

Lesson 1: Starting a Space Pilot game

Objective: Create a simple environment and an interactive top-down character that a player can control.

 **Time:** 30 Minutes

Description: A basic introduction to Flowlab, and how to add/edit behavior logic.

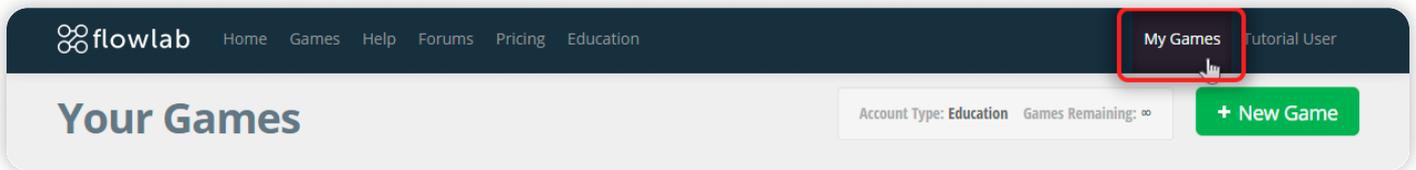
 **Level:** 2 - Beginner

Step 1

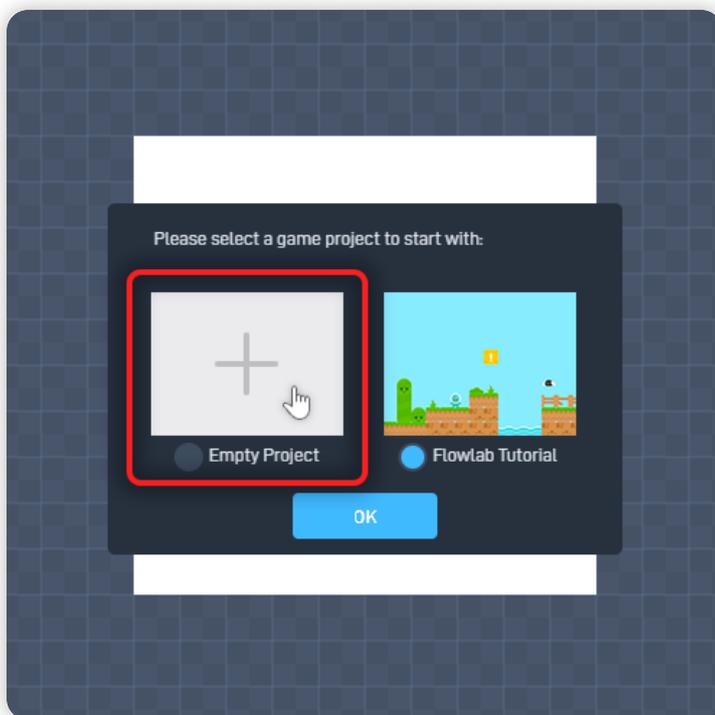
Create a new Game

Log in and start at your "My Games" page <https://flowlab.io/game/list>

Click on "New Game":



You will be prompted to choose an Empty Project, or Begin the tutorial. Choose Empty Project and click "OK".

