Lesson 6: A Portal to the Next Level

Objective: Add a Winning condition to the Game by adding a Portal Objective at the end of the Level Course.

Time: 30 Minutes

Level: 2 - Beginner

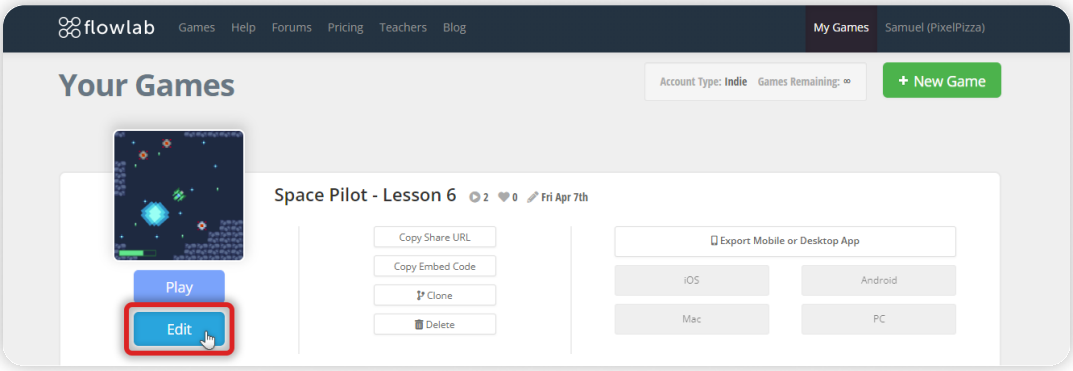
Description: Adding a Portal to the Level that allows the Player to win and complete it. Using custom Logic, make the portal move, show a Victory Text, and load the next Level.

Step 1

Edit your Game

Log in and start at your “My Games” page https://flowlab.io/game/list

Then, click “Edit” next to your game to open the game editor.

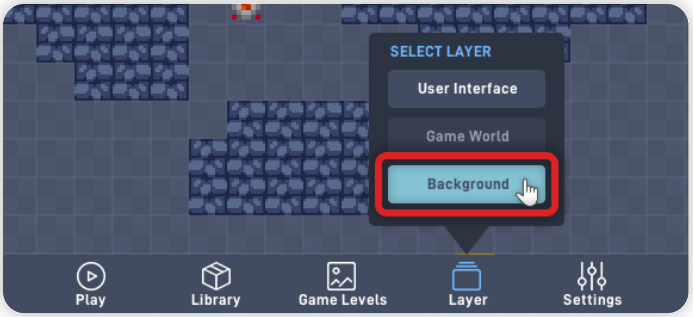


Step 2

Copy the Background Stars

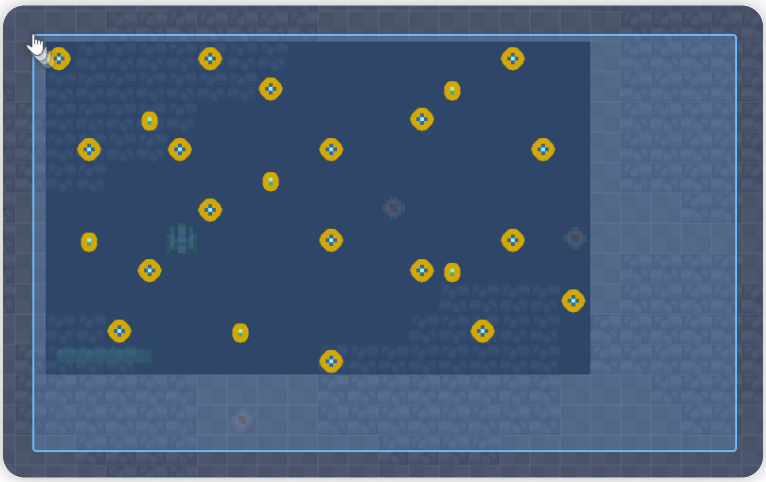
Click “Layer” on the bottom toolbar to open the Layers panel.
Change to the “Background” layer.

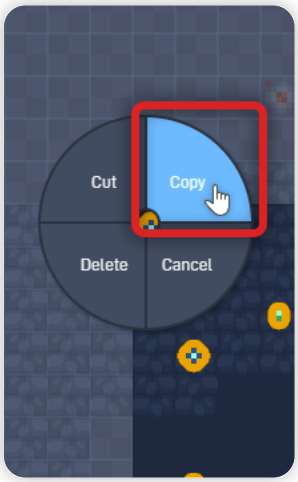
Click on the Rectangle Marquee “Select Objects” tools on the right side toolbar to select multiple objects.

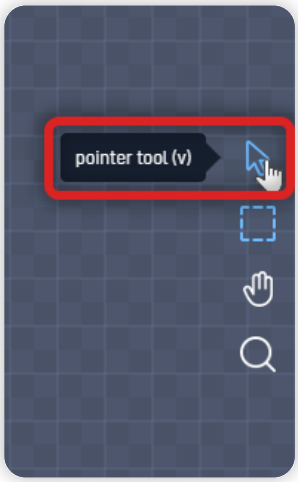


Click and hold near one of the game view corners and drag to select all the Star background objects.
Click on one of the selected stars, and select “Copy” from the circle menu.

