

API Functions - Updated Jul 18, 2025

The following tables list the API functions available for this year's game. The functions are organized into 3 main sections:

1. Getters: Functions to get values of variables such as object and robot positions.
2. Movement: Functions to move the Astrobees around the playing field.
3. Operations: Functions to do actions like plant and harvest crops
4. Debugging: Function to print messages and variable names to the terminal.

These functions can be called in the C text editor using the syntax "game.Function()". E.g. to call the MoveToVoid function:

```
1 void loop() {  
2     game.MoveToVoid(0,0.4);  
}
```

The functions can be called in the graphical editor using the corresponding Blockly blocks.

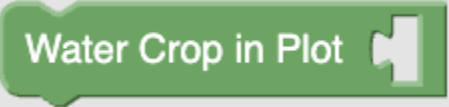
The table provides a detailed description of each function's behaviour in the right column and the input and output variables in the left column.

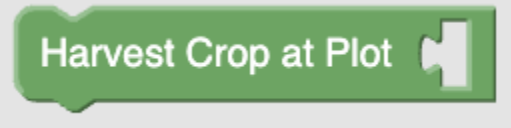
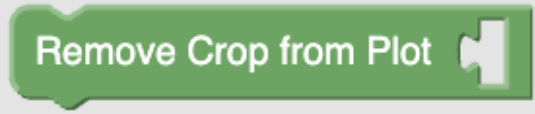
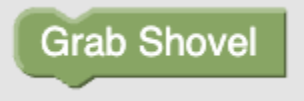
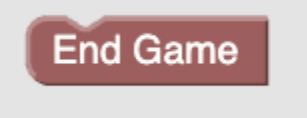
Movement

API Function	Description
<pre>bool game.MoveTo(float x,float y,float z)</pre> 	Moves Astrobees to a particular location in the playing field, specified by (x,y,z) coordinates. To be successful, destination coordinates must be within the boundary. Returns True if the movement is valid. Returns False if movement is invalid.
<pre>void game.MoveToVoid(float x, float y,float z)</pre> 	Moves Astrobees to a particular location in the playing field, specified by (x,y,z) coordinates. To be successful, destination coordinates must be within the boundary. Does not return an output.
<pre>bool game.MoveToHome()</pre> 	Move Astrobees to the home position (0,0,0). Similar to MoveTo function above. Returns True if the movement is valid. Returns False if movement is invalid.

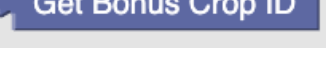
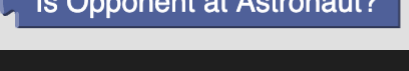
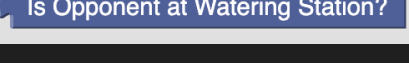
<pre>void game.MovePlot(int plotID)</pre> 	Move Astrobees to the location of a garden plot, specified by plotID. Options: 1,2,or 3. Does not return an output.
<pre>void game.MoveWatering()</pre> 	Move Astrobees to the watering zone (0,0,0). Note that only one robot can enter the watering zone at a time. If another robot is currently using the zone, the movement will fail, and Astrobees will continue with the next command. Does not return an output.
<pre>void game.MoveAstronaut()</pre> 	Move Astrobees to the astronaut zone (0,1.5,±0.5). Note that only one robot can enter the astronaut zone at a time. If another robot is currently using the zone, the movement will fail, and Astrobees will continue with the next command. Does not return an output.
<pre>void game.LeaveAstronaut()</pre> 	Move Astrobees to leave the astronaut zone, to the exit location (0,1,±0.5). Does not return an output.

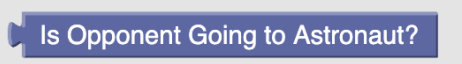
Operations

API Function	Description
<pre>void game.PlantCrop(int plotID, int cropID)</pre> 	Plant a particular crop in a particular garden plot. plotID is ID of the plot (1, 2, or 3) cropID is ID of the crop: 1=Tomato, 2=Cabbage, 3=Strawberry, 4=Melon. The robot must be at the correct location of the plot to successfully plant.
<pre>void game.WaterCrop(int plotID)</pre> 	Water the crop at a particular garden plot, specified by plotID. The robot must be at the correct location of the plot to water successfully.

<pre>void game.HarvestCrop(int plotID)</pre> 	Harvest the crop at a particular garden plot, specified by plotID. The robot must be at the correct location of the plot to harvest successfully. If a crop is harvested too early, no points are earned and the crop disappears. Make sure the crop has been watered sufficiently before harvesting.
<pre>void game.RemoveCrop(int plotID)</pre> 	Remove a failed crop at a particular garden plot, specified by plotID. The robot must be at the correct location of the plot and be holding the shovel to remove successfully. After removal, new crops can be planted in the garden plot.
<pre>void game.FillWateringCan()</pre> 	Fill the watering can. The robot must be at the correct location of the watering zone to fill successfully. This action takes 4 seconds to complete.
<pre>void game.GrabShovel()</pre> 	Pick up the shovel.
<pre>void game.DropShovel()</pre> 	Drop the shovel.
<pre>void game.EndGame()</pre> 	Ends the current simulation and stops the timer.

Getters

API Function	Description
<pre>float game.GetTime()</pre> 	Returns the current elapsed time in seconds since the start of the game.
<pre>float game.GetScore()</pre> 	Returns the current game score for Astrobee.
<pre>float game.GetWaterUnits()</pre> 	Returns the Number of Water Units the Astrobee Is carrying
<pre>float game.GetRobotPositionX()</pre> 	Returns a float of the X coordinate of Astrobee
<pre>float game.GetRobotPositionY()</pre> 	Returns a float of the Y coordinate of Astrobee
<pre>float game.GetRobotPositionZ()</pre> 	Returns a float of the Z coordinate of Astrobee
<pre>float game.GetBonusCrop()</pre> 	Returns an int of the Bonus Crop ID only when at Astronaut
<pre>bool game.OpponentAtAstronaut()</pre> 	Returns True if the Opponent is currently in the astronaut Zone.
<pre>bool game.OpponentAtWatering()</pre> 	Returns True if the Opponent is currently in the Watering Zone.

<pre>int game.GetRobotID()</pre> 	Returns 0 if blue robot, returns 1 if red robot
<pre>bool game.OpponentGoingToWatering()</pre> 	Returns True if the Opponent is going to Watering Zone
<pre>bool game.OpponentGoingToAstronaut()</pre> 	Returns True if the Opponent is going to Astronaut Zone
<pre>float game.GetBatteryLevel()</pre> 	Gets Robot Battery Level

Debugging

API Function	Description
<pre>DEBUG("Some Text");</pre> <pre>DEBUG("message, %d", variableName);</pre> 	Prints the supplied text to the console. In the quotations, write your message, followed by a comma, and then if you're returning a float or if you're returning an int. Outside the quotations, type the variable name.