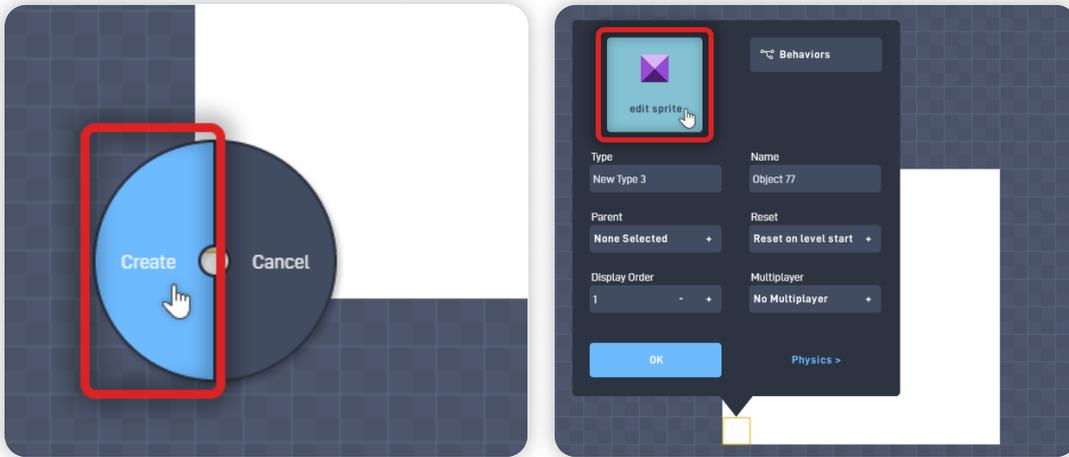


Step 2

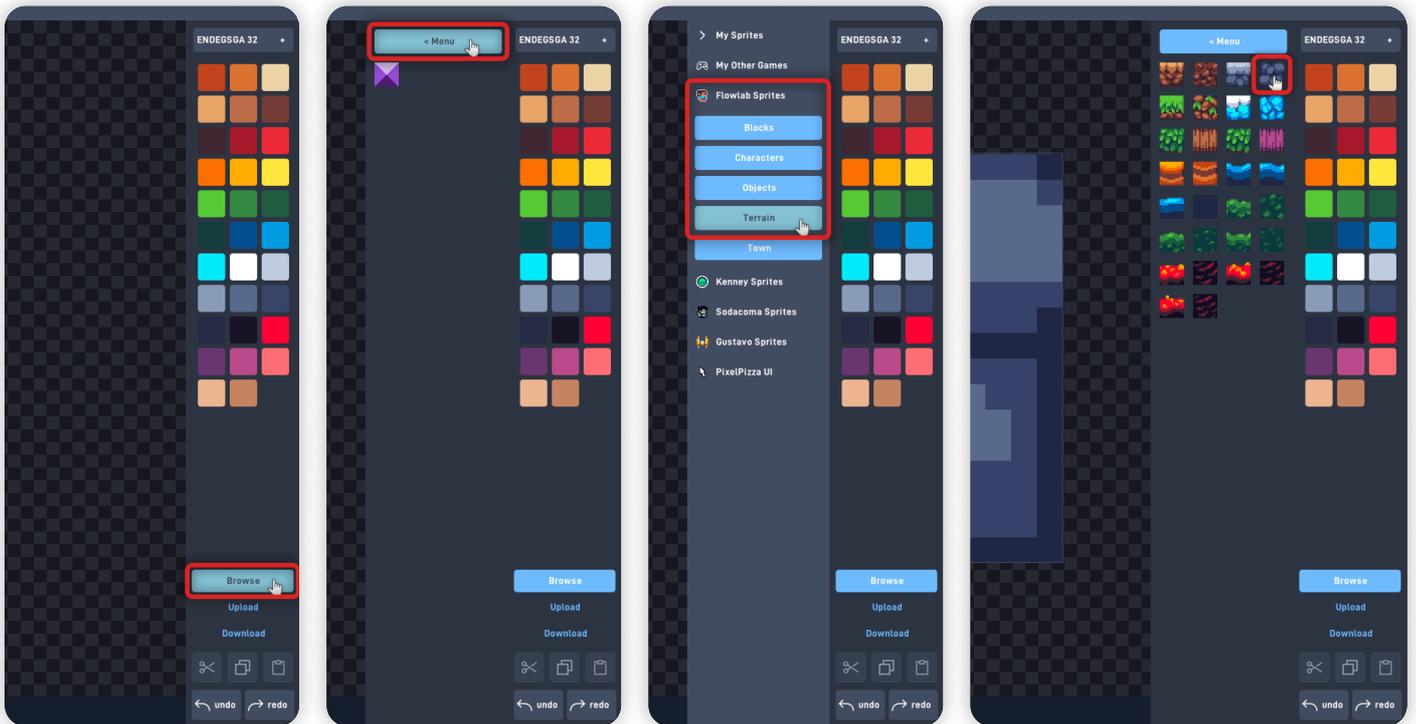
Create a Wall block

In the editor, click anywhere and select "Create" in the menu that appears. This will create a new game object and open its properties panel.

In the properties panel, click the sprite ("edit sprite") to open the Sprite Editor.

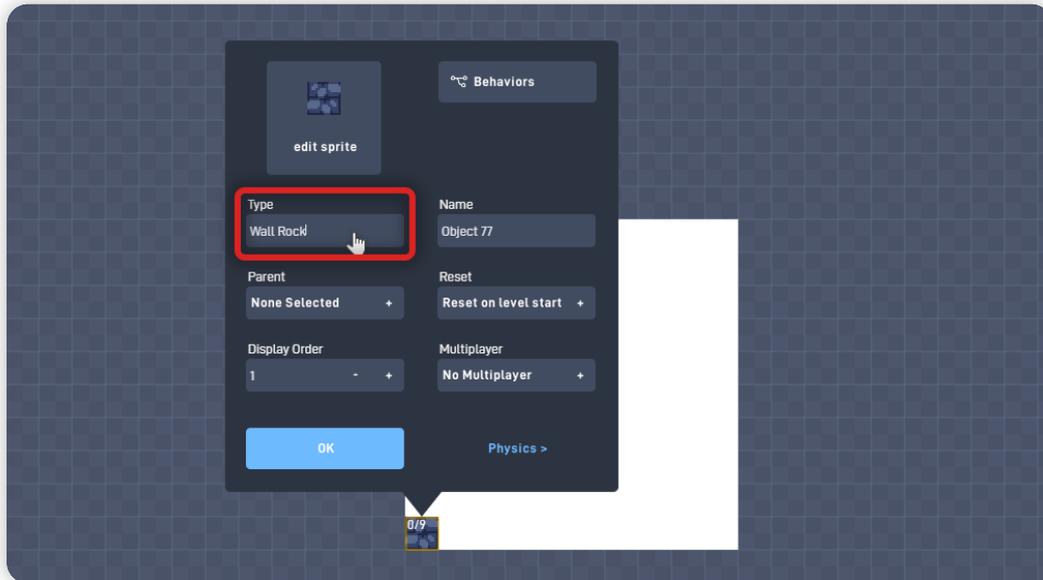


Inside the Sprite Editor, select "Browse", "< Menu", "Flowlab Sprites", "Terrain", and then select a block to use to begin building your game environment:



Click "OK" to close the Sprite Editor and save your changes.

