Lesson 4: Sword Attack & Hitboxes

Objective: Using custom Logic, create and attach a Sword object with an Attack animation and Damage Area.

Time: 30 Minutes

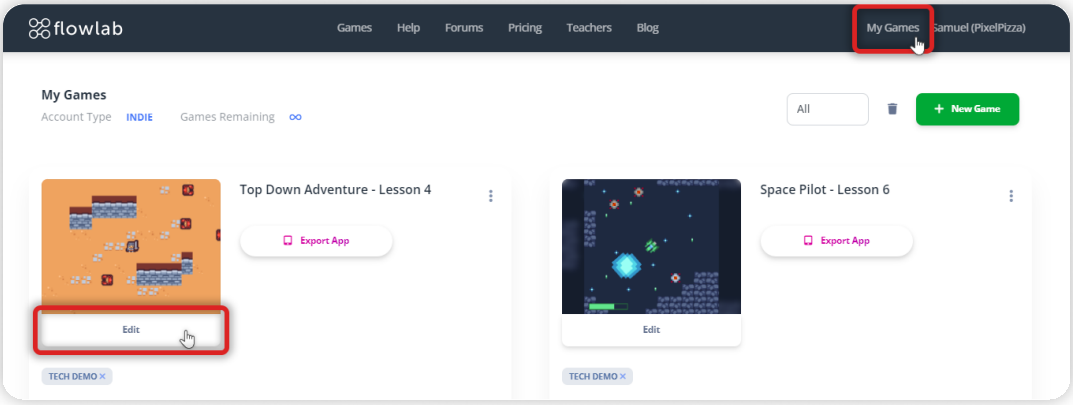
Level: 2.5 - Beginner/ Intermediate

Description: Introduction to the Attacher and Emit behaviors. We'll create a dynamic Sword object that plays a Procedural animation and emits a Hitbox/ Damage Area.
"Procedural animations" are a way for Game Developers to bring more life to their game with animations, but using logic instead of manually drawing every frame.

Step 1

Edit your Game

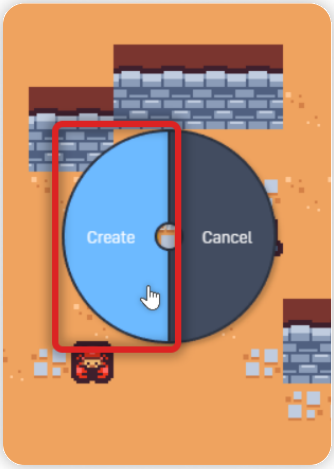
Log in and start at your "My Games" page https://flowlab.io/game/list
Then, click "Edit" next to your game to open the game editor.



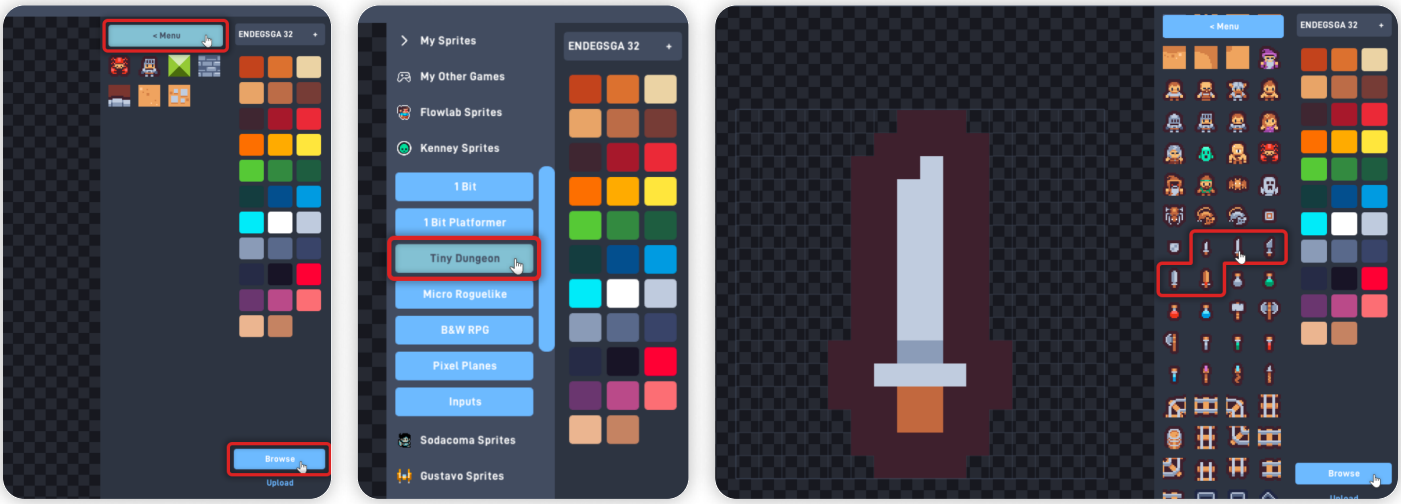
Step 2

Create a Sword Object

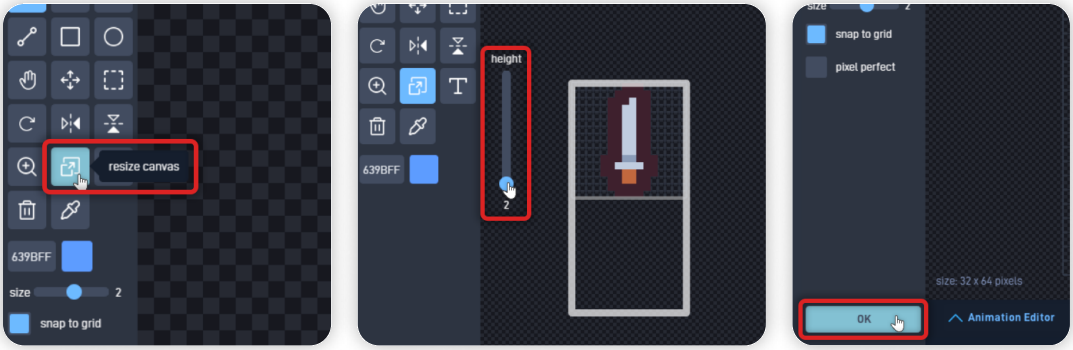
Inside the Game's Level editor, click anywhere inside the visible game area and select "Create" from the circle menu to create a new object.
Name this object by setting its Type to "Sword".
Click "edit sprite" to open the Sprite Editor and change this object sprite.



