

# Lesson 1: Starting a Top-Down Adventure Game

**Objective:** Create a simple environment and an interactive character that a player can control, both viewed from a top-down perspective.

 **Time:** 30 Minutes

 **Level:** 2.5 - Beginner/Intermediate

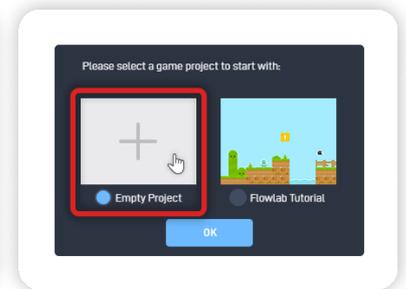
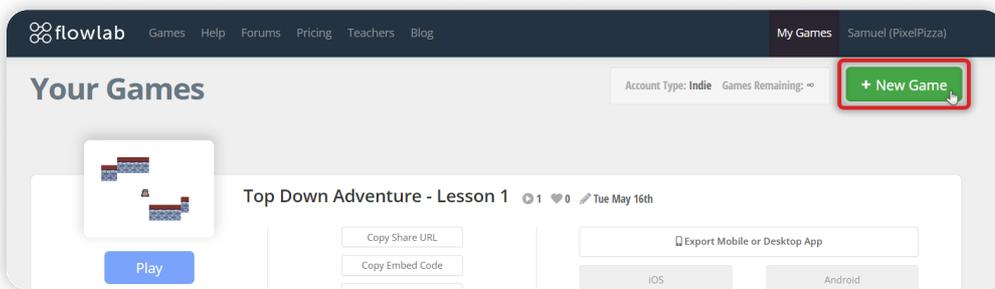
**Description:** A more intermediate introduction to Flowlab and how to create a movement Logic from scratch using Behavior blocks.

## Step 1

### Create a New Game

Login 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

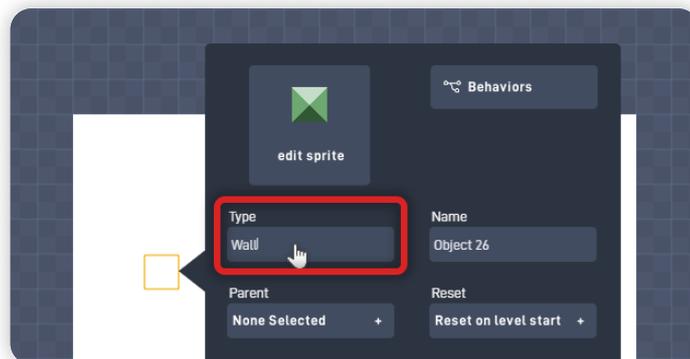
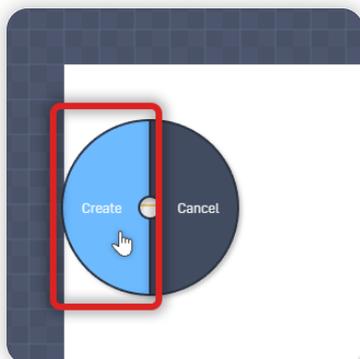
### Create a Wall Block

Inside the Level Editor, click anywhere inside the visible game-area and select "Create" from the circle menu that appears.

*This action will create a new game Object and open its properties panel.*

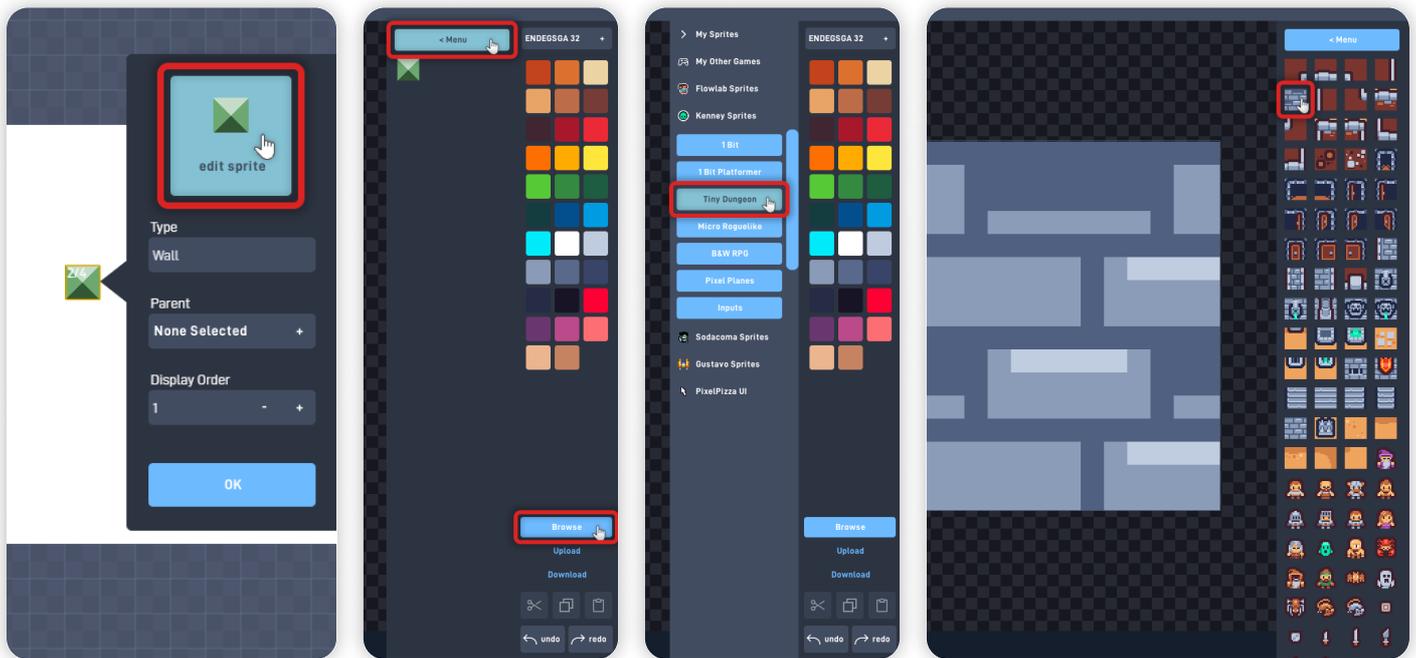
In the properties panel, choose and type in a descriptive name for the new game Object by writing in the "Type" area.

This name can be whatever you like - a good example is "Wall".



Click on the Sprite (it says “edit sprite”) to open the Sprite Editor.

Inside the Sprite Editor, select “Browse”, then “< Menu”, then “Kenney Sprites”, then “Tiny Dungeon”. From here, choose a “front-facing” block Sprite 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, click “OK” again to close the object properties panel.

### Step 3

#### Create the Upper Part of the Wall

By repeating the last step, create a new Object to put on top of the “Wall” object we just made.

We will create an **“Illusion of Perspective”** by combining these two Wall objects. This Illusion is crucial to make our game seem as if the game camera was looking down from above - also known as the “Top Down” or “Bird View” perspective/games.