Change back to the Pointer tool by clicking the “Mouse Pointer” tool on the side toolbar.





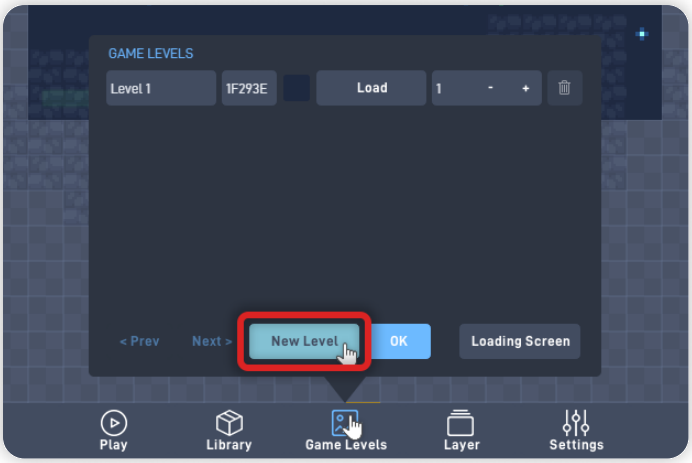


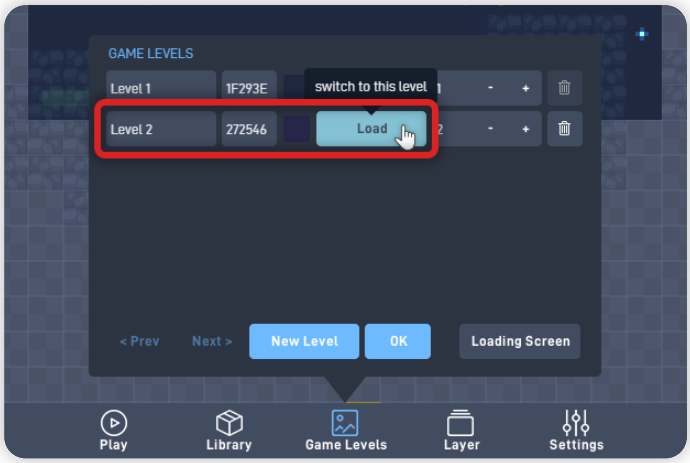
Step 3

Create a New Level
Click “Game Levels” on the bottom toolbar to open the Levels panel.
Click on “New Level”.

Now, change the new level name to “Level 2” and set a Background Color.
We are using this Background Color: 272546 (Dark Purple)

On the Level 2 row, click on its respective “Load” button to load and edit the newly created level.





Select the Color for the Background keeping the space theme in mind. This Background Color can be the same color as the first level, or a slightly different dark color.

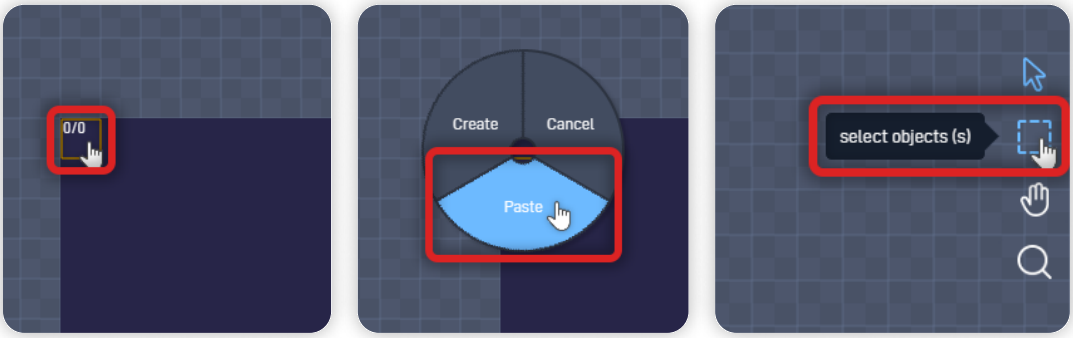
Different colors can help your Player **differentiate the two levels.**

Step 4

Paste the Background Stars and Add the Player Object

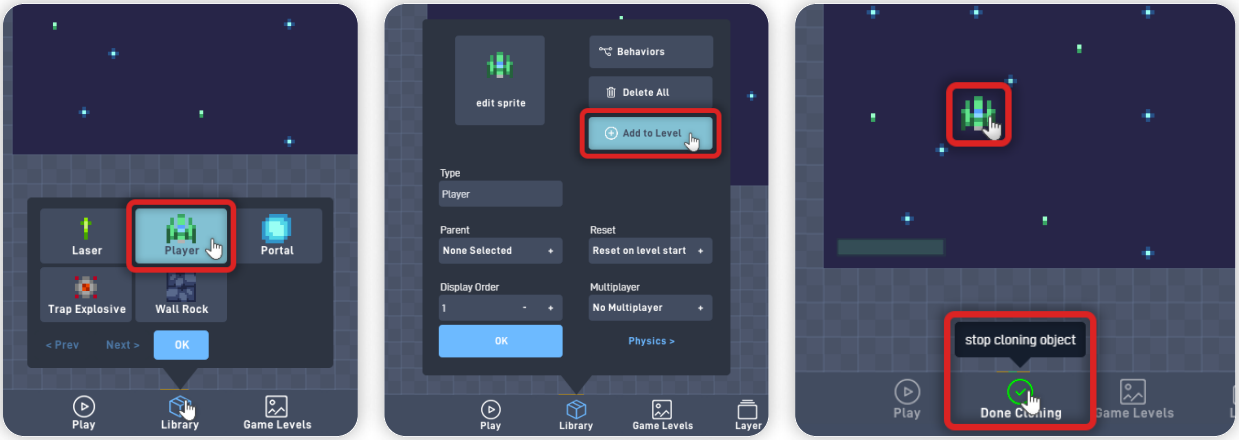
On the empty Level 2, click in the top left corner (0, 0) and click “Paste” on the circle menu to paste the background Stars objects we copied in Step 2.