Inside the Sprite Editor, open the "Browse" panel and click on "< Menu" to open the Sprite Collections.
From the "Kenney Sprites" collection, open "Tiny Dungeon" and select a Weapon Sprite. *We picked the "Regular Sword" sprite, but you can choose your preferred weapon sprite (from the highlighted section shown below).*



Click "Browse" to close the Browse panel, and click on the "Resize Canvas" tool to change the sprite's dimensions.
Set the Sprite Height to "2", and then change back to the "Pencil" tool to confirm the new dimension.
Press "OK" to save your changes and close the Sprite Editor.



On the Sword Object's Properties panel, set the Display Order to "2".
This change makes this object display above the Player object.
Click on "Physics >" to go to the Physics tab, then **uncheck "is solid"**, **check "movable"**, and **uncheck "affected by gravity"**. Click "OK" to close the Object Properties panel.

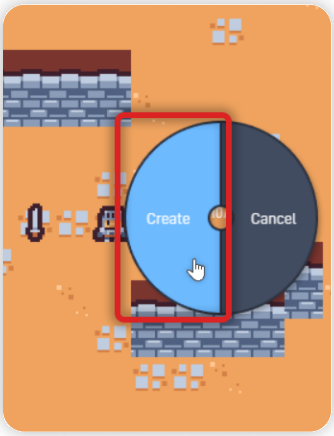
Step 3

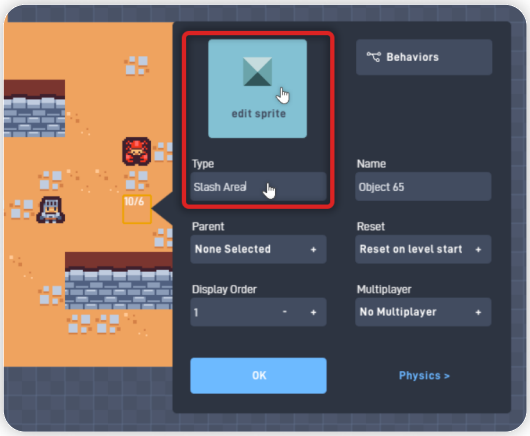
Create a Damage Area Object

Click inside the visible game area, and select "Create" from the circle menu.
Name this object by setting its Type to **"Slash Area"**.

Now, click "edit sprite" to open the Sprite Editor and change this object sprite.
From the Color Palette on the side, select a Red color.
You can use any Red color tone, but we used this **Alerting Color:** FF0044 *(Red)*

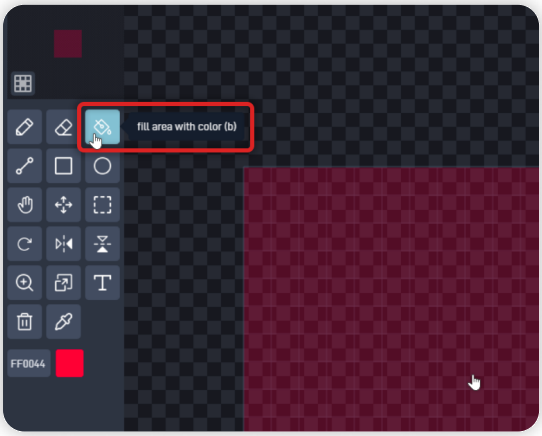
Open the Color Picker by clicking on the "Red Square".
Reduce the color Alpha to about "25%" and click "OK" to confirm the color selection.

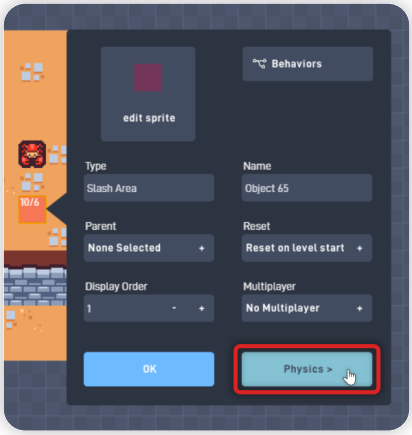


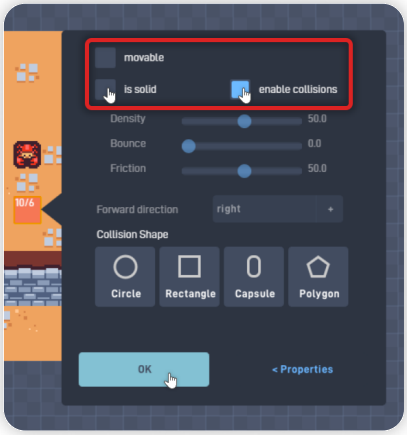


Click on the Bucket icon to select the "Bucket Tool".
Then, click inside the Sprite repeatedly to **paint & fill the entire Sprite with this color.**
Click "OK" to close the Sprite Editor and save your changes.

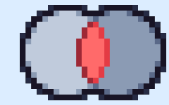
Back on the Object's Properties panel, click on "Physics >" to go to the Physics tab,
then **uncheck "is solid"**, and **check "enable collisions".**
Press "OK" to close the Object Properties panel.