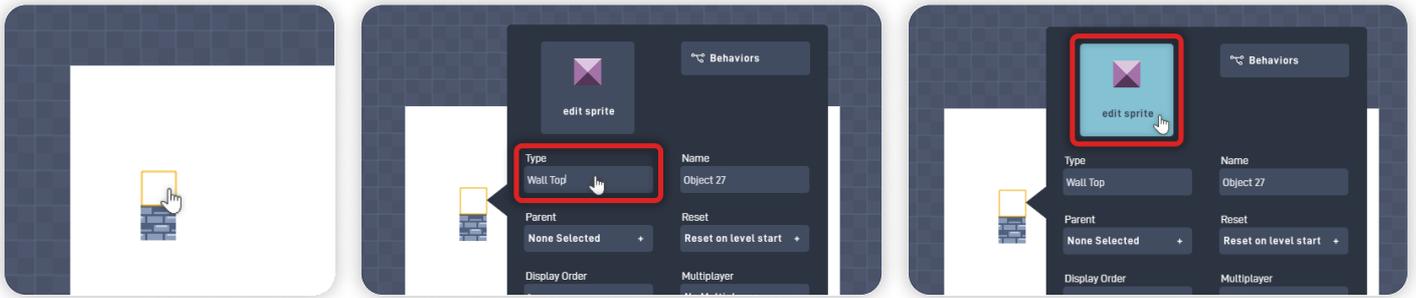


Top Down Game: “Tiny Dungeon” by Kenney

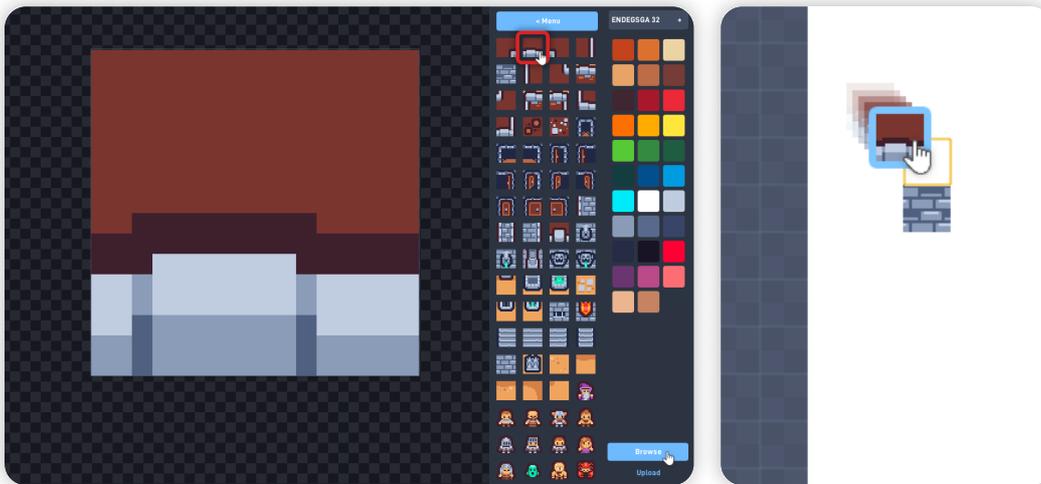


Camera looking down from above

Click on the area above the Wall object we just created, and click “Create” from the circle menu. Set its Type to **“Wall Top”**, and click “Edit Sprite” to open the Sprite Editor.



Inside the Sprite Editor, select “Browse”, then “< Menu”, then “Kenney Sprites”, then “Tiny Dungeon”. From this Collection, select the block Sprite shown below:



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

Click “OK” again to close the object properties panel.

*If the “Wall Top” object is not directly above the stone brick Wall object, you can click and hold to move the object and place it in the desired position.*

## Step 4

### Copy & Paste Objects

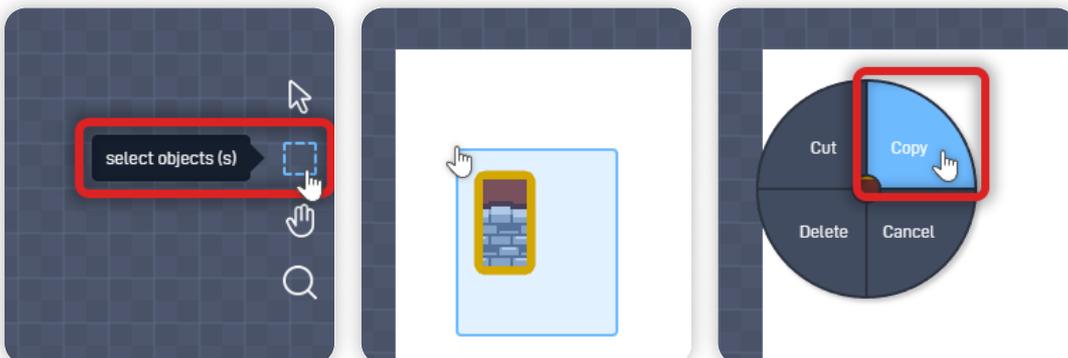
Let’s clone the two Wall objects around the game level to build the level environment.

Select the Rectangle Marquee “Select Object” tool from the tool sidebar.

*Or, you can use the Keyboard Shortcut: “S” to switch to the Select Tool faster.*

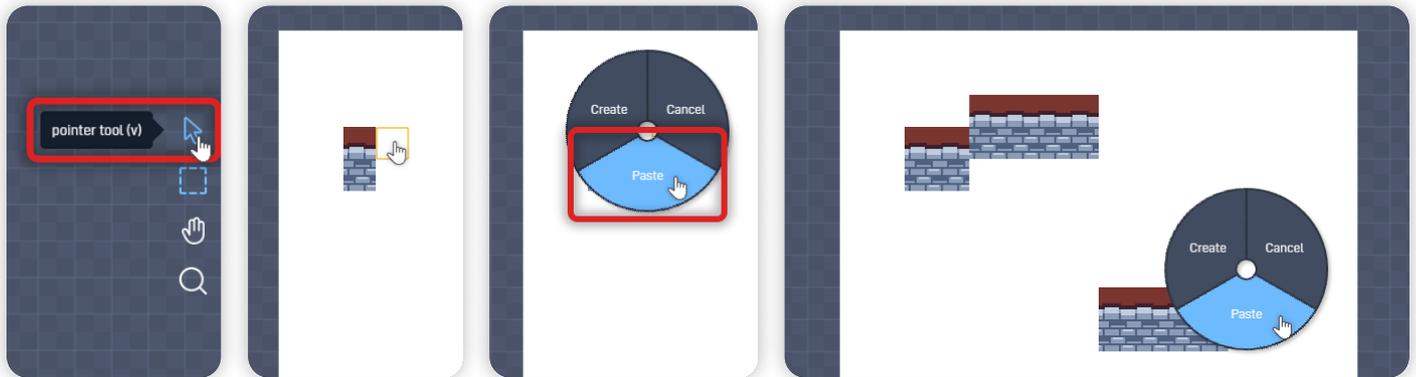
Click and hold on the level empty space, and select the two Wall objects.

Then, click on the selected objects and click “Copy” from the circle menu.



Change to the Pointer tool by clicking the “Mouse” icon from the tool sidebar.  
Or, you can use the Keyboard Shortcut: “V” to switch back to the Pointer Tool faster.

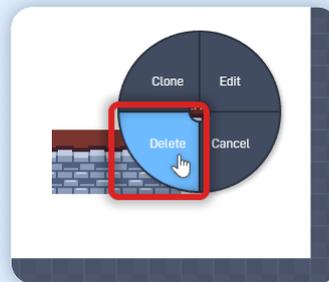
Click right next to the two Wall objects and click “Paste” from the circle menu.  
Repeat this step and “draw” some simple walls for your level, as shown below.



### Don't worry about making mistakes...

Once you finish duplicating, any extra Objects can be deleted quickly by mouse hovering on them and pressing the “Backspace” key.

