



Maydm

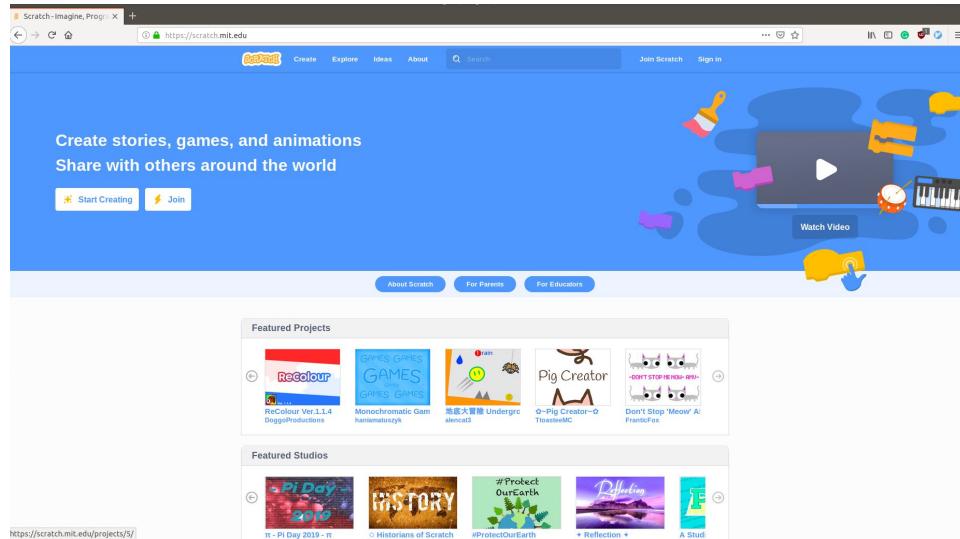
—

# Build a Game: Fly Gal

# Building a Game

---

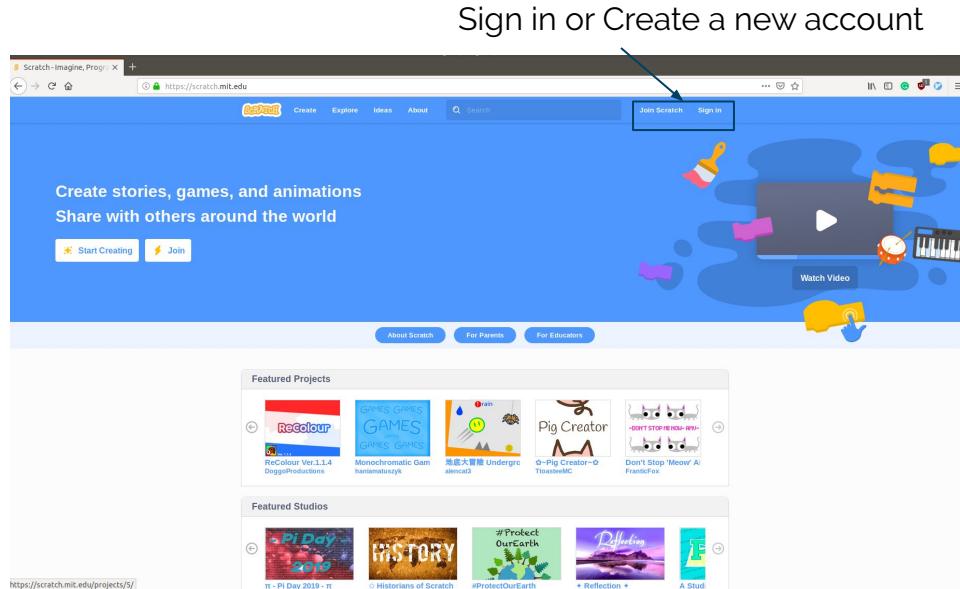
We're going to build a 2D side scroller where a character flies between moving barriers and collects gems to increase her energy.