Hitboxes and Damage Areas:
"Hitboxes" or Damage Areas are a term used in Game Development, usually associated with an Attack, and ***they define where that attack is effective.***

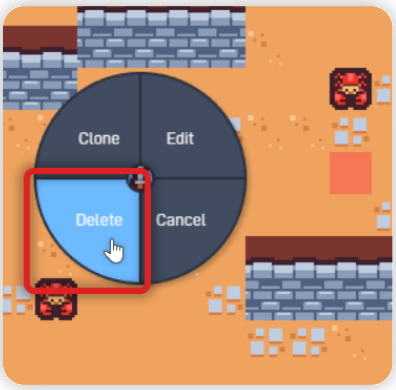


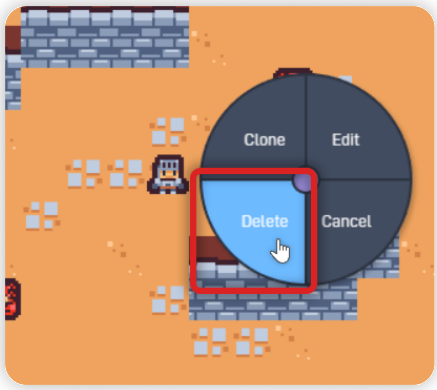
In the next Lesson, we make the "Slash Area" hit and damage the Enemy object.
We can detect when the Hitbox collides with the "hittable" Object using Collisions.
In this case, we will make the Enemy react/lose Health when Colliding with it.

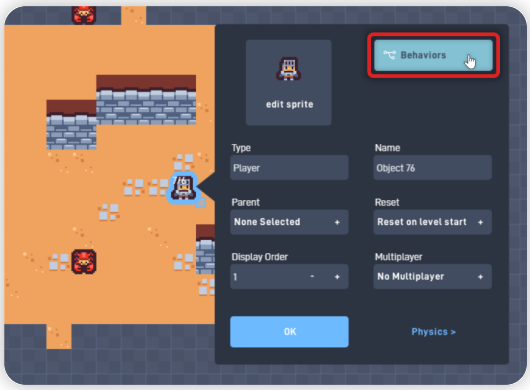
Step 4

Delete Objects from the Level & Add Custom Logic to the Player
Click on the **"Sword" object** and select "Delete" from the circle menu to delete this Object from the Level.
Click on the **"Slash Area" object**, and select "Delete" from the circle menu.
You can still access these objects from the "Object Library" on the bottom toolbar.

Now, click on the "Player" Object and select "Edit" from the circle menu.
Click on "Behaviors" to open the Player Behavior Editor.

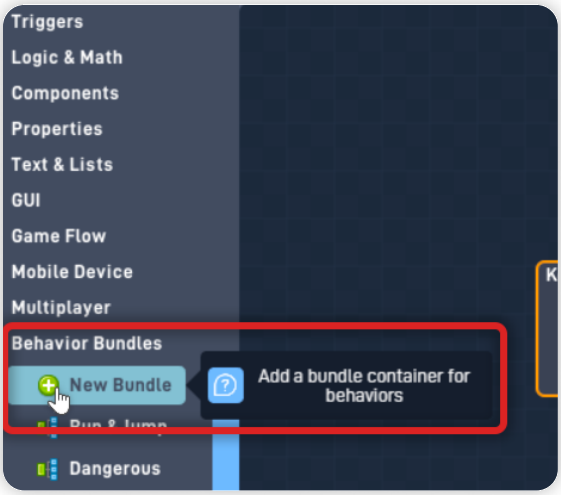






From the Behavior Bundles section, click on **"New Bundle"** to add a new empty bundle.
Place the new Bundle near the other code, then click on the "Bundle" block to open its behavior panel.

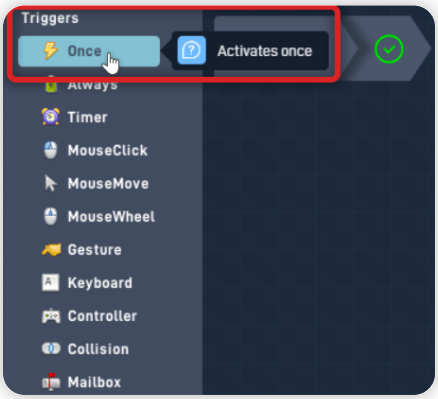
Inside the Bundle"s behavior panel, set its Label to "Sword Logic" and click "Open Bundle".

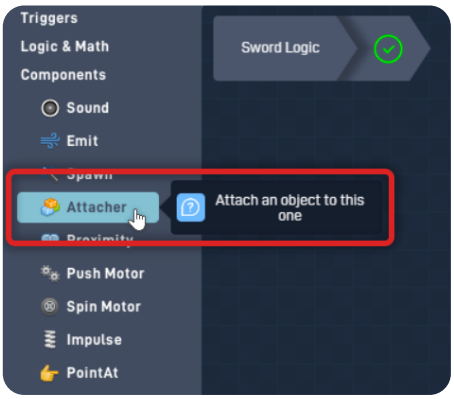


Behavior Bundles help the Game Developer organize their Logic and behaviors, making sections and specific mechanics easier to find, edit, or expand if necessary.