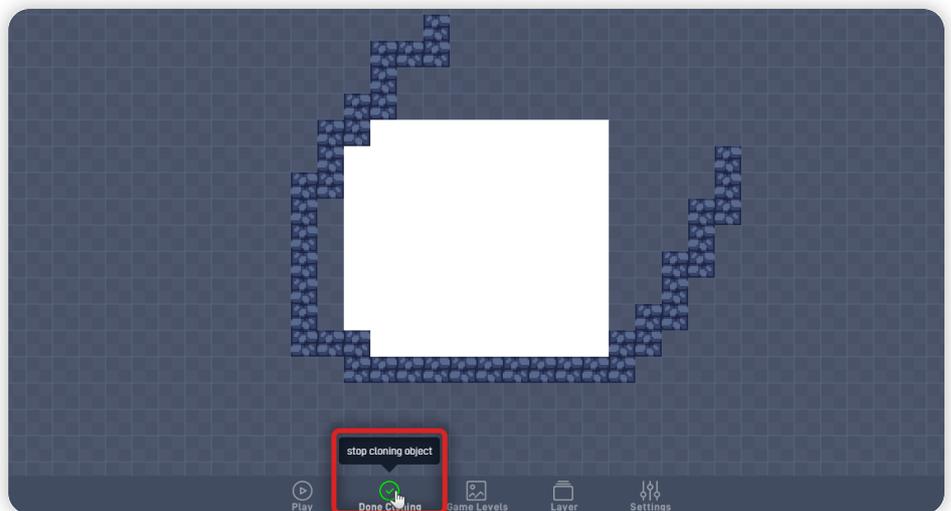
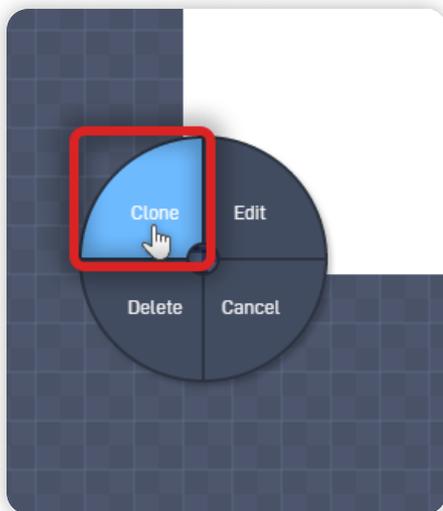
Back on the object properties panel, choose a descriptive name for your new game object. This name can be whatever you like - a good example might be **“Wall Rock”**. Once you finish typing the name click **“OK”** to save your changes.



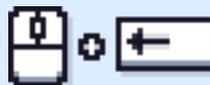
Now clone this object around the game level to start building up the level environment.

Click your object and select **“Clone”**, then click or drag to add copies of this object in the environment. It’s a good idea to start by creating the walls inside the visible level area, but they can be expanded from there.

When you’re done adding blocks, click **“Done Cloning”** in the bottom toolbar:



Don’t worry about making mistakes. Once you finish cloning, any extra objects can be deleted quickly by mouse hovering on them and pressing the **“Backspace”** key.