If the Stars aren't in the same position as the previous level, you can align the objects by selecting and moving the stars on the background layer using the “Select objects” tool.

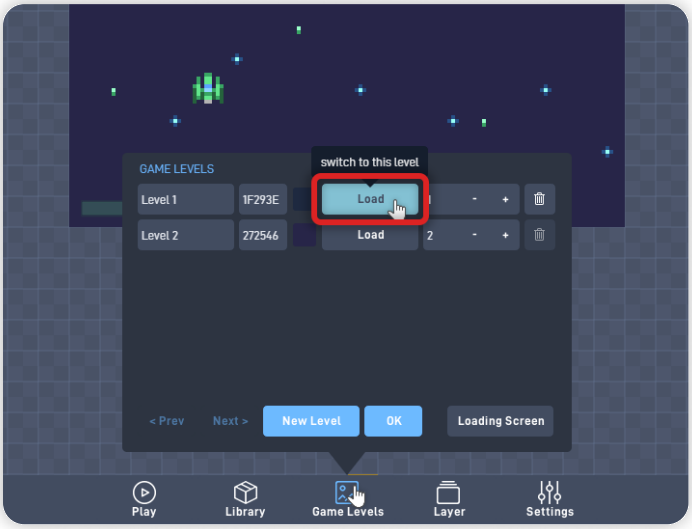


Click on “Library” on the bottom toolbar, select the “Player” object, and click “Add to Level”.

Click anywhere inside the visible game area to add the Player spaceship object to the level. Click “Done Cloning” to stop adding the object.



Click “Game Levels” on the bottom toolbar, and on the Level 1 row, click “Load” to return to the first level.



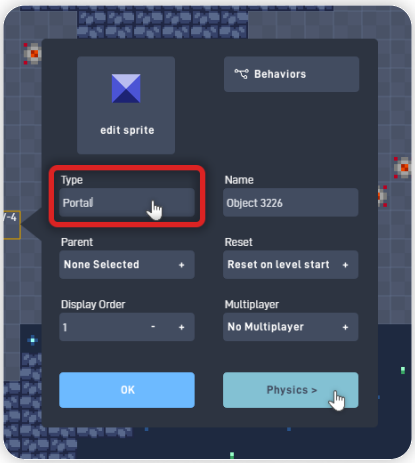
Step 5

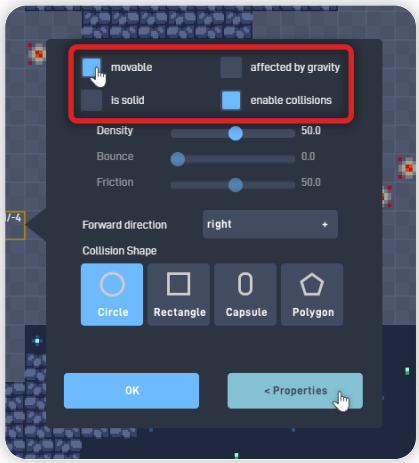
Create the Game's Objective Portal

> In previous lessons, we added a Lose Condition:
> - Restart the level if the Player Health reaches 0.
>
> Now, we need to **add a Win Condition so Players can have an objective:** A Portal that proceeds to the next level.

Click and hold to **move the editor view to the end of your Level Course.**
Click where you would like your Level Goal/Portal to be.
Then, click "Create" from the circle menu.
Don't worry about the position - you can move the object later if necessary.

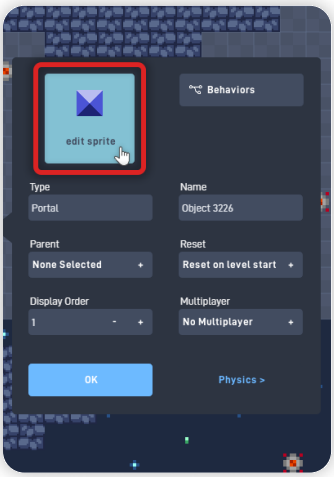
Change its Type to "Portal", and click on "Physics >" to go to the physics tab.