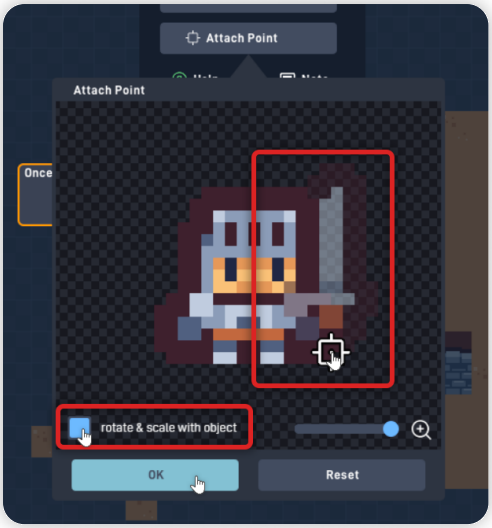
From the Triggers section, **add a "Once"** behavior.
From the Components section, **add an "Attacher"** behavior.
Organize the newly added behaviors as shown below.

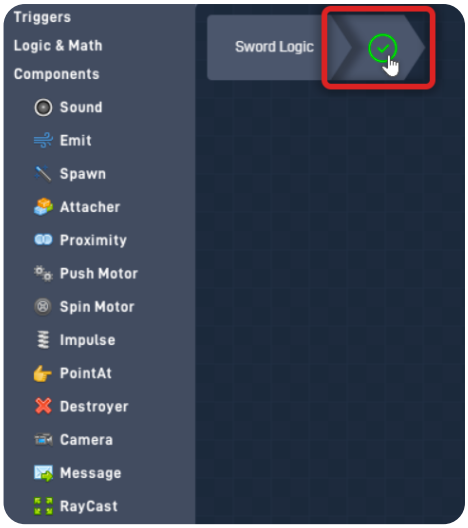
Click on the "Attacher" behavior to open its behavior panel.
Click on "Type to Attach" and **select the "Sword"** object.





Click on "Attach Point" and click inside the Origin Editor to move and choose the area where the Sword object should be attached.
In this case, we want the sword to be near the Player's hand - Using the Origin Editor Preview, we can align both objects.
You can also use the arrow keys to adjust the Origin Point position precisely.

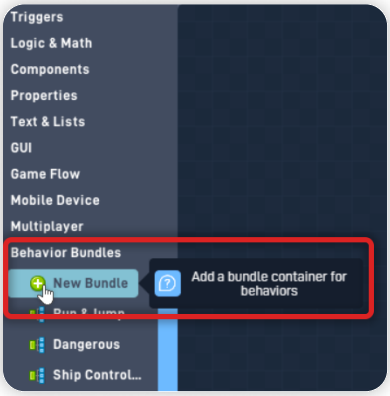


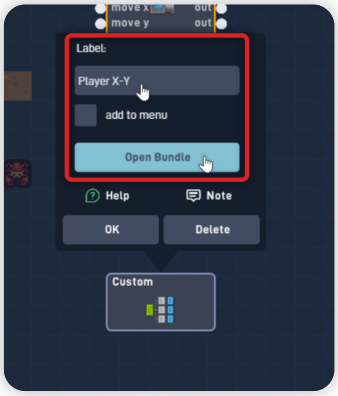


Select the "rotate & scale with object" option.
Click "OK" to close the Origin Editor and confirm the Origin position.
Click "OK" again to close the Attacher Behavior panel.

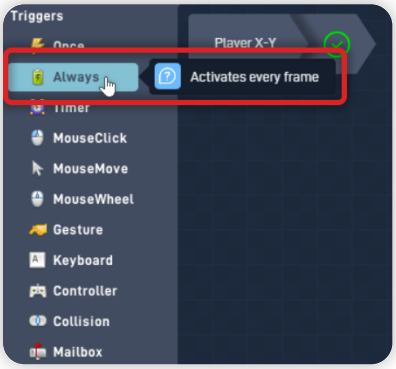
Connect the Once "out" to the "on" input from the Attacher behavior.
Click on the "Green Check" in the top corner to close the Bundle.

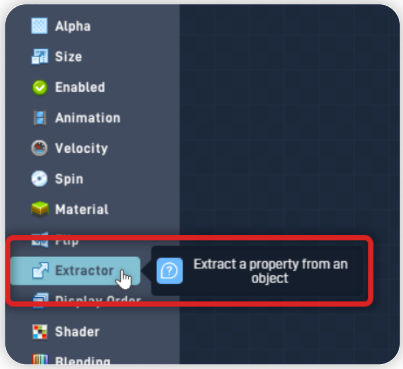
From the Behavior Bundles section, click on "New Bundle" to add a new empty bundle.
Open the Bundle"s behavior panel, set its Label to "Player X-Y", and click "Open Bundle".

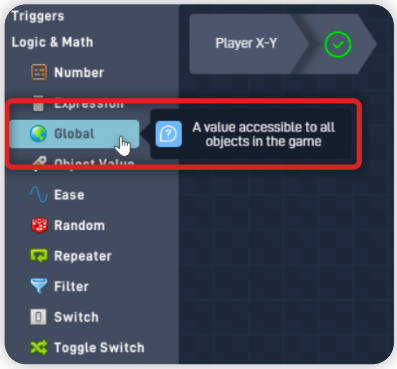




From the Triggers section, **add an "Always"** behavior.
From the Properties section, **add two "Extractor"** behaviors.
From the Logic & Math section, **add two "Global"** behaviors.
Organize the newly added behaviors as shown below.