Or, you can also delete Objects by clicking on the object you want to delete and selecting “Delete” from the circle menu.



## Step 5

### Create a Controllable Character

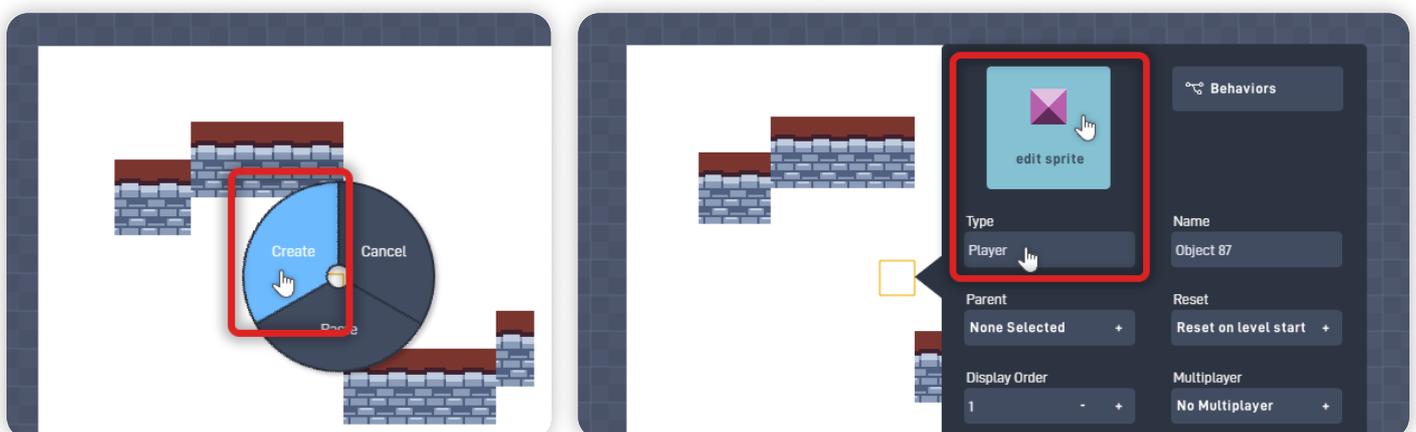
Next, create a new Object for a Character meant to be controlled by the Player.

As before, click inside the visible game area in the editor, and select “Create” from the circle menu.

Let's call this object type “**Player**”.

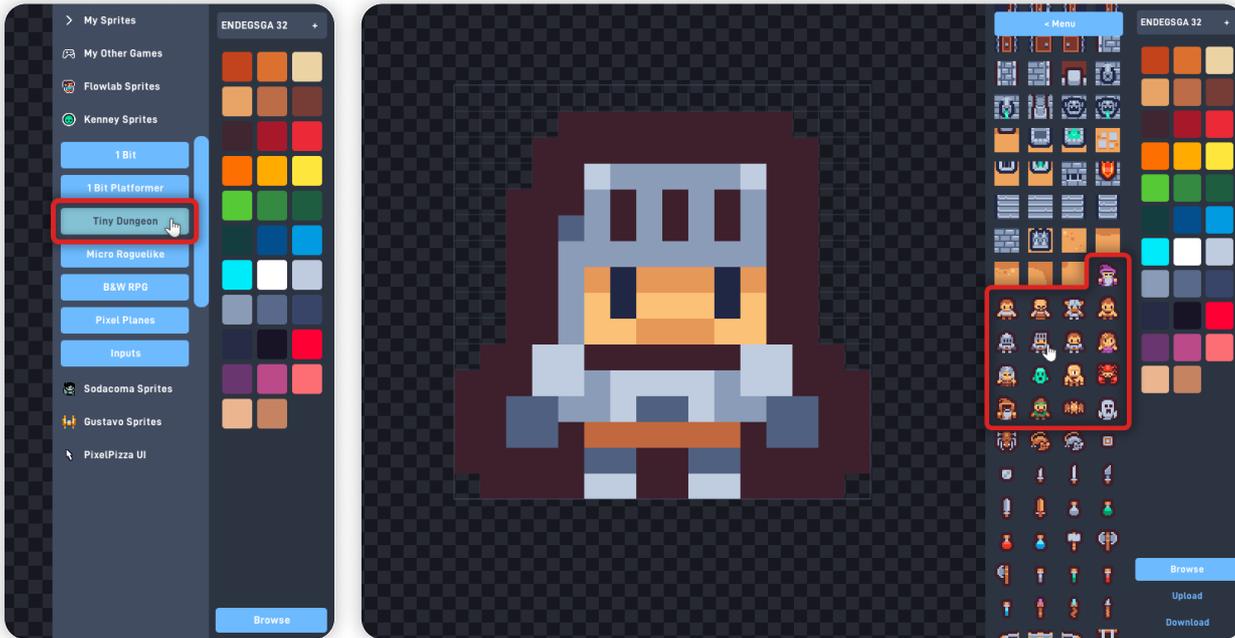
*Giving your objects meaningful names is essential so we can refer to them later.*

Click “Edit Sprite” to open the Sprite Editor.



Inside the Sprite Editor, click “Browse”, then “< Menu”, then “Kenney Sprites”, then “Tiny Dungeon”.

From this section, choose one of the Character Sprites for your Player.

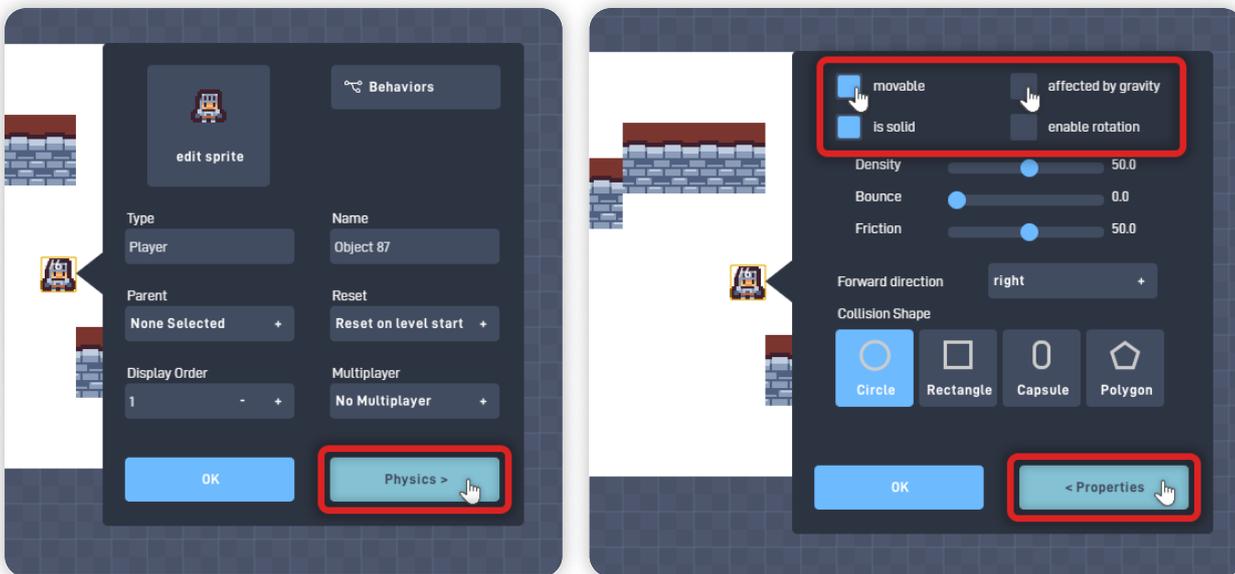


Click “OK” to close the Sprite Editor and save your changes.