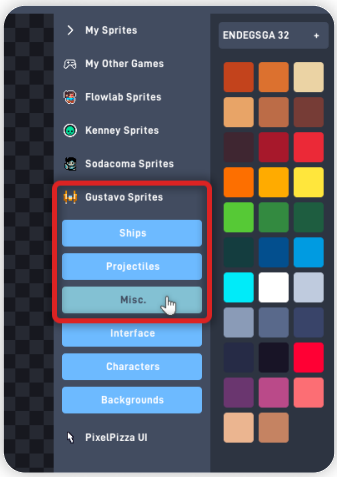


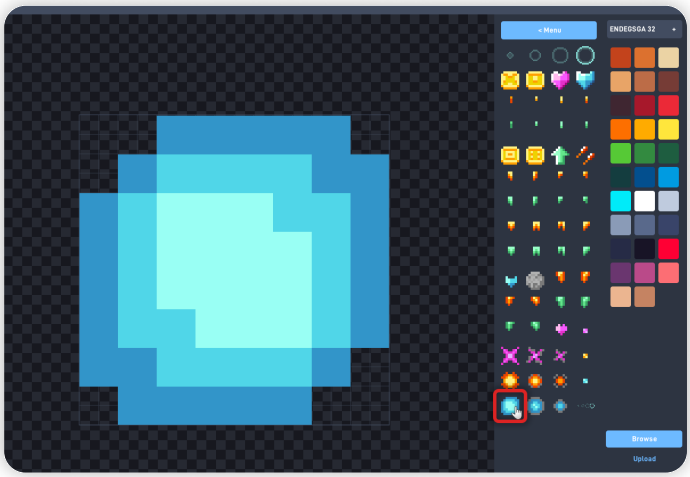


In the Physics tab, **uncheck "Is solid"**, and **enable "Movable".**
Then, **uncheck "Affected by gravity"** and **enable "Enable collisions".**
Click "< Properties" to go back to the properties tab.

Select "Edit Sprite" to open the Sprite Editor.
Click "Browse", "< Menu", "Gustavo Sprites", and "Misc".
From this section, select the Blue Portal sprite at the bottom of the list.







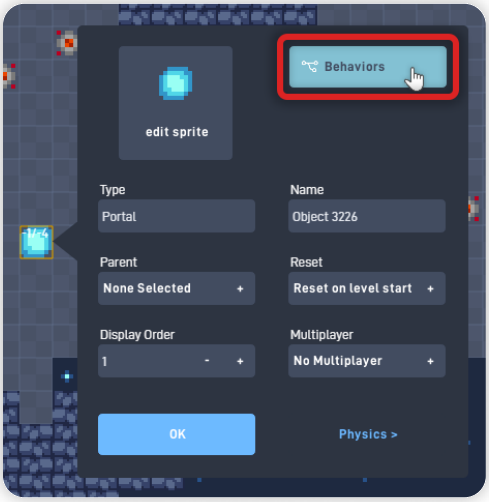
Click "OK" to close the Sprite Editor and save your changes.

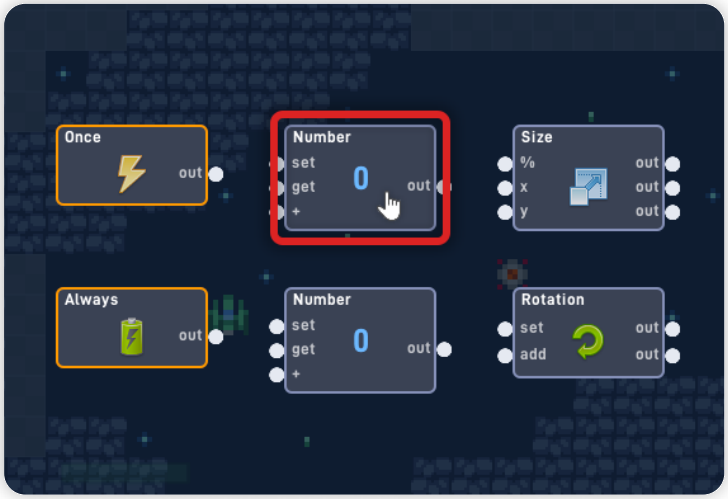
Step 6

Add Logic to the Portal

Click on “Behaviors” to open the Behavior Editor.
From the Triggers section, add a “Once” and an "Always" behavior.
From the Logic & Math section, add two “Number” behaviors.
From the Properties section, add a “Rotation” and a “Size” behavior.

Organize the behaviors as shown.



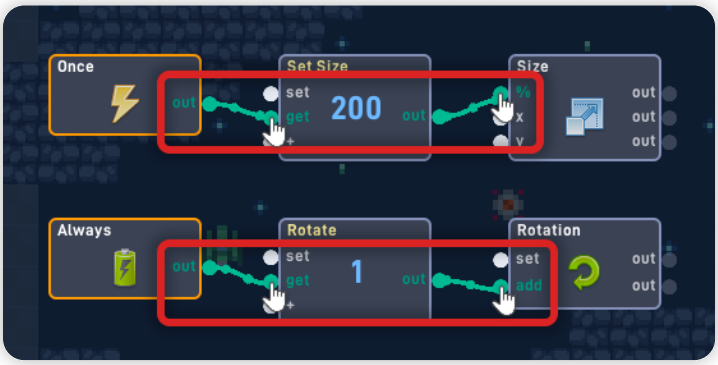


Click on the Number near the Once behavior to open its behavior panel.
Set its **Label to “Set Size” and its Current value to “200”.**
Click “OK” to close the Number behavior panel and save your changes.