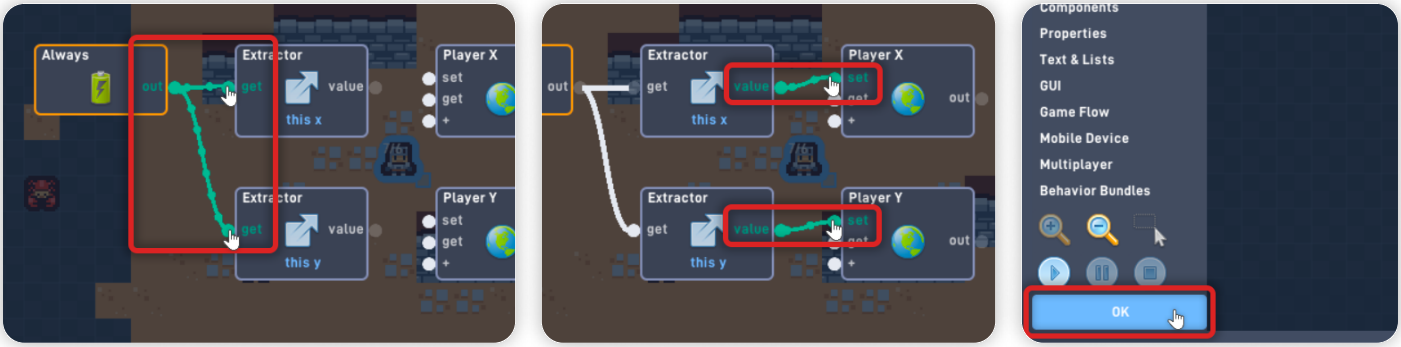


Click on the Extractor from the second Row to open its Behavior panel.
Click "X" to select the properties this behavior should extract, and select "Y".
Click "OK" to close the Behavior panel.

Open the Behavior panel from the "Global" on the first row.
Click on "Global Name" and select "Add New Name", **set the Global Name to "Player X"**.
Click "OK" to confirm the name and click "OK" to close the behavior panel.

Open the Behavior panel from the "Global" on the second row.
Click on "Global Name" and select "Add New Name", **set the Global Name to "Player Y"**.
Click "OK" to confirm the name and click "OK" to close the behavior panel.

Now, let's connect the behaviors.
Connect the Always "out" to the "get" input from both Extractors behaviors.
Connect each Extractor "value" output to the "set" input from its respective Global.



This logic (Always) triggers every frame and extracts the Player's object X and Y coordinates.
X and Y coordinates are the positions of an object in the two-directional axis, X being the left/right axis and Y being the up/down axis.

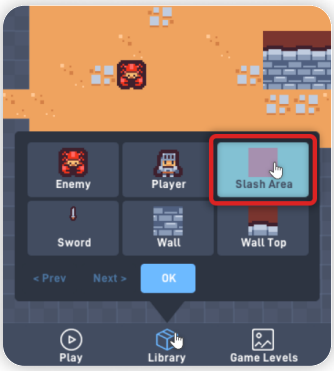
Then, the extractors output the Object's position in pixels and set their Values to the respective Global.
Globals are variables that any object with that Global Behavior can access.

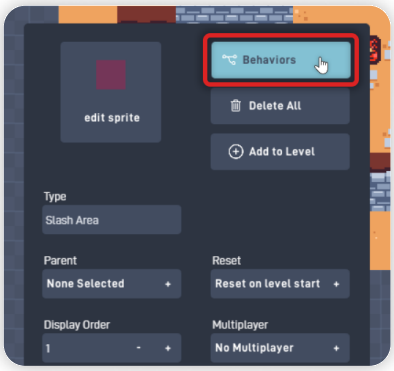
Click "OK" to close the Behavior Editor, and click "OK" to close the Player object properties panel.

Step 5

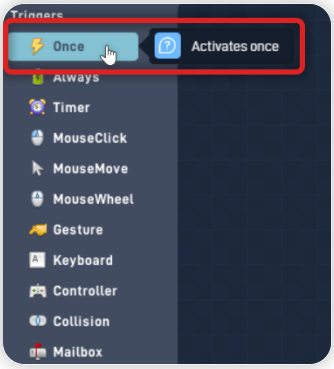
Add custom Logic to the Slash Area

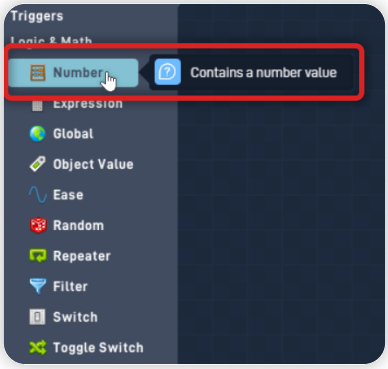
Click on "Library" on the bottom toolbar, and select the "Slash Area" object.
Click on "Behaviors" to open the Behavior Editor.

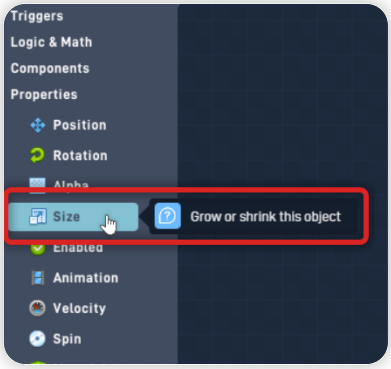




From the Triggers section, **add a "Once"** behavior.
From the Logic & Math section, **add a "Number"** behavior.
From the Properties section, **add a "Size"** behavior.
Organize the newly added behaviors from left to right, by the order they were added.