This object should be able to move through your game world freely.

So, with the Player Object’s properties panel open, go to the “Physics >” tab.

**Select the “Movable” checkbox, then unselect the “Affected by gravity” checkbox.**



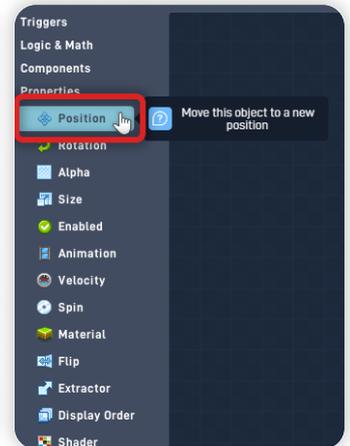
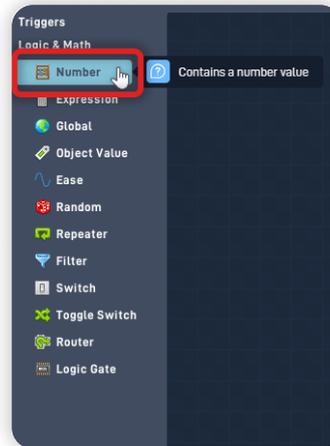
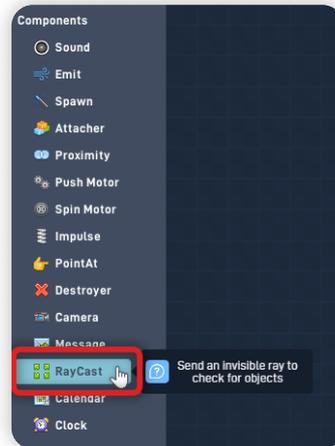
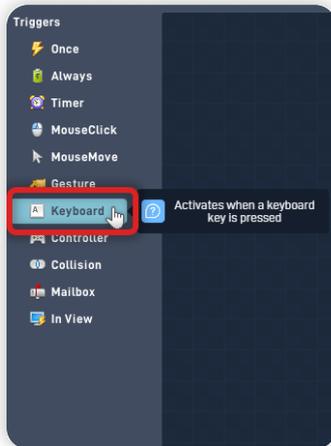
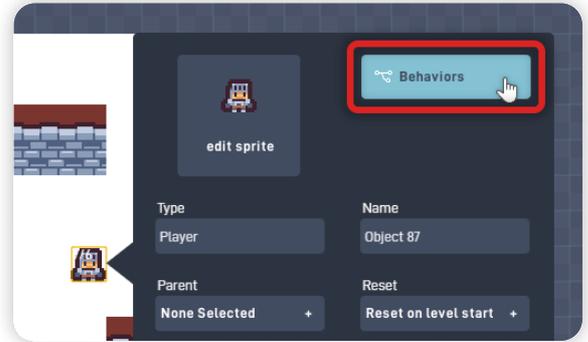
Click “< Properties” to go back to the Properties tab.

## Step 6

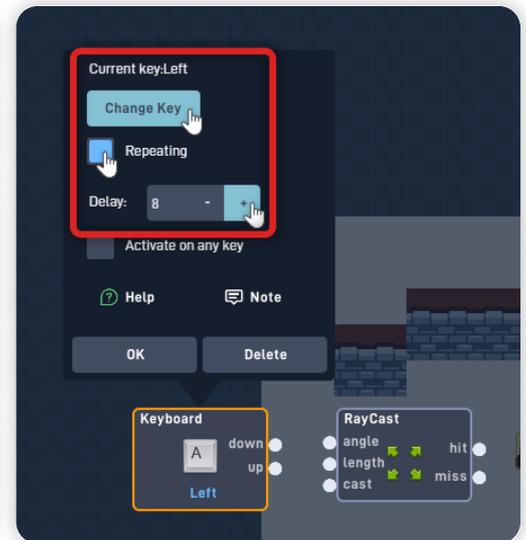
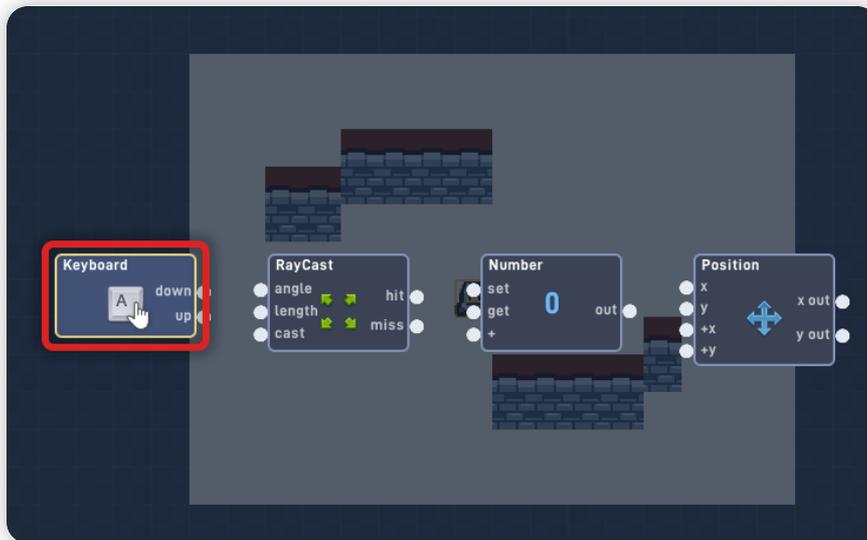
### Add Logic to Your Character

Now, click on the “Behaviors” button to open the Behavior Editor and add logic to the Player Object.

From the Triggers section, add a **Keyboard** behavior.  
From the Components section, add a **RayCast** behavior.  
From the Logic & Math, add a **Number** behavior.  
From the Properties, add a **Position** behavior.



Align the behaviors from left to right, as shown below.  
Click on the “Keyboard” Behavior to open its behavior panel.



Click on “Change Key” to set the Keyboard key, and set it to “Left” by pressing the Left Arrow key.

Select “Repeating” and set the Delay to “8”.

Click “OK” to close the behavior panel.

Click on the RayCast Behavior to open its behavior panel.  
Set the **Ray Angle to "180"**, and ensure the Ray Length is set to "32".  
Click "OK" to close the behavior panel.

*The RayCast behavior sends out an invisible ray from your Object to detect if there is another Object in the set direction.  
The hit or miss outputs trigger depending on whether the ray intersects an object.*

When opening the RayCast behavior panel, the Behavior Editor background will fade out so you can see the **RayCast Visualizer/Preview**.