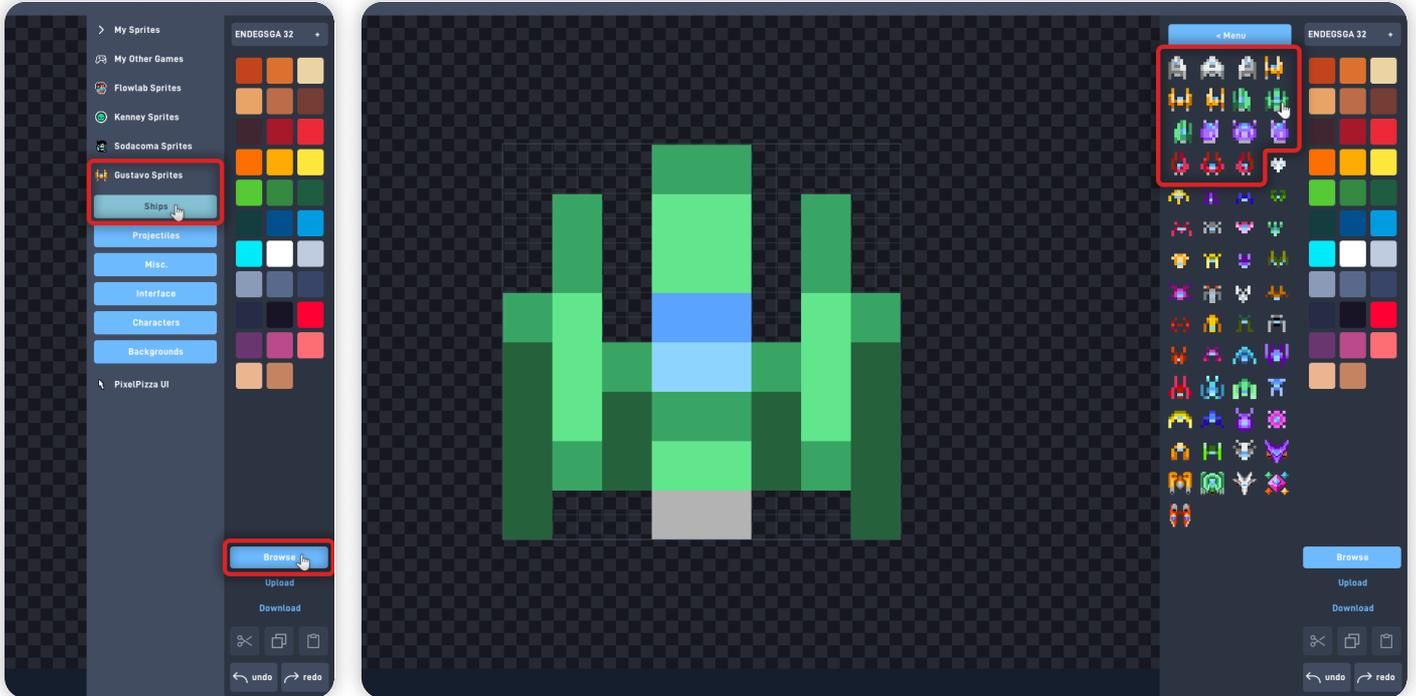


Step 3

Create a Controllable Character

Next, create a new object for your controllable character. As before, just click the editor and select "Create" in the menu that pops up.

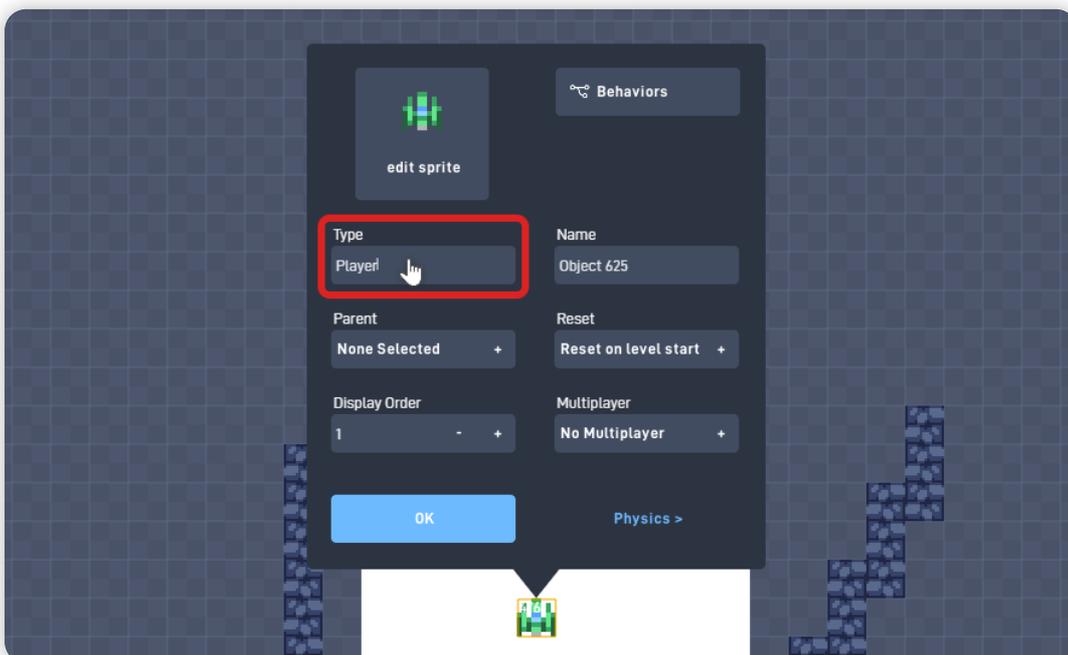
Open the sprite editor and click "Browse", then click "< Menu". From there, select "Gustavo Sprites", and inside it, the "Ships" section:



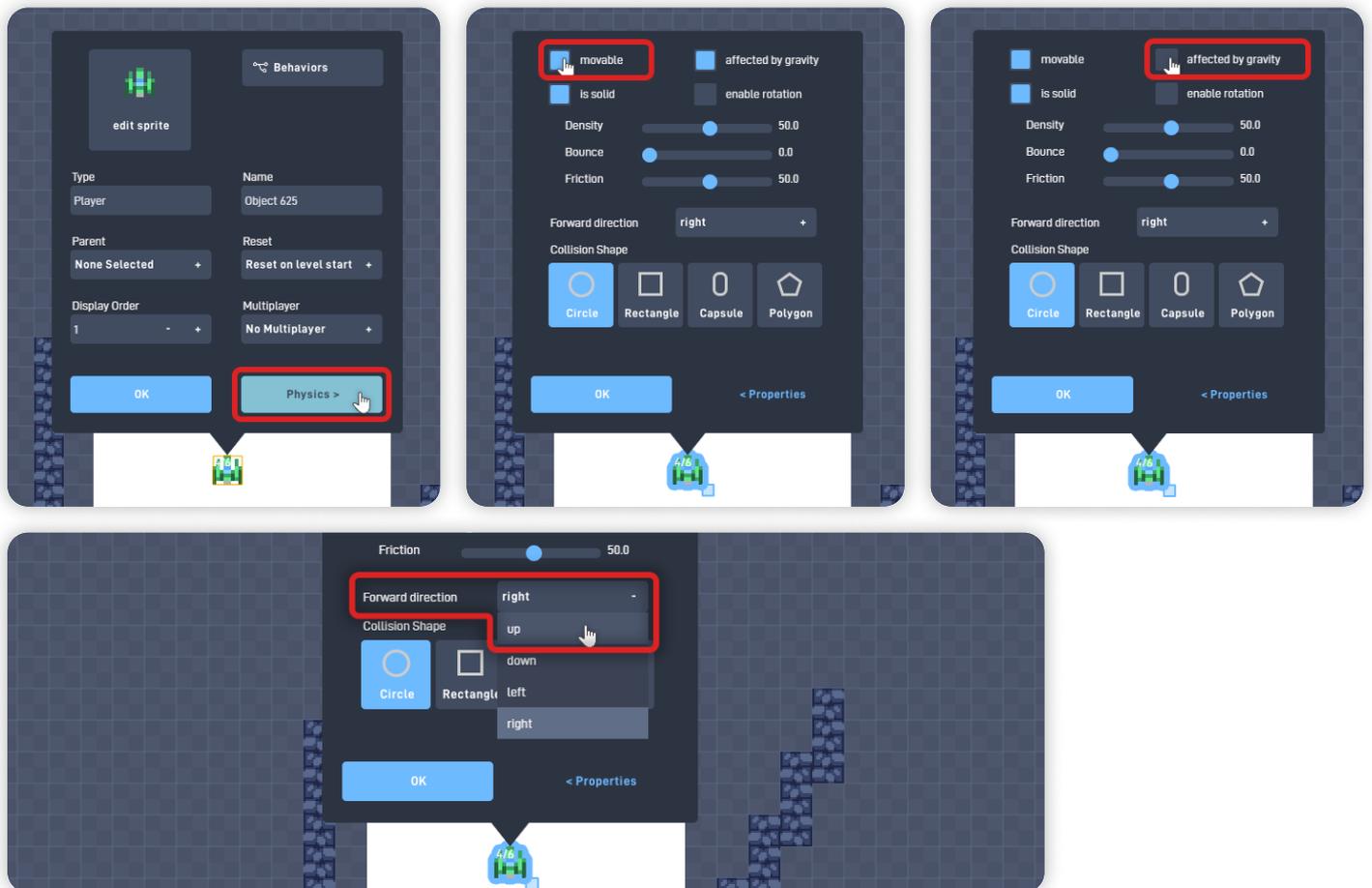
Inside the "Ships" section, choose one of the spaceship sprites (any sprite that is facing upwards is fine), then click "OK" to save.

Giving your objects meaningful names is essential so we can refer to them later.

Let's call this object type "**Player**":

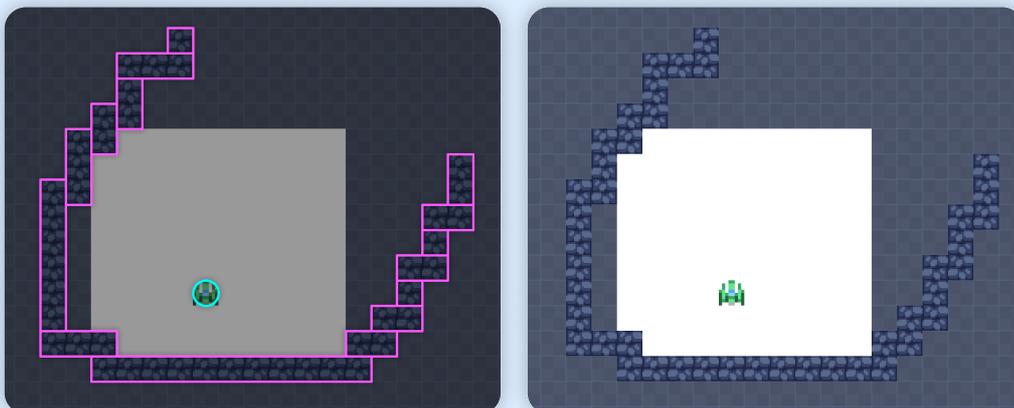


This object should be able to move through your game world freely. So, still, in the object's property panel, go to the "Physics" tab and select the "movable" checkbox, then **unselect the "affected by gravity"** checkbox, and change its "Forward direction" from "right" to "up":



You may notice that upon making this object movable, it automatically defaults its "Collision Shape" from a "Rectangle" to a "Circle". This is because a Circle collision shape is better suited for movable objects as it has less friction across its surface.

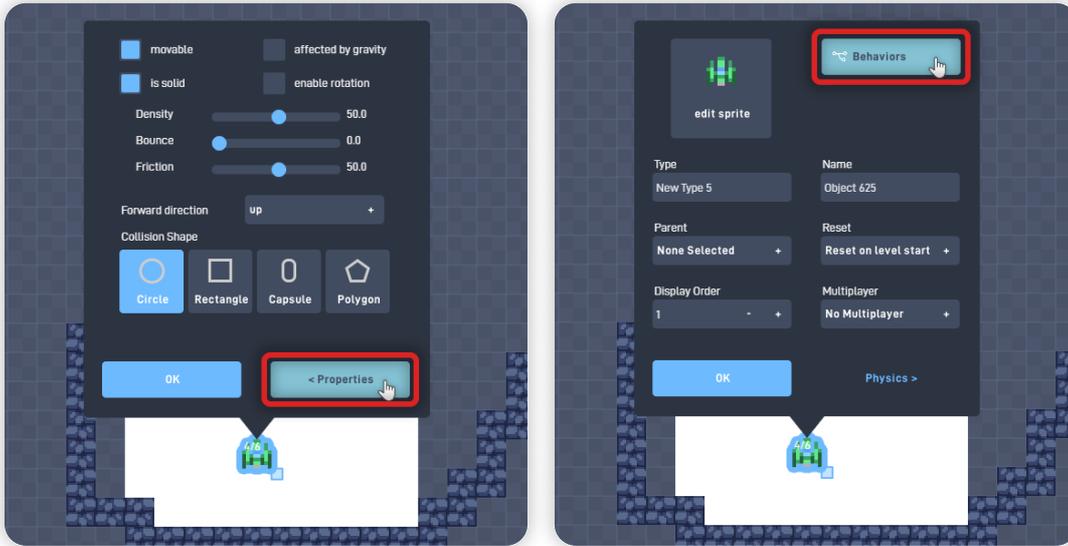
This "Collision Shape" change means that this Object will now be treated as a circle to detect collisions instead of a rectangle. You can see below how the different invisible Collision shapes affect our game in its current state:



Step 4

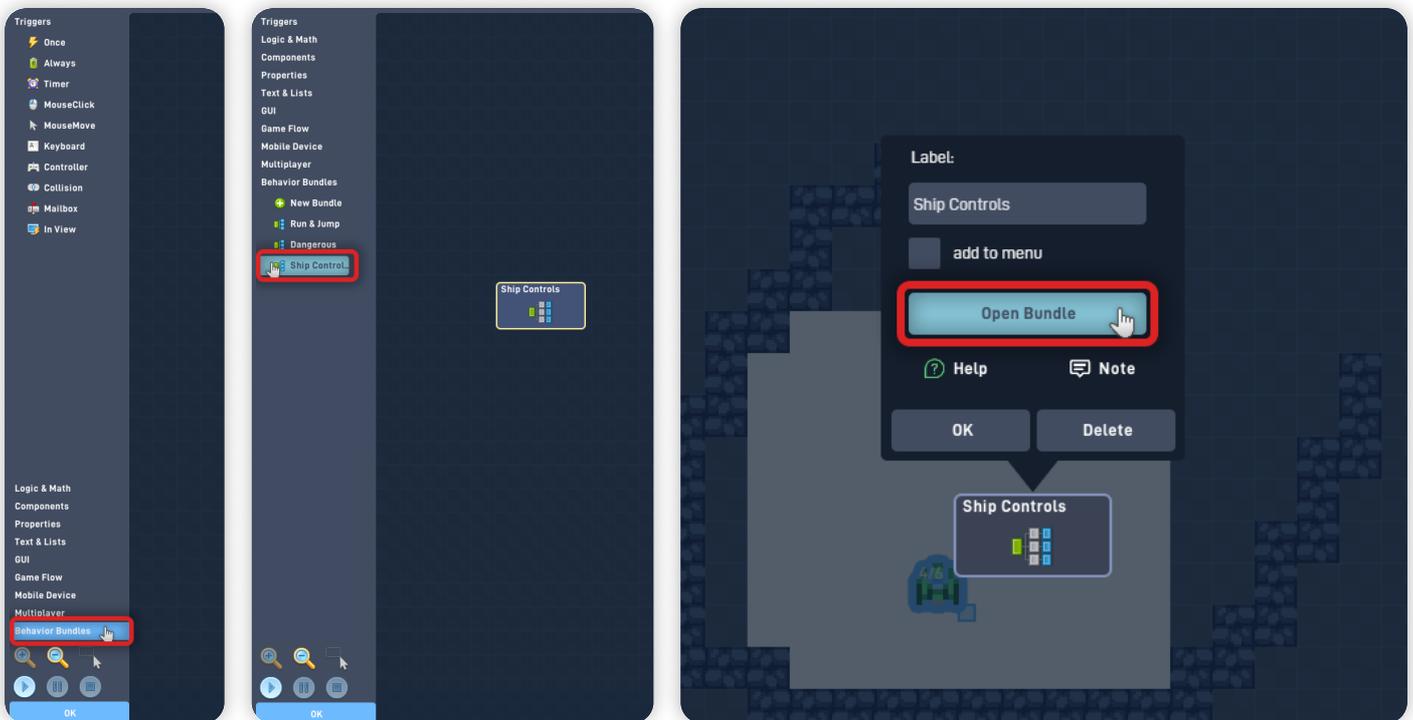
Add Logic to your Character

Next, go back to the “< Properties” tab and click the “Behaviors” button to edit this object’s behaviors:



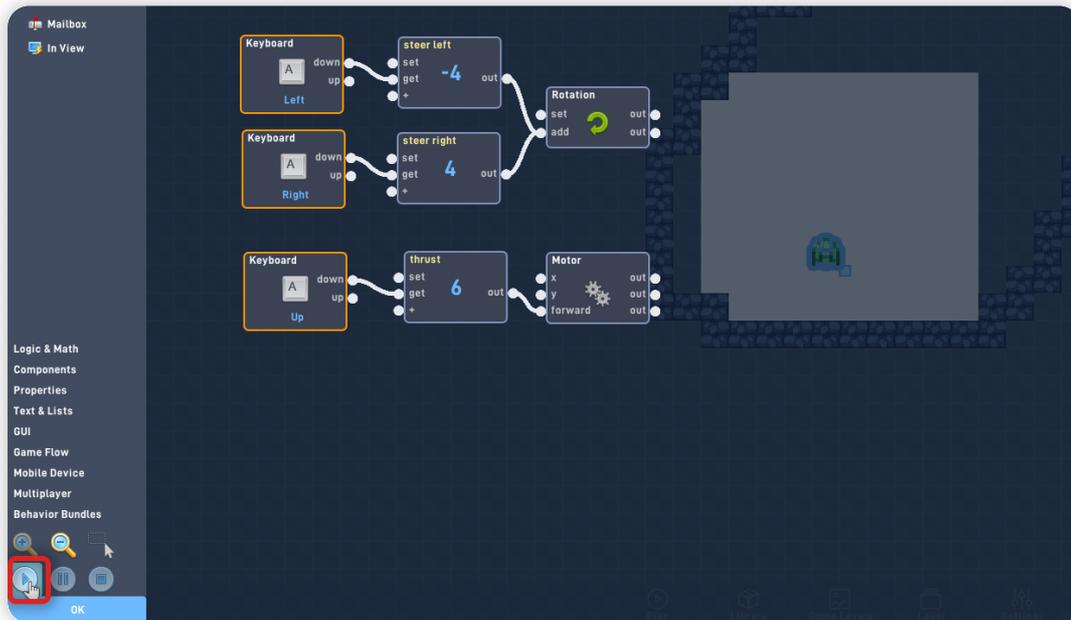
To make the new object movable, we'll add the provided “Ship Controls” bundle, which contains logic blocks that allow this character object to move using the keyboard arrow keys.

To add this bundle, open the “Behavior Bundles” section and click on the “Ship Controls” bundle.



Now, click on the “Ship Controls” bundle you just added, and click on “Open Bundle” to see its contents and logic inside.

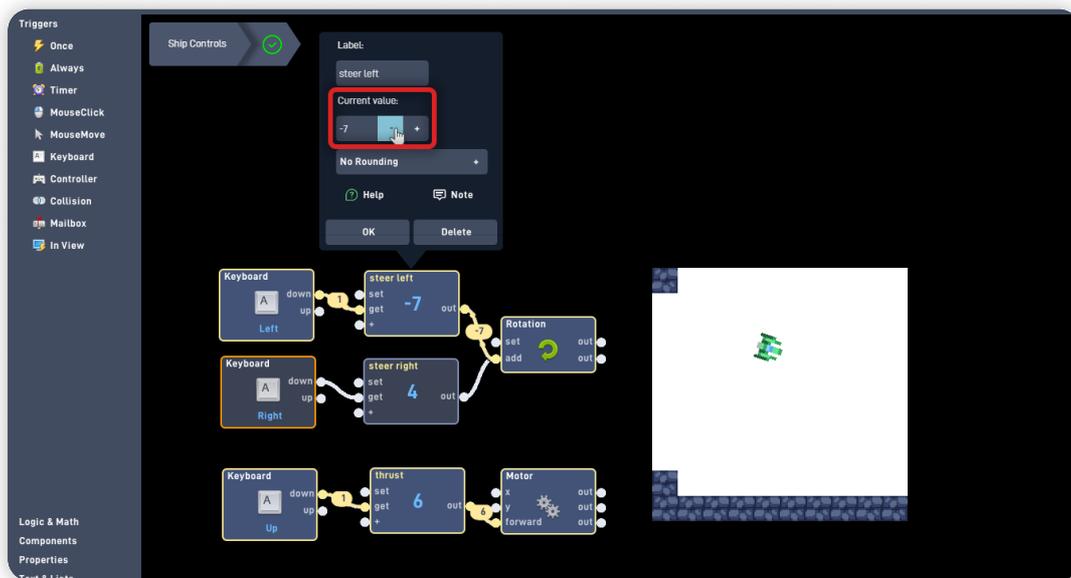
When inside the bundle, you can click on the empty area to move the behaviors out of the view so that you can see the game below. Then, press the “Play” button on the corner to playtest the game inside the editor:



Use the Left, Right, and Up directional keys to control and rotate the ship.

While the game is playing, you can see the Logic trigger and flow. This view is helpful and crucial when testing and troubleshooting your game.

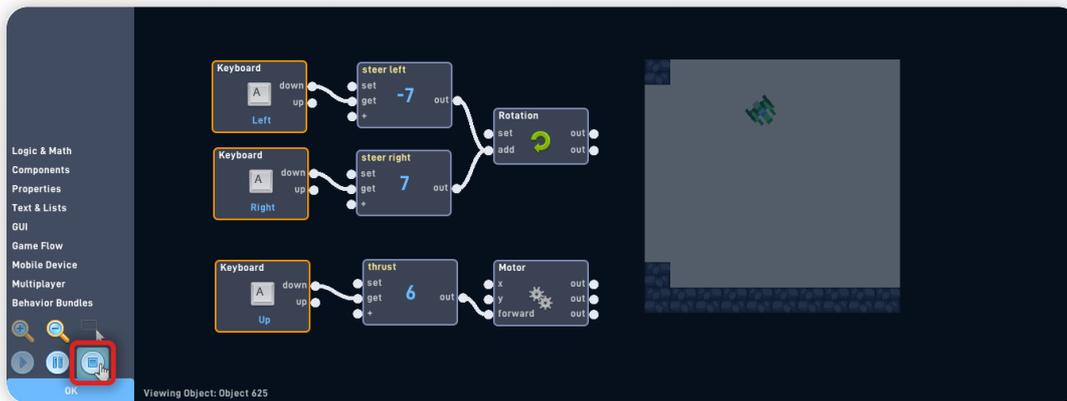
Now, click on the “Steer Left” Number behavior and adjust its “Current value” from “-4” to “-7” or any other negative value that feels right to you. Changing this value to a lower number will make the Player ship turn left faster, but changing the value closer to “0” will make it turn slower.



