_GAME`, identical engine to the original 2016 SPYSpheres game) and matches the tournament "Game Intel" page on the Zero Robotics website.

The story

A NASA satellite has broken apart in low Earth orbit. Two rival corporations each operate one SPHERES satellite. Your satellite scores points by **photographing the opponent** and **collecting debris items**, while managing **energy** between the sunlit and shadowed halves of the arena. Each match lasts **180 seconds (3 minutes)** and the satellites are fully autonomous; the students' program is the only pilot.

The arena

- The playing field is a box: **X from −0.64 to +0.64 m**, **Y from −0.80 to +0.80 m** (the game is 2D; Z stays 0). Units everywhere are meters.
- **Blue starts at (0.4, −0.6)** and **Red starts at (−0.4, −0.6)**, both in the Light Zone.
- If a satellite drifts out of bounds the game automatically pushes it back in (no points penalty). Built-in collision avoidance keeps the satellites from crashing into each other.

Light and Dark Zones

- At the start, the **Y < 0 half is LIGHT** and the **Y > 0 half is DARK**.
- The halves **swap at t = 60 s, and again at t = 150 s**. (The block `get light switch time` tells you how many seconds until the next swap.)
- **In the Light Zone**: your solar panels recharge **+0.5 energy per second**, but the opponent can photograph you.
- **In the Dark Zone**: you cannot be photographed (you're invisible to their camera), but you do not recharge.

Energy

Fact	Value
Energy capacity / starting energy	5.0
Recharge rate (Light Zone only)	+0.5 per second
Cost to take a picture (even a failed one)	1.0
Cost to preview a picture's value (`get pic points`)	0.1
Cost to check opponent's energy	free
Thruster cost	tiny (≈0.0003 per second of thrust)

If you hit 0 energy you can't take pictures until you recharge: "your satellite goes dark."

Photographs (the main way to score)

A picture taken with `take pic` **succeeds** only if ALL of these are true:

1. The camera is on. It shuts off for **3 seconds** after every attempt (`is camera on`).
2. You are **facing the opponent**, within about 30° (`is facing other`).

3. The opponent is **not in the Dark Zone**.
4. The opponent does **not** have an active mirror (and you don't either).
5. You are **at least 0.5 m away** from the opponent.

Points = `2.0 + 0.1 / (distance − 0.4)` → **3.0 points at exactly 0.5 m**, sliding down toward 2.0 as you get farther away. Plus a +0.1 bonus for a successful picture and +0.01 just for attempting. So: a good picture is worth roughly **2–3 points**, and being close (but ≥ 0.5 m) is better.

A **failed** attempt still costs 1.0 energy and still triggers the 3-second cooldown. This is why "check before you shoot" is a core strategy lesson.

Items (debris)

There are 9 items at fixed, known locations (see diagram). **To pick one up: park on top of it**, within **5 cm**, moving slower than **0.01 m/s**. The satellite's normal "move to position" control does this naturally: tell it to go to the item's location and let it come to a stop there. First satellite to do so gets the item.

Item ids	Type	On the arena map (Fig.01 style)	Effect	Starts in
4, 5, 6	Score+ (the "asteroids")	**squares**	**+1.5 points** each	Dark Zone
0, 1, 2, 3	Energy Pack	**circles**	refill energy to 5.0	0 & 1 in Light, 2 & 3 in Dark
7, 8	Mirror	**diamonds**	gain 1 mirror	Light Zone

Item locations (meters):

id	type	x	y
0	Energy	0.25	−0.40
1	Energy	−0.25	−0.40
2	Energy	0.25	0.40
3	Energy	−0.25	0.40
4	Score	0.00	0.60
5	Score	0.40	0.60
6	Score	−0.40	0.60
7	Mirror	0.60	−0.70
8	Mirror	−0.60	−0.70

(Red's view of the field is mirrored across X automatically by the game. Students do not need to handle this; item ids work the same for both satellites.)

Mirrors (defense)

- You start with 0 mirrors; collect items 7/8 to hold one (`get num mirrors held`).
- `use mirror` activates a held mirror for **24 seconds** (`get mirror time remaining`).
- While your mirror is active, opponent photographs of you are worth **0 points**, and they still pay the 1.0 energy and the cooldown. (While it's active you can't take pictures yourself, so time it defensively.)

Fuel

Separate from energy: each satellite has **60 seconds of total thruster firing time** (`get remaining fuel`). Constant fidgeting wastes it; a satellite out of fuel drifts. For these assignments fuel only becomes a real constraint in the final, full-match strategy.

Timing summary

Event	Time
Match length	180 s
Student `loop()` runs	**once per second, every second, all match**
Light/Dark swap	t = 60 s and t = 150 s
Camera cooldown after any attempt	3 s
Mirror duration	24 s

What the game does NOT have as blocks

These exist only in the C API and are **not needed for any assignment in this series**: `sendMessage` / `receiveMessage` (1-byte radio to the opponent), `posInLight` / `posInDark` (check an arbitrary coordinate), `getNumItem`, and the memory-slot functions (pictures upload automatically in this version). Everything used by the assignments has a real block; see the Block Dictionary.