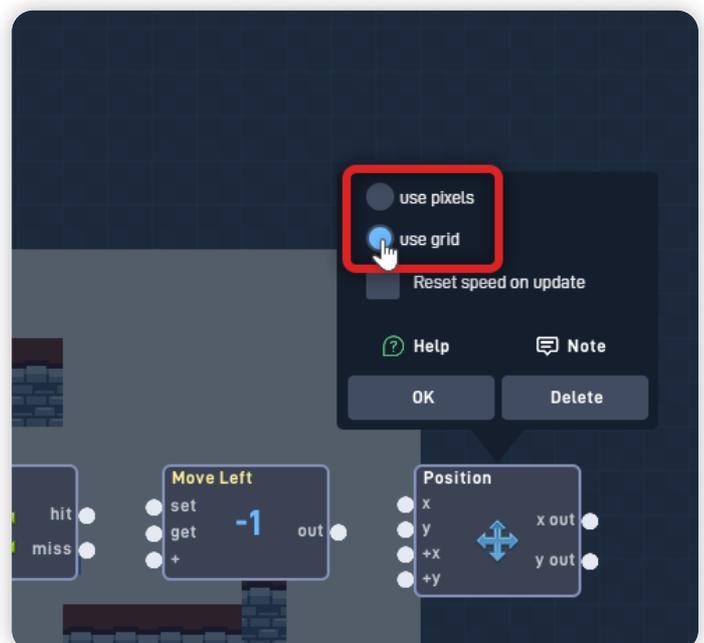
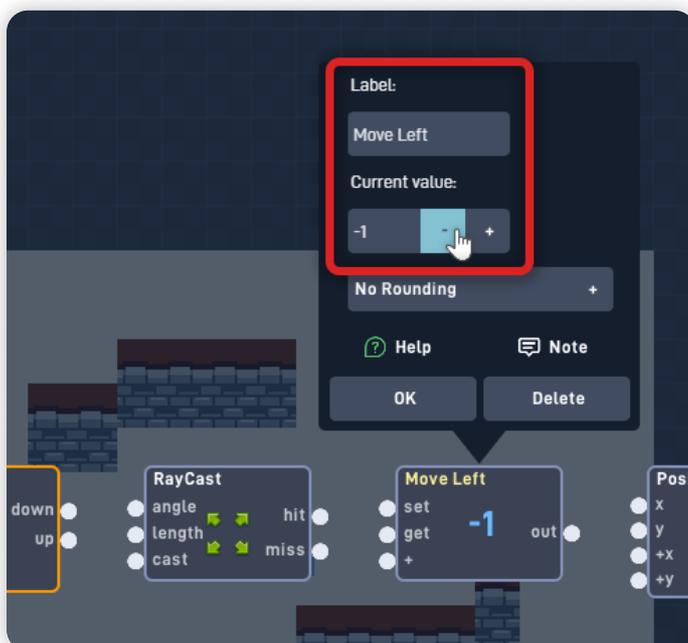
The size and direction of the RayCast are represented by the flashing **"Orange Rectangle"** coming out of the selected object (in this case, the Player object).

For the Logic we are creating, you can use this Preview to ensure the Ray angle matches the direction in which we want the Player to move;  
This Logic will make the Player move Left, so the Ray must be pointing to the Left.



Click on the Number Behavior to open its behavior panel.  
Set its **Label to "Move Left"** and the **Current Value to "-1"**.  
Click "OK" to close the behavior panel.

Click on Position Behavior to open its behavior panel.  
Select **"Use Grid"** and click "OK" to close the behavior panel.



Now, let's connect the Behavior blocks together.

Connect the Keyboard "down" output to the "cast" input from the Raycast behavior.

Connect the Raycast "miss" output to the "get" input from the Number behavior.

Connect the Number "out" to the "+x" input from the Position behavior.



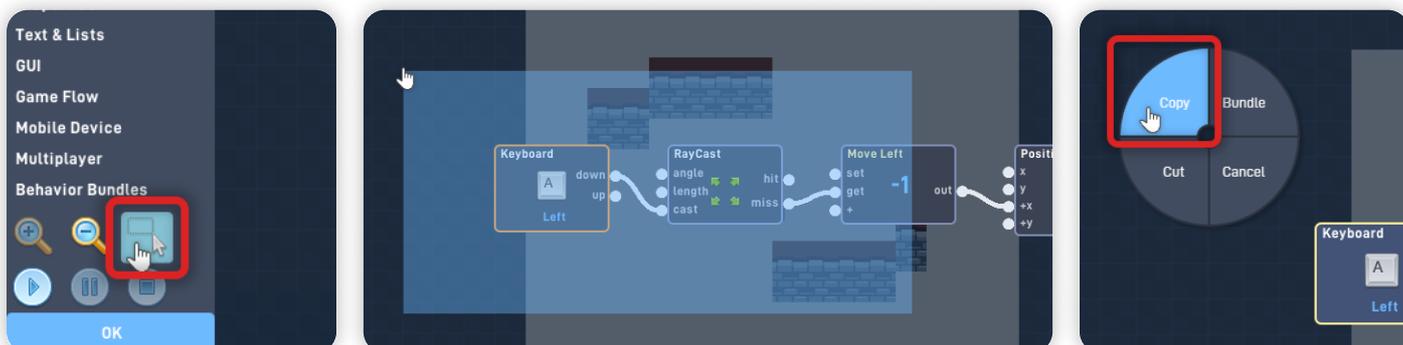
*This batch of connected Behaviors triggers every time the Player presses the "Left" Key, casting a Ray to ensure there isn't a Wall block blocking the way.*

*If the Raycast "misses" (meaning there's nothing in that direction), it will move the Player object "-1" grid block in the X-axis (moves to the left).*

Click on the Mouse "Select Tool" in the bottom left corner to select multiple behaviors simultaneously.

Click and hold to draw the selection area, and **select the Number, the Raycast, and the Keyboard behavior**, as shown below.

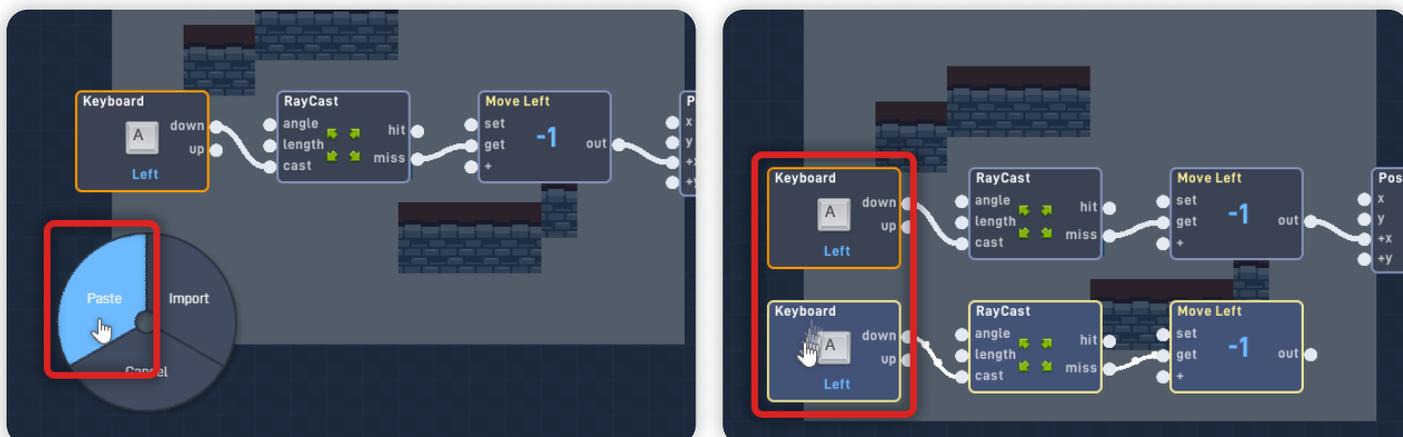
Click "Copy" from the circle menu that pops up.



Click anywhere on the workspace area to open the circle menu.

Click "Paste" to paste the behaviors we just copied.