Click on the Number near the Always behavior to open its behavior panel.
Set its **Label to “Rotate” and its Current value to “1”.**
Click “OK” to close the Number behavior panel and save your changes.

Now let's connect the behaviors.
Connect the Once "out" to the "get" input from the Set Size Number.
Connect the Set Size Number "out" to the "%" input from the Size behavior.

Connect the Always "out" to the "get" input from the Rotate Number.
Connect the Rotate Number "out" to the "add" input from the Rotation behavior.

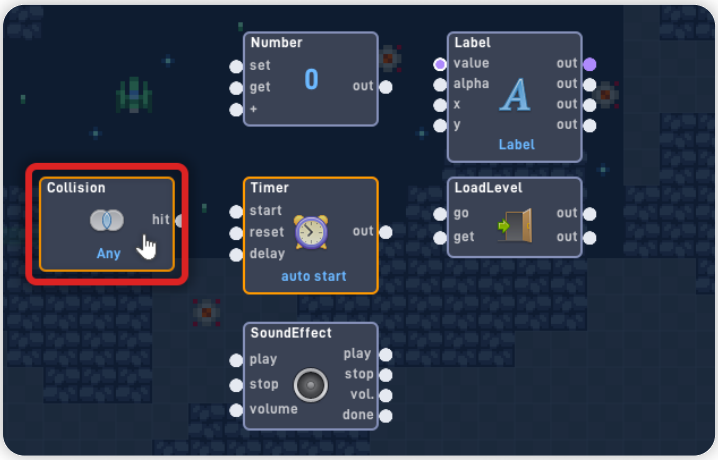


Nice work! This logic bit we just added makes the portal object double its size Once the game starts and rotates the object 1 degree every frame (Always).
This logic also makes the Portal not look so static, making it more appealing and seem more like a magic portal.

> The Numbers behaviors we added work as **Variables**, which allow us to easily change their values and adjust the Logic without needing to recreate it.
>
> You can keep these values, or change its rotation speed (Rotate Number) and its size (Set Size Number) by editing the respective Number behaviors/Variables.

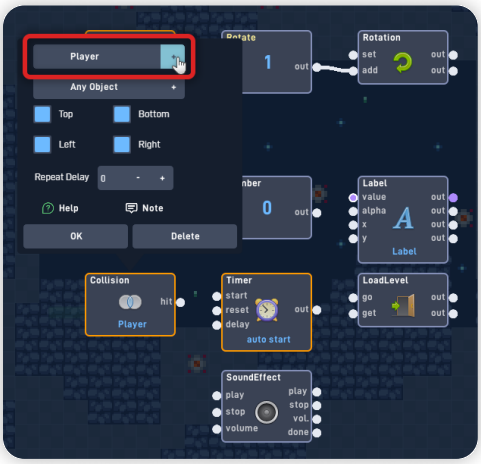
Let's create logic for the Player to proceed to the next level.
From the Triggers section, add a "Collision" and a "Timer" behavior.
From the Logic & Math section, add a "Number" behavior.
From the Components section, add a "Sound" behavior.
From the GUI section, add a "Label" behavior.
From the Game Flow section, add a "Load Level" behavior.

Below the Logic we added before, organize the behaviors as shown.



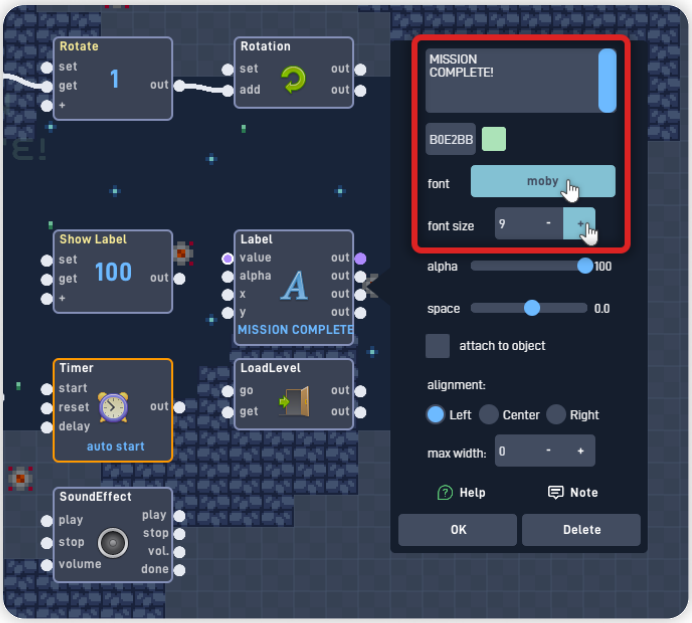
Click on the Collision behavior to open its behavior panel and change the collision from Any Type to “Player”.
Click “OK” to close the behavior panel and save your changes.