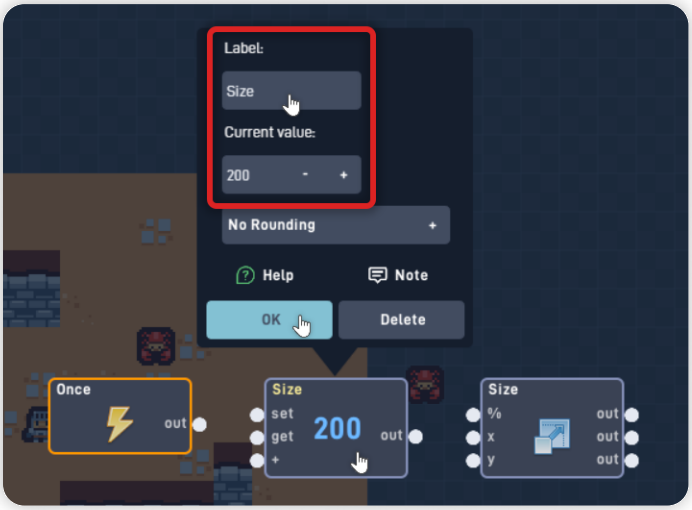




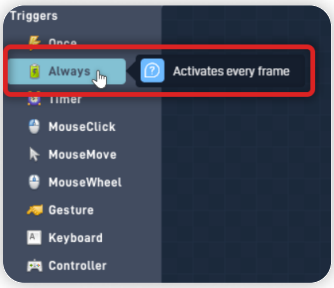


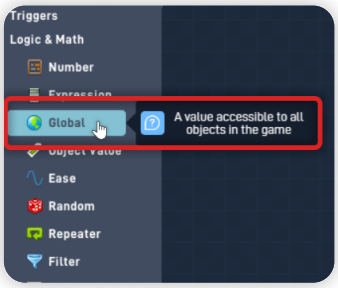
Open the "Number" Behavior panel, set its Label to "Size", and set its Current Value to "200".
Click "OK" to close the behavior panel and save your changes.

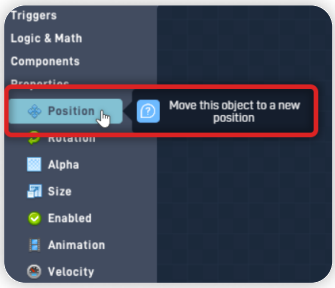
Connect the Once "out" to the "get" input from the Number behavior.
Connect the Number "out" to the "%" input from the Size behavior.

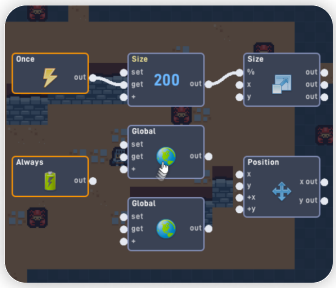


From the Triggers section, **add an "Always"** behavior.
From the Logic & Math section, **add two "Global"** behaviors.
From the Properties section, **add a "Position"** behavior.
Organize the newly added behaviors as shown below.



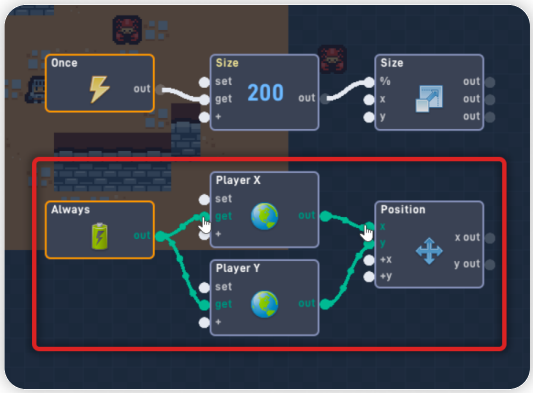






Open the upper "Global" Behavior panel, click on "Global Name", and select "Player X".
Click "OK" to close the behavior panel.
Open the bottom "Global" Behavior panel, click on "Global Name", and select "Player Y".
Click "OK" to close the behavior panel.

Connect the Always "out" to both "get" inputs from the Globals behaviors.
Connect the "Player X" Global to the "x" input from the Position behavior.
Connect the "Player Y" Global to the "y" input from the Position behavior.



This logic triggers Once the game starts, it reads the "Size" Number value and sets this object's Size. In this case, the Size is set to 200% of the original object's Size, making it two times bigger than it was.

Since the Slash Area object is the same Size as the Player's sprite, making the Slash Area bigger allows the Player to attack what's around it in all directions.

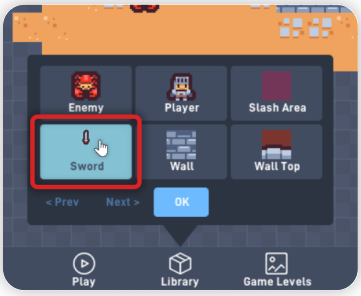
The Always triggers every frame and reads both Global values, Player X and Player Y, which then sets this object X and Y Position.
Making the Slash Area always be in the same Position as the Player allows the Player to attack and move simultaneously without "leaving the Slash Area behind".

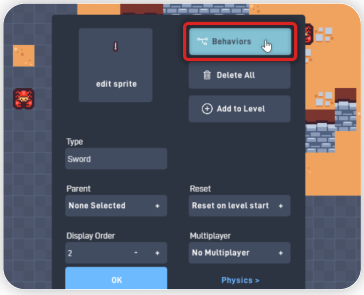
Click "OK" to close the Behavior Editor.
Click "OK" to close the behavior panel, and click "OK" again to close the Slash Area Object properties panel.

Step 6

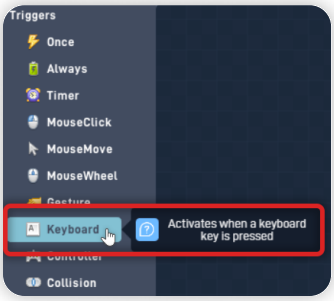
Using Custom Logic, create a Sword Attack

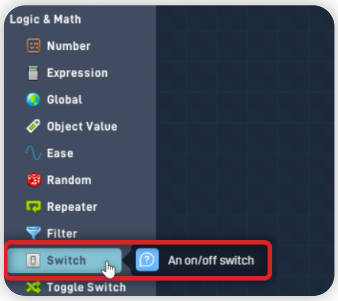
With the Object Library open, select the "Sword" object and click on "Behaviors".