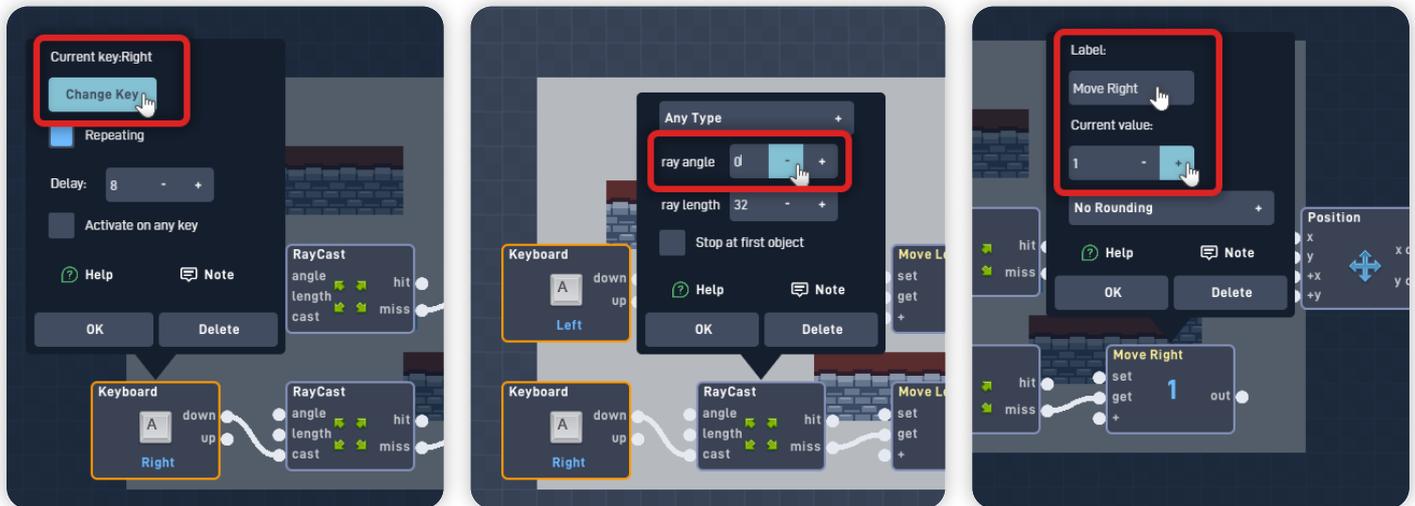
Click and hold on to the Keyboard behavior to move it and place it below the other row of Behaviors, as shown below.



Now, let's adjust the pasted code to move the Player in the opposite direction (Right). Open the new Keyboard behavior panel, set the **key to "Right"** (Right arrow key). Click "OK" to close the behavior panel.

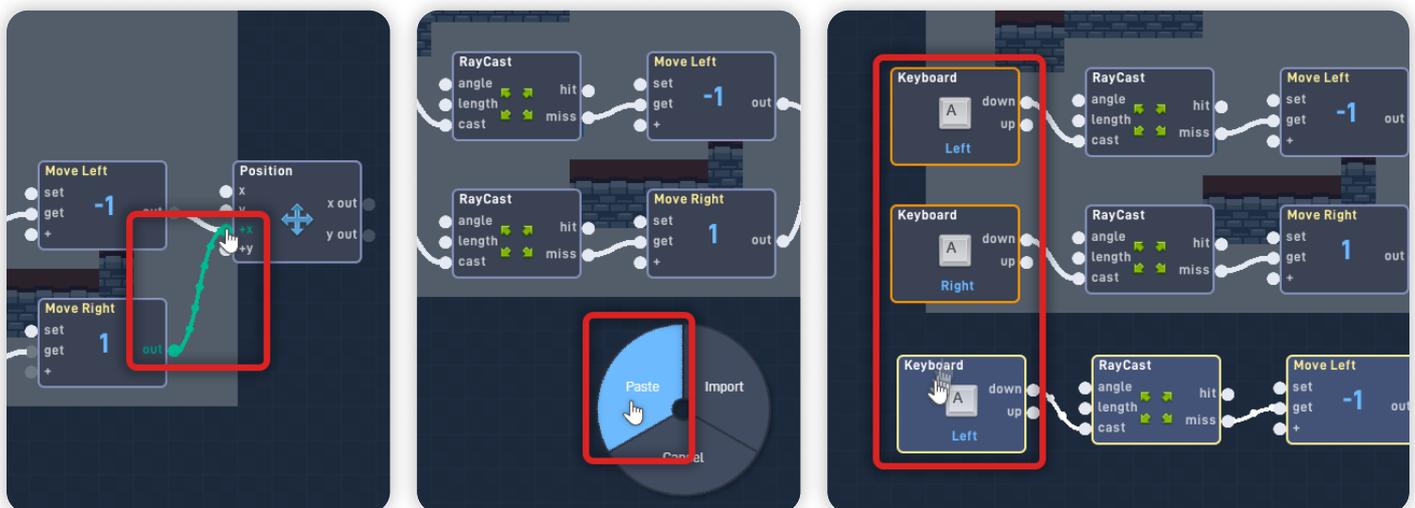
Open the new Raycast behavior panel, set the **Ray Angle to "0"**. Click "OK" to close the behavior panel.

Open the new Number behavior panel and change the **Label to "Move Right"**. Set the **Current Value to "1"**, and click "OK" to close the behavior panel.



Connect the "Move Right" Number "out" to the "+x" input from the Position behavior.

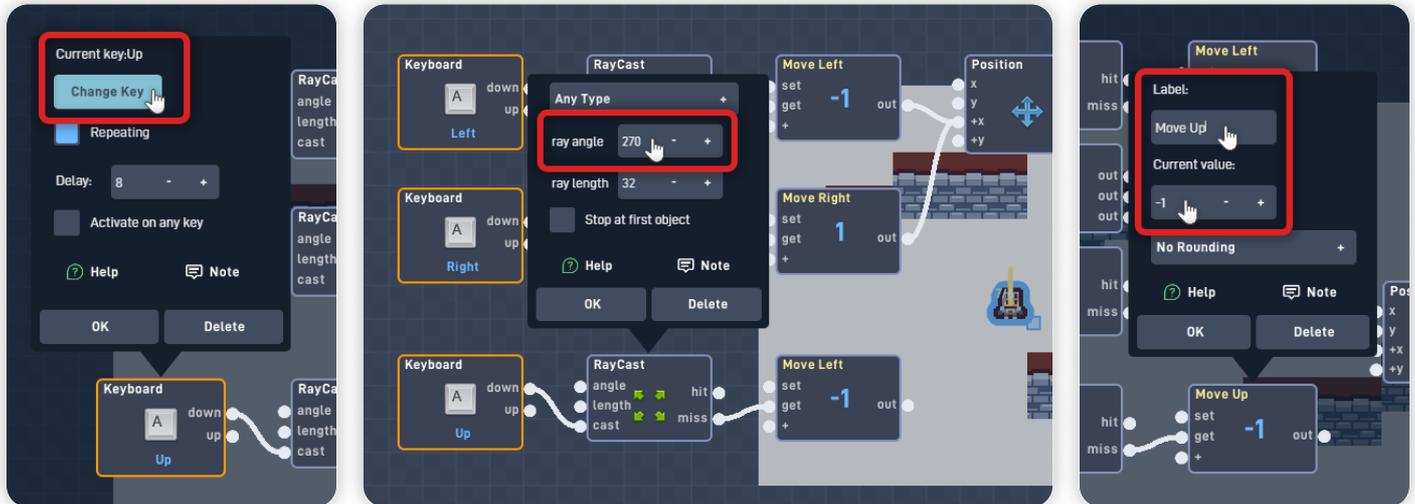
Now let's repeat this process to allow the Player character to move in the other directions: Up and Down. Click anywhere on the workspace area to open the circle menu and click "Paste". Click and hold to move the pasted behaviors below the other rows.