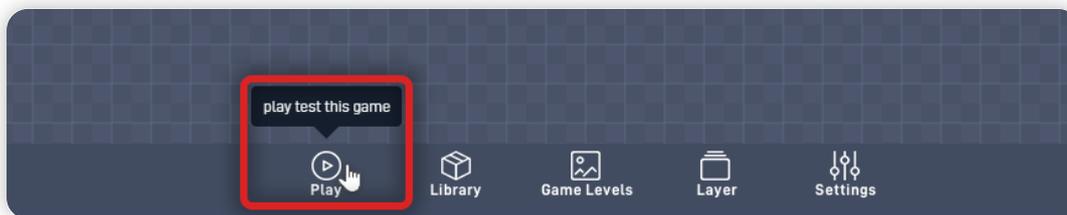
Do the same thing to the “Steer Right” Number behavior, but change its “Current value” from “4” to “7” or the positive value equivalent to the Left value you picked.

This way, the rotation speed is balanced and equal to both sides.

Once you finish editing these values, you can click the “Stop” button to stop the game from running, then press “OK” to exit the editor and save your changes.



To test your game so far, click the “Play” button in the bottom toolbar to switch from “Edit” mode to “Play” mode.



When in Play mode: The Player object should now be movable using the keyboard arrow keys. The left arrow rotates the Ship to the left, the right arrow rotates right, and the up arrow makes the object move forward in the direction it’s currently facing.

If you have problems, check the troubleshooting section.

Troubleshooting

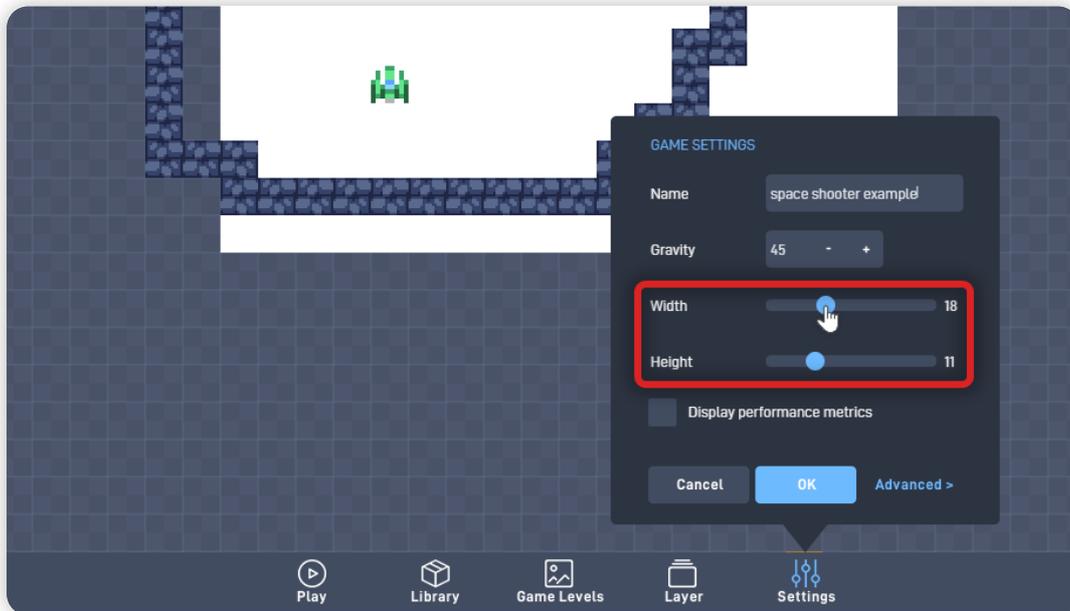
A big part of game development is figuring out why things sometimes don’t behave as you expect. If your game is misbehaving, check the following points:

- **If the Player object doesn’t move or “falls” with the gravity,** ensure that it has “movable” selected and “affected by gravity” unselected on the object Properties panel; *(Step 3)*
- **If the Player ship is not moving in the direction it is facing,** make sure the facing direction of the Ship sprite matches its “Forward direction”. If your Ship sprite is facing up, its “Forward direction” should be set to “up” on the object Properties panel; *(Step 3)*
- **If the Wall object moves when colliding with the player,** ensure that your wall object is not set to “movable”;
- **If the Player Ship can go through the Wall object,** make sure that the Wall object is set to “solid” on the object Properties panel;

Optional Game Enhancements

Once you have a basic working game in place, here are some simple enhancements to try:

- **Make additional environment objects** by repeating Step 2 and selecting different sprites;
- **Adjust the game's viewable size and give it a name** using the game settings panel:



- **Keep your game's viewable size with a 16:9 ratio** to fit horizontal computer screens better. You can keep a coherent ratio by resizing both axes equally, such as:
 - Width: 16 Height: 9
 - Width: 17 Height: 10
 - Width: 18 Height: 11
 - etc.

Space Pilot - Part 1

Nice work!

You've now finished **Lesson 1 out of 6.**