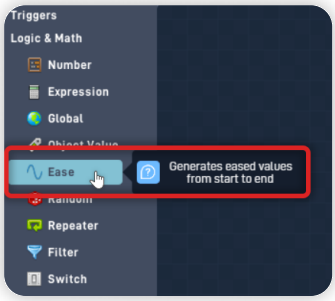


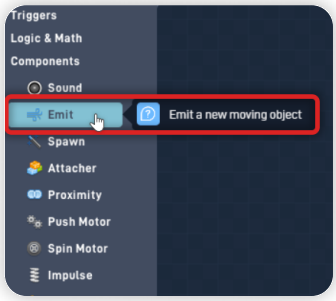


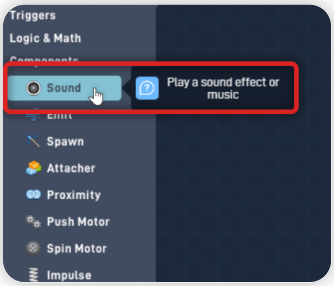
From the Triggers section, **add a "Keyboard"** behavior.
From the Logic & Math section, **add a "Switch"** behavior and **an "Ease"** behavior.
From the Components section, **add an "Emit"** behavior and **a "Sound"** behavior.
From the Properties section, **add a "Rotation"** behavior.
Organize the newly added behaviors as shown below.





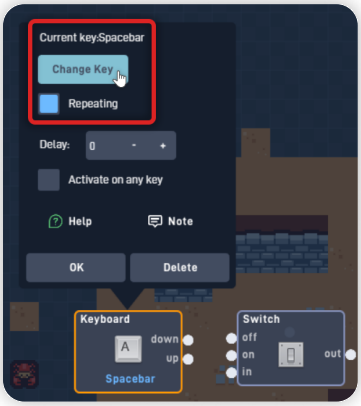


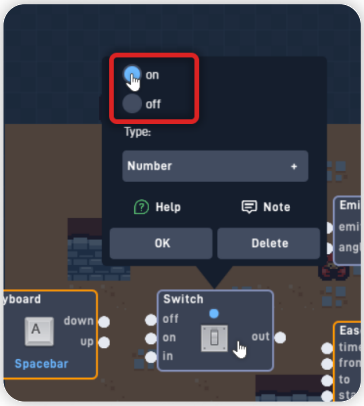




Open the "Keyboard" Behavior panel, click "Change Key", and press "Spacebar" on your keyboard.
Select the "Repeating" option and click "OK" to close the Behavior panel.

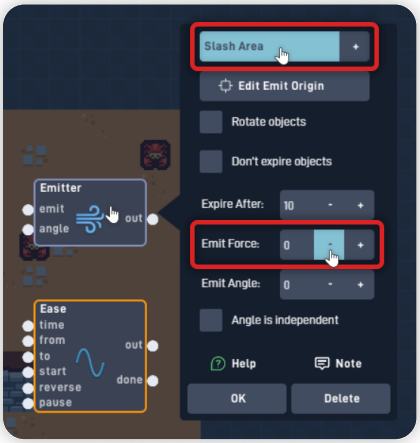
Open the "Switch" Behavior panel and **set its initial state to "On".**
Click "OK" to close its behavior panel.

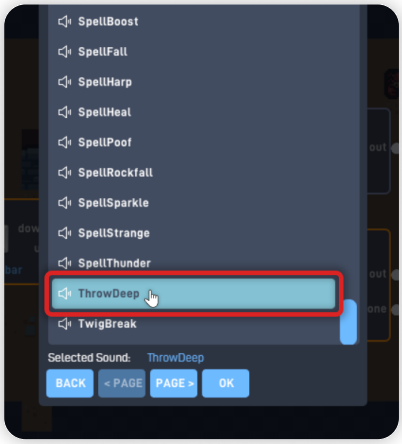




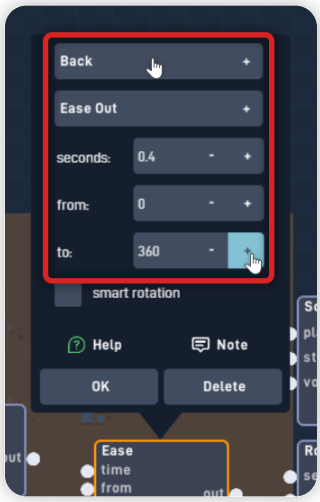
Open the "Emitter" behavior panel, click on "Object to Emit", and select the "Slash Area" object.
Set the Emit Force to "0", and click "OK" to close the behavior panel.

Open the "SoundEffect" behavior panel and change its Volume to "25".
Click on "Choose Sound", and navigate through the following folders:
Effects > Premium > IndiesFx > Adventure and select the **"ThrowDeep"** sound at the end of the list.
Click "OK" to confirm the selected sound and click "OK" to close the behavior panel.

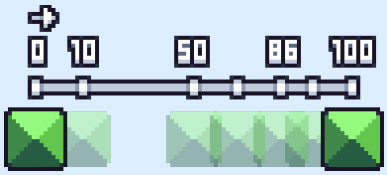




Open the Ease behavior panel, click "Quadratic" to change its Easing Formula, and set it to "Back".
Click on "Ease In" and set it to "Ease Out".
Set the “Seconds” to "0.4", and set the "To" value to "360".
Click “OK” to close the behavior panel.