# Open Scratch

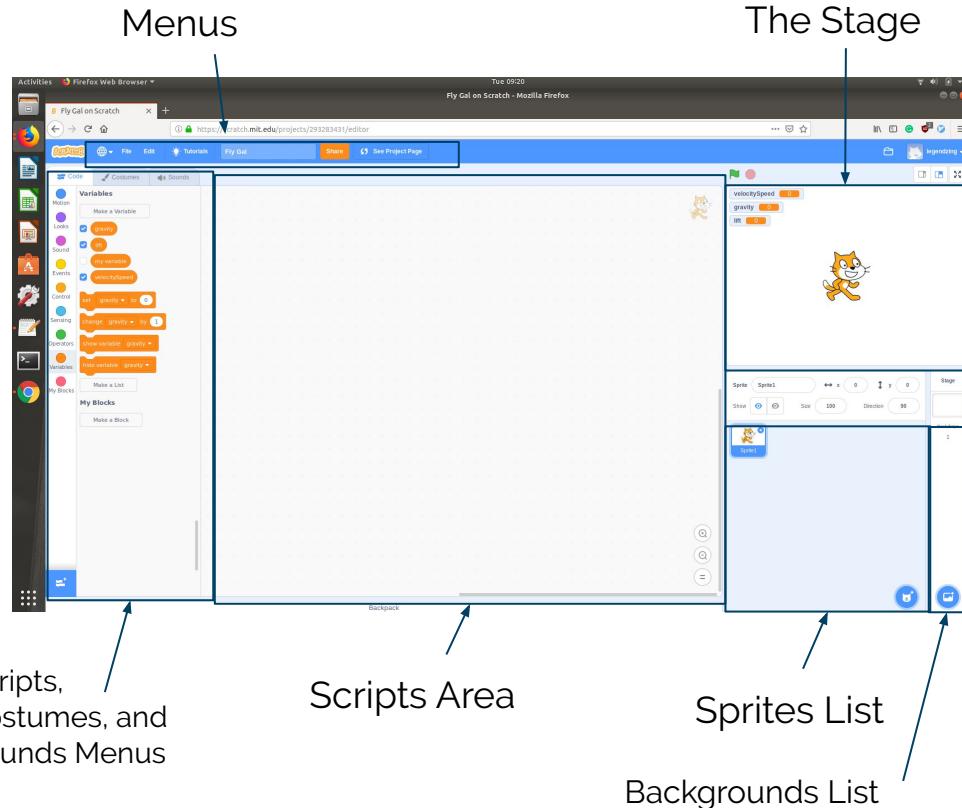
---

If you have a Scratch account then log in at scratch.mit.edu  
If you don't have an account then create one.



# The Parts of Scratch

- Menus
- The Stage
- Scripts Area
- Scripts, Costumes, and Sounds Menus
- Sprites List
- Backgrounds List

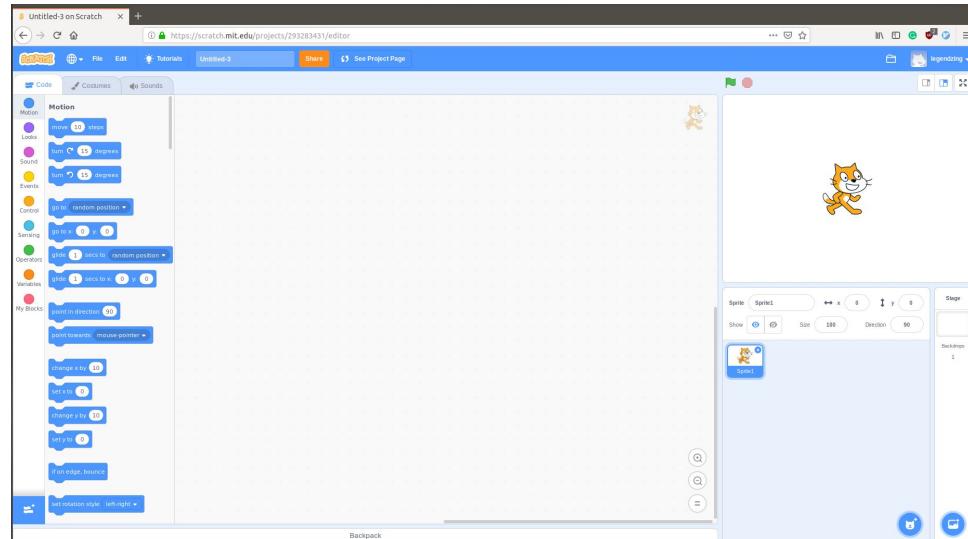


# Create a New Project

---

After logging in,  
click the “Create” link to  
open a new project.

Let's name the project  
“Fly Gal”

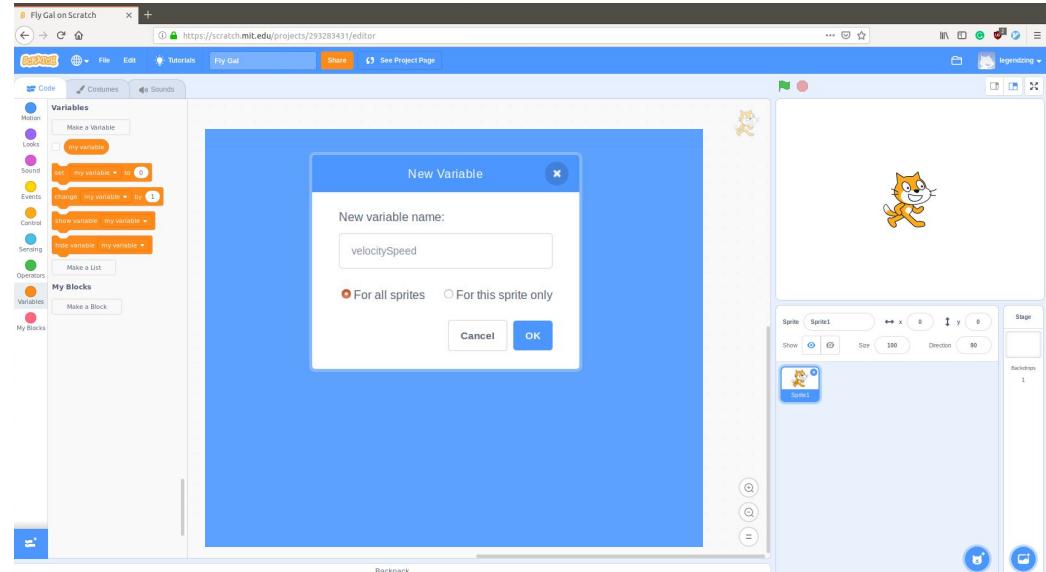


# Create The First Variable

---

In this project we will incorporate much of what we've learned about variables, if-then statements and loops.

Our first step is to create a new Var “velocitySpeed” for all sprites