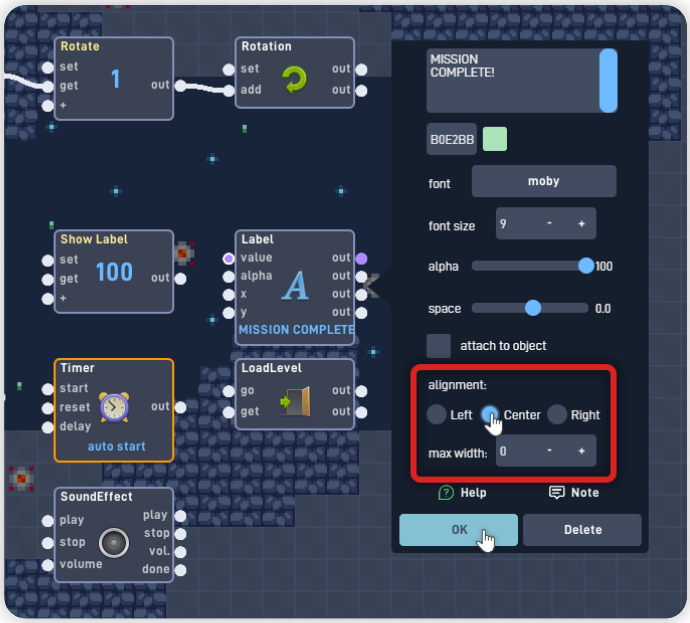
Click on the Number near the Label behavior to open its behavior panel.
Set its Label to “Show Label” and its Current value to “100”.



Now let’s edit the Label that appears when the Player completes the level.
Open the Label behavior panel by clicking on it, and set the Label text to:
“MISSION
COMPLETE!”

Set the Label color by clicking on the colored square, to open the color wheel or pasting the color HEX code.
We are using this **color for the Label:** B0E2BB (Light Green)





Click on the “font button” to select the label font from the list.
We are using “Moby”, but feel free to use a Font of your preference.
Set the font size to “9”.

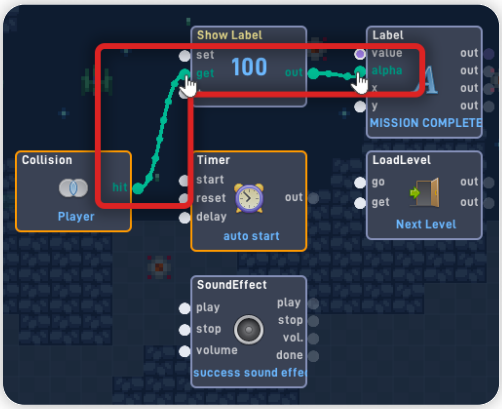
Set the alignment to “Center”, and click “OK” to close the behavior panel and save your changes.

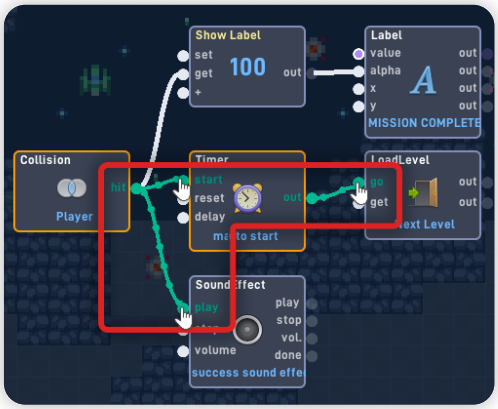
Open the LoadLevel behavior panel, click “Pick Level”, and set it to “Next Level”.
Click “OK” to close the behavior panel and save your changes.

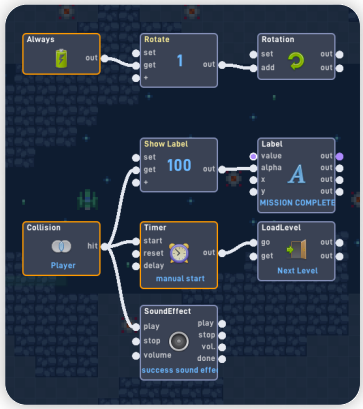
Open the SoundEffect behavior panel, set its Volume to “30”, and click “Choose Sound”.

Navigate through the folders and go to “Effects” and then “Success”.
From that folder, select **“Success sound effect 11”**.
Click “OK” to confirm your sound choice, and click “OK” again to close the behavior panel and save your changes.

Now let’s connect the behaviors.
Connect the Collision “hit” to the “get” input from the Show Label Number.
Connect the Show Label Number “out” to the “alpha” input from Label behavior.







Connect the Collision “hit” to the “start” input from the Timer behavior.
Connect the Timer “out” to the “go” input from the LoadLevel behavior.
Connect the Collision “hit” to the “play” input from the SoundEffect behavior.

Perfect! This new logic bit will trigger when the Player object collides with the Portal object: Showing the “Mission Complete!” Label, playing the “success” sound effect, and one second later, it loads and goes to the Next Level.

Currently, the Label starts visible so we can preview and move it, but in the next step, we will make the Label start invisible/hidden.

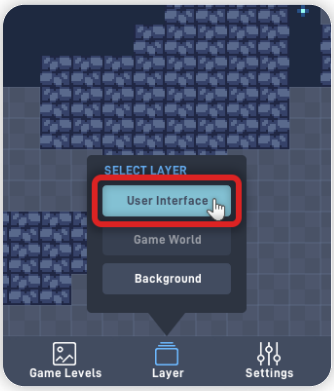
Click “OK” to close the Behavior Editor and save your changes.
Click “OK” again to close the Portal object properties panel.