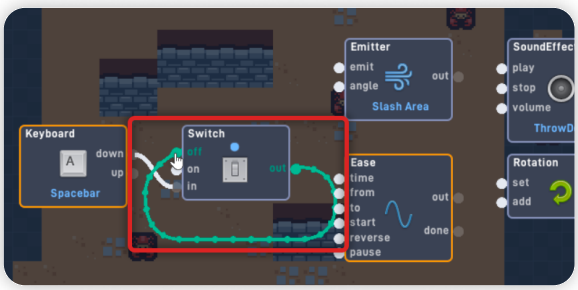


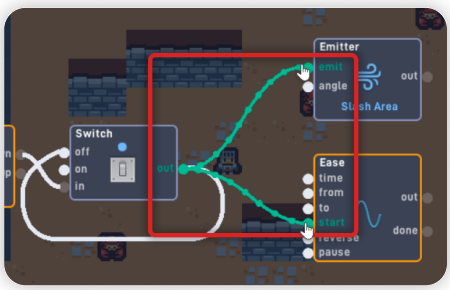
Ease behaviors generate and output ***a set of interpolated (or eased) values between two numbers.***
This is sometimes also called ***"Tweening"*** when used for animations.
Easing is a great way to smoothly animate objects or their properties, in this case, to make an object rotate 360 degrees.



A visual example of Interpolated numbers from 0 to 100

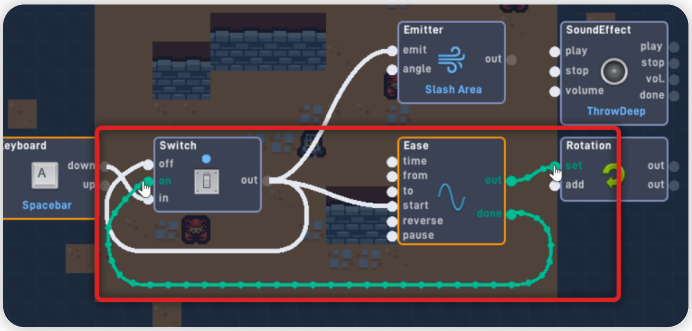
Now, let’s connect these behaviors together.
Connect the Keyboard "down" to the "in" input from the Switch behavior.
Connect the Switch "out" to its own "off" input.
Connect the Switch "out" to the "emit" input from the Emit behavior.
Connect the Switch "out" to the "start" input from the Ease Behavior.

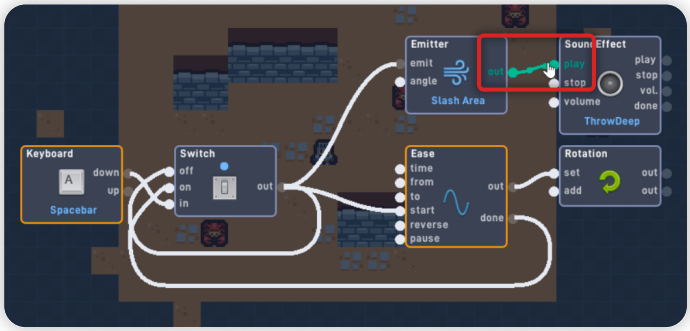




Connect the Ease "out" to the "set" input from the Rotation behavior.
Connect the Ease "done" to the "on" input from the Switch Behavior.

Connect the Emitter "out" to the "play" input from the SoundEffect behavior.





This logic triggers every time the Player presses the Spacebar, triggering the Switch, which turns itself Off - This Switch logic makes it so the player can only start this logic once until the Switch is turned back on.

Once activated, the Switch triggers the Emitter behavior, emitting a Slash Area and playing a "swoosh" Sound Effect.
The Switch also starts the Ease, which continuously outputs values from 0 to 360 until it reaches its End value and sets the Sword's object rotation.

Once the Ease animation is "done", the Ease "done" output will trigger and turn the Switch back on, allowing the Player to start this Logic and Attack again.

This Ease-Rotation creates an animation that makes it seem like the Player is slashing the Sword, creating a **"Code Animation"** - An animation made through code that didn't require the Developer to create a Sprite Animation.

Click "OK" to close the Behavior Editor and save your changes.
Click "OK" to close the Object Properties panel and "OK" again to close the Object Library.

Step 7

Playtest your Game
Now, click the "Play" button in the bottom toolbar to play your game.

When in play mode:
- The Sword object follows the Player object and its orientation, so when the Player faces left, the Sword matches its rotation.

- Pressing the "Spacebar" key makes the Sword object play its "Code Animation", rotating 360º degrees, emitting the Slash Area (Damage Hitbox), and playing a "Swoosh" sound effect.

- If the Player attacks while moving, the Slash Area object will match its Position (at the center of the Player Sprite).

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 expected. If your game is misbehaving, check the following points:

- **If the Slash Area object is pushing away the Enemy object,** ensure that on the Slash Area object Physics tab, the "is solid" option is unselected and "enable collisions" is selected; *(Step 3)*
- **If the Sword object doesn't match the Player object's orientation,** ensure the "rotate & scale with object" option is selected inside the Attacher behavior from the "Sword Logic" bundle on the Player object's Behaviors; *(Step 4)*
- **If the Slash Area doesn't match the Player's position,** ensure that the Extractor behaviors are connected to their respective Globals. *(Step 5)*
- **If the Player can attack only once,** ensure the Ease behavior "done" output is connected to the "on" input from the Switch behavior, in the Sword object's Behaviors; *(Step 6)*
- **If the Player can't attack at the start of the game,** ensure the Switch behavior has its initial state set to "On", on the Sword object's Behaviors; *(Step 6)*

Optional Game Enhancements

Once you have this Lesson's content working, here are some simple enhancements to try:

- Create the rest of the Game's Level Area by creating new Wall objects and decorating the level with Sprites from the "Tiny Dungeon" Asset Pack.

  *Remember that you can reuse already created objects by "cloning" them or adding them through the **Library on the bottom toolbar.***

Important Note!: Ensure that all Wall objects have their "Parent" set to the original "Wall" object, as we did in the previous lesson - "Top Down Adventure" Lesson 3.

Top Down Adventure - Part 1

Nice work!

You've now finished **Lesson 4 out of 6.**