Open the new Keyboard behavior panel, set the **key to "Up"** (Up arrow key), and click "OK" to close the behavior panel.

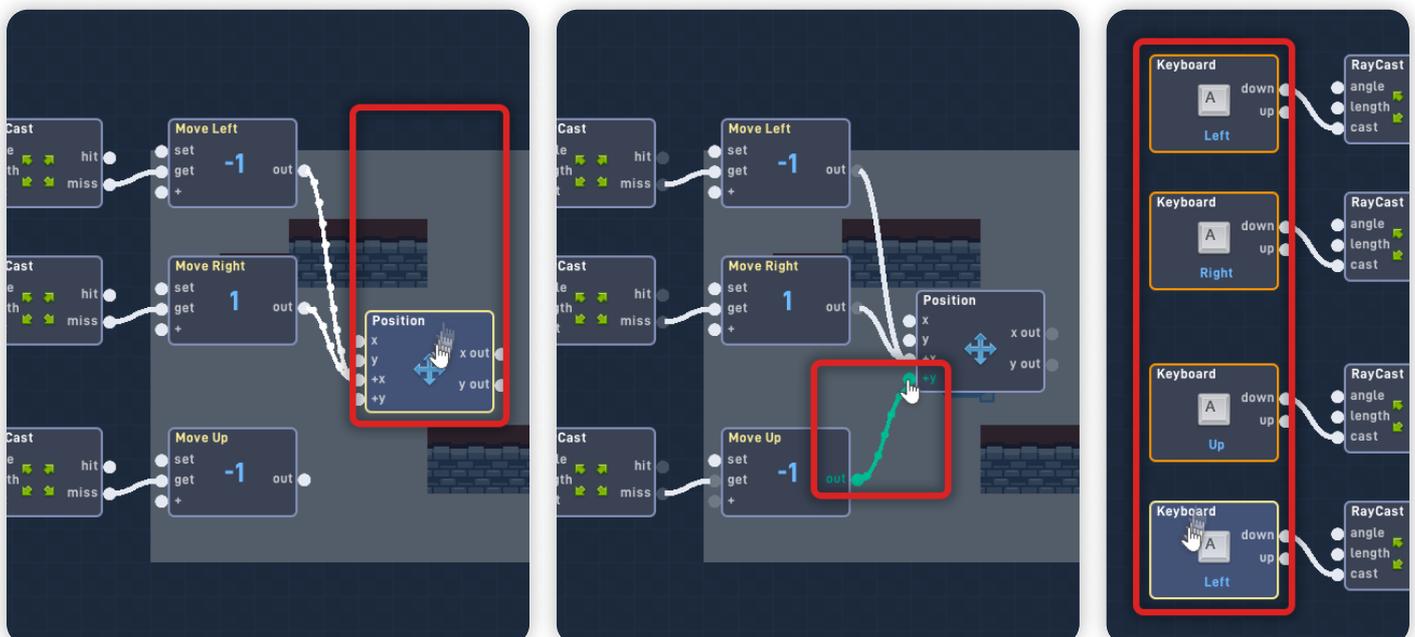
Open the new Raycast behavior panel, set the **Ray angle to "270"**, and click "OK" to close the behavior panel.

Open the new Number behavior panel, change the **Label to "Move Up"**, and ensure the **Current Value is set to "-1"**. Click "OK" to close the behavior panel.



Click and hold onto the Position behavior and position at the center of the logic. Connect the "Move Up" Number "out" to the "+y" input from the Position behavior.

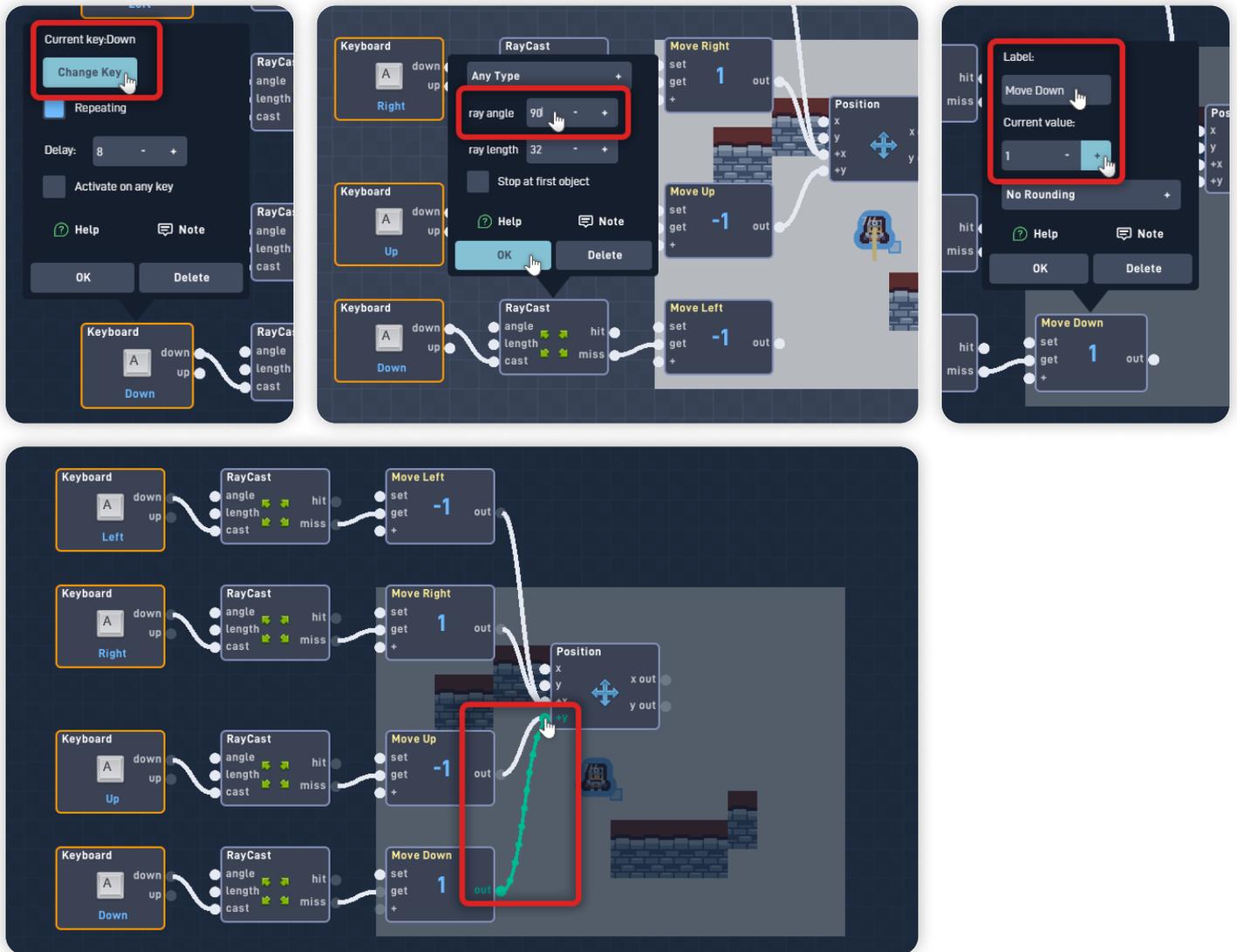
Again, Click anywhere on the workspace area to open the circle menu and click "Paste". Click and hold to move the pasted behaviors below the other rows.