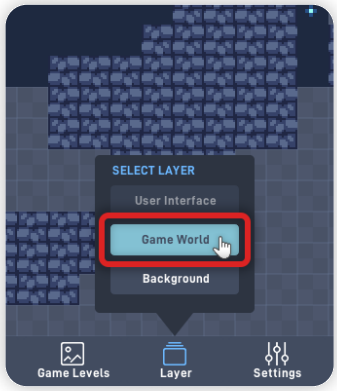
Step 7

Move the Label and Adjust its Alpha

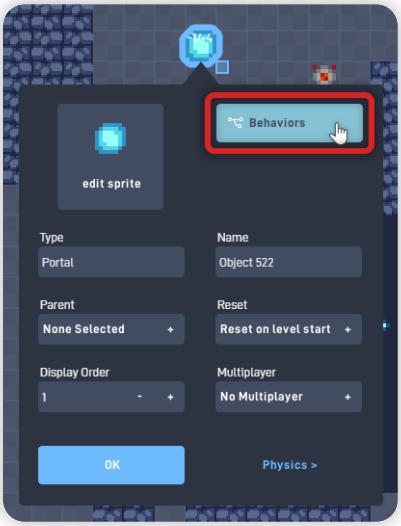
Click “Layer” on the bottom toolbar and change to the “User Interface” layer.
Click, hold, and drag to move the Label and place it in the game view center near the top side.

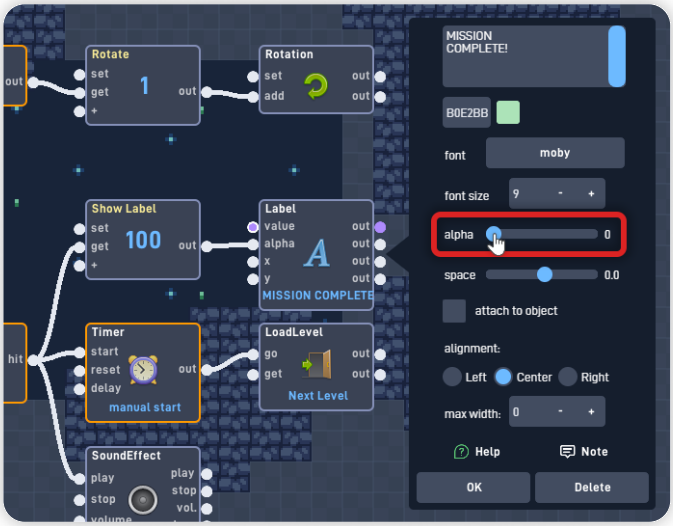
Click “Layer” again on the bottom toolbar and change to the “Game World” layer.





Open the Portal object “Behaviors”.
Click on the Label behavior, and set its Alpha to “0” by dragging the alpha slider.
This change makes the Label start as invisible/hidden and only be shown when triggered by the collision.





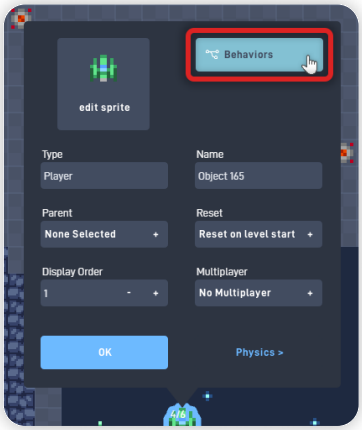
Click “OK” to close the Behavior Editor and save your changes.
Click “OK” again to close the Portal object properties panel.

Step 8

Add Logic to Disable the Player Spaceship

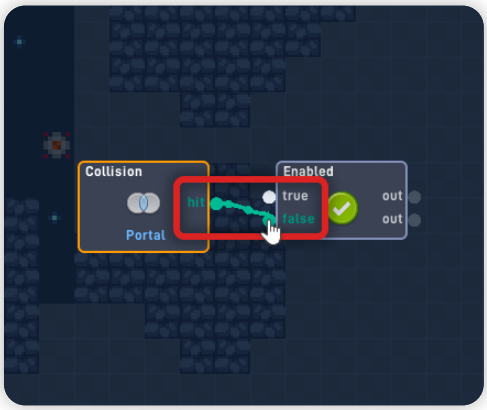
Let’s add logic to the Player object so it “freezes” in place and becomes invulnerable after colliding with the portal/Winning.

Open the Player Behavior Editor by clicking on the Player object, selecting “Edit”, and then clicking “Behaviors”.



From the Triggers section, add a “Collision” behavior.
From the Properties section, add an “Enabled” behavior.
Align the behaviors side by side as shown.

Open the Collision behavior panel and set the collision to “Portal”.
Click “OK” to close the behavior panel and save your changes.



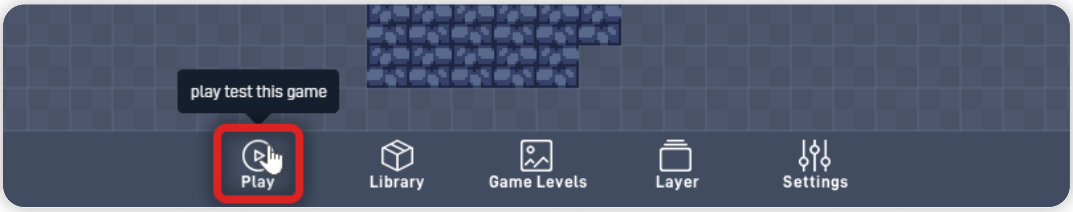
Connect the Collision “hit” to the “false” input from the Enabled behavior.

Fantastic! This new logic bit triggers once the Player collides with the Portal object.
Which disables the Player object physics, making it unmovable and turning off its collisions.

If the Player object has no collisions, it will be ‘invulnerable’ to the explosions/damage logic and ensures that the Player only collides and activates the Portal Logic once.

Click “OK” to close the Behavior Editor and save your changes.
Click “OK” again to close the object properties panel.

Click "Play" on the bottom toolbar to play your game.



When in Play mode:
After the Player reaches the end of the Level and after colliding with the Portal:

- The player stops in place;

- The portal plays the “Success” Sound, shows the “Mission Complete!” Label, and one second later, it takes the Player to the next Level;

If you run into any 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 "Mission Complete!" Label shows at the start of the game,** instead of only showing when the Player completes the level: Make sure that on the Portal object Behaviors, the Label behavior panel has Alpha set to "0"; *(Step 7)*

- **If the Player collisions with the Portal aren't working as expected:** Ensure that on the Portal Properties panel, on the Physics tab, the "movable" and "enable collisions" are selected, and "is solid" and "affected by gravity" are unselected; *(Step 5)*

- **If the Player collides with the portal, but the game doesn't proceed to the Next Level:** Ensure that there is another Level the game can load on the Level panel. Also ensure that on the Portal object Behaviors, the LoadLevel behavior has "Next Level" selected; *(Step 6)*

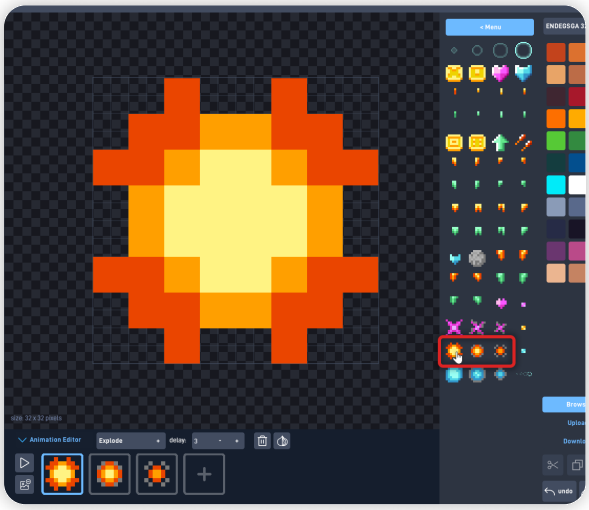
Optional Game Enhancements

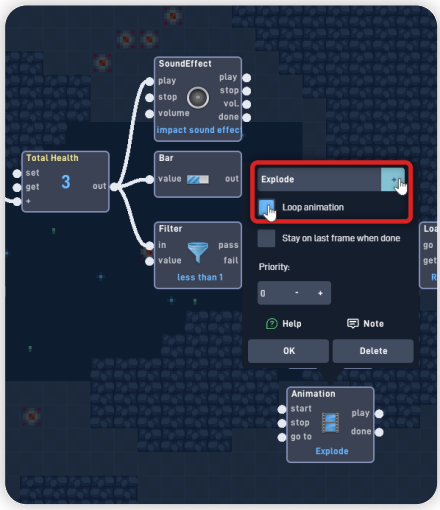
Once you have this lesson's game working, here is a simple enhancement to try:

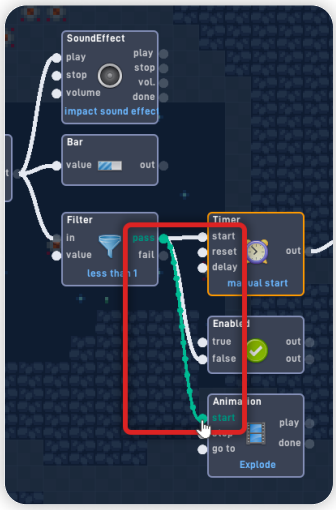
- **Create a new & different Level Course for the Level 2.** Go to "Level 2" through the Levels panel, and using the Library, add the objects necessary to complete a new level course: A Portal, Wall Rocks, and Trap objects.

If the new level becomes more extensive than the first, you can adjust the Camera Boundaries inside the Player object Behaviors, as we did before in Lesson 2.

- **Add an Explode animation when the Player loses.**
  Open the Player Sprite Editor, and create an explosion animation using the Explosion sprites from the “Gustavo Sprites > Misc” pack from the Browse panel.
  Set the animation Delay to “3” and click “OK” to save your Sprite changes.







Then, inside the Player Behaviors and the “Health / Lose” bundle, add an Animation behavior below the Enabled behavior.

Open the Animation behavior panel, set the animation to “Explode”, and select “Loop Animation”.
Click “OK” to close the Animation behavior panel and save your changes.

And finally, connect the Filter “pass” output to the “start” input from the Animation behavior.
This logic will make the Explode Animation play when the Player Health reaches 0.

- **Add additional Levels** to your game by repeating Step 3.

Space Pilot - Part 1

Nice work!

You’ve now finished **Lesson 6 out of 6.**