Open the new Keyboard behavior panel, set the **key to "Down"** (Down arrow key), and click "OK" to close the behavior panel.

Open the new Raycast behavior panel, set the **Ray angle to "90"**, and click "OK" to close the behavior panel.

Open the new Number behavior panel, change the **Label to "Move Down"**, set the **Current Value to "1"**, and click "OK" to close the behavior panel.



Connect the Move Down Number "out" to the "+y" input from the Position behavior.

*This Logic and Behaviors will trigger every time the Player presses one of the Directional Arrow Keys, casting a Raycast to ensure there isn't a Wall block blocking the way in that direction.*

*If the Raycast "misses" (meaning there's nothing in that direction), it will move the Player object one grid block in the X-axis (left & right) or the Y-axis (up & down).*

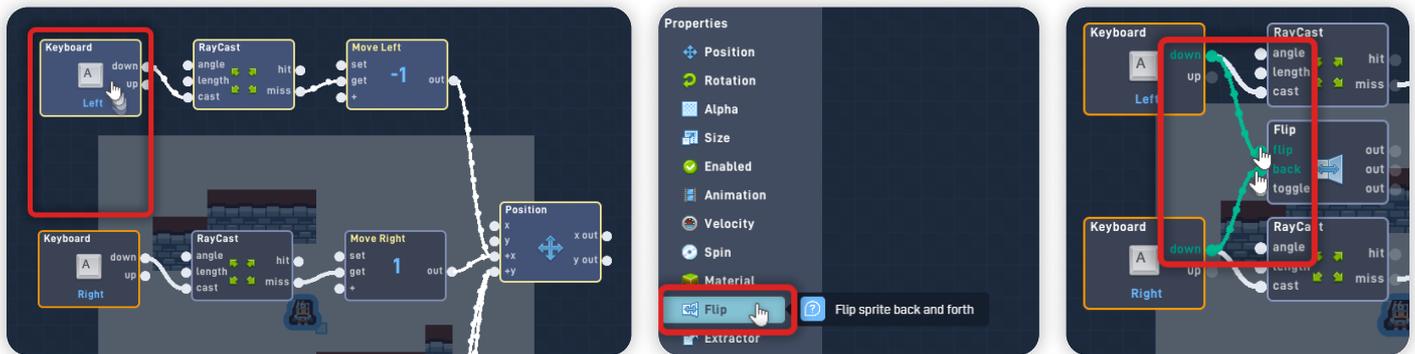
Fantastic work! Now we need to make the character look in the direction they are moving: **We can flip the object Sprite using a Flip behavior.**

Drag the Left Keyboard behavior slightly up to make room for a new block.

From the Properties section, add a **Flip** behavior.

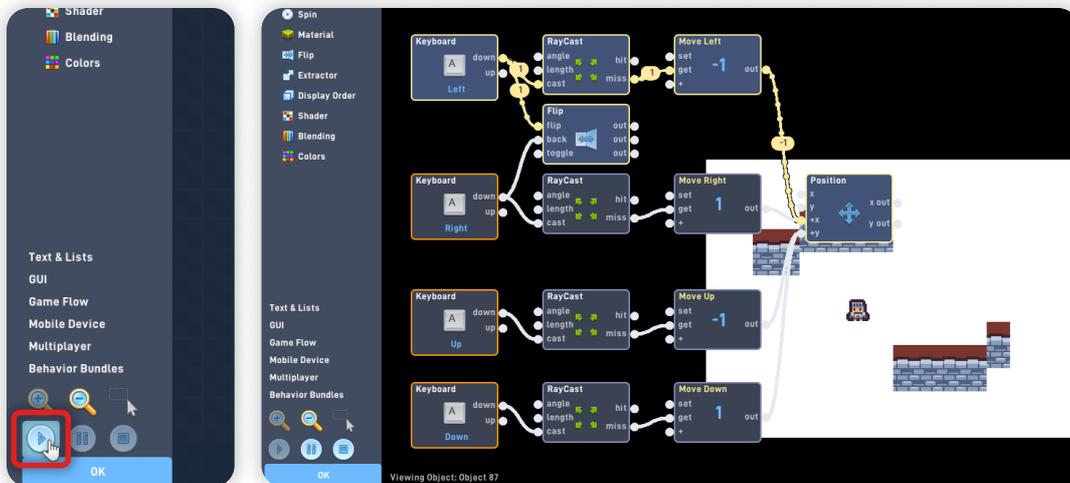
Click and hold on to the Flip behavior and place it near the Left and Right Keyboard behaviors, as shown below.

Connect the Left Keyboard "down" output to the "flip" input from the Flip. Connect the Right Keyboard "down" output to the "back" input from the Flip.



Now, click on the Editor "empty space" to move the behaviors out of view so you can see the game below, and click the "Play" button to playtest inside the editor.

Click the "Stop" button, near the Play button, to stop the playtesting.



Click "OK" to close the Behavior Editor.

Click "OK" to close the Player object properties panel.

Now click "Play" on the bottom toolbar to play your game.

You can open the Level Editor again to adjust the Level if necessary.



### When in Play mode:

- You can move the Player object using the Keyboard Arrow keys. Pressing each Arrow key will make the Player move one grid block in the respective direction;
- The Player won't be able to move in the direction they are pressing if a Wall object is blocking the way, making the player unable to "go through walls";

*If you have problems, check the troubleshooting section.*

## Troubleshooting

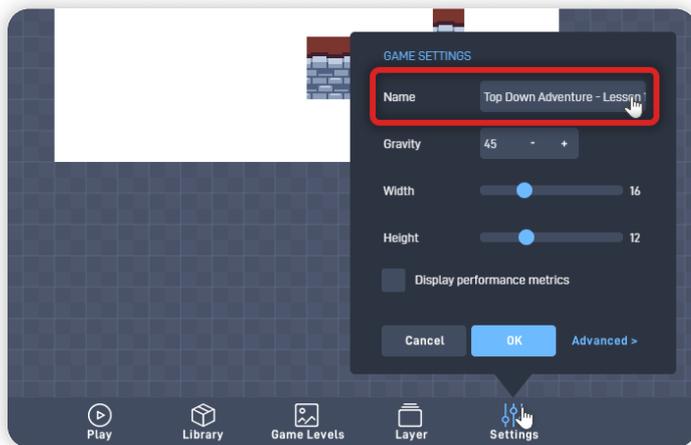
A big part of game development is figuring out why things sometimes do not behave as you expect. If your game is misbehaving, check the following points:

- **If the Player object doesn't move or "falls" with the gravity**, ensure that it has "movable" and "solid" selected and "affected by gravity" is unselected on the Object Properties panel; *(Step 5)*
- **If the Player object doesn't move in the correct direction** when pressing the Arrow keys, make sure the Number behavior values are correct and connected to the right inputs from the Position behavior; *(Step 6)*
- **If the Player moves a single pixel instead of one Grid block at a time**, ensure the Position behavior is set to "Use Grid" on its behavior panel; *(Step 6)*
- **If the Wall object moves when colliding with the player**, ensure that your wall object is not set to "movable" on the object Properties panel;
- **If the Player object 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:

- Set the **Game's Viewable Size and Name** it using the game Settings panel.



---

Top Down Adventure - Part 1

# Nice work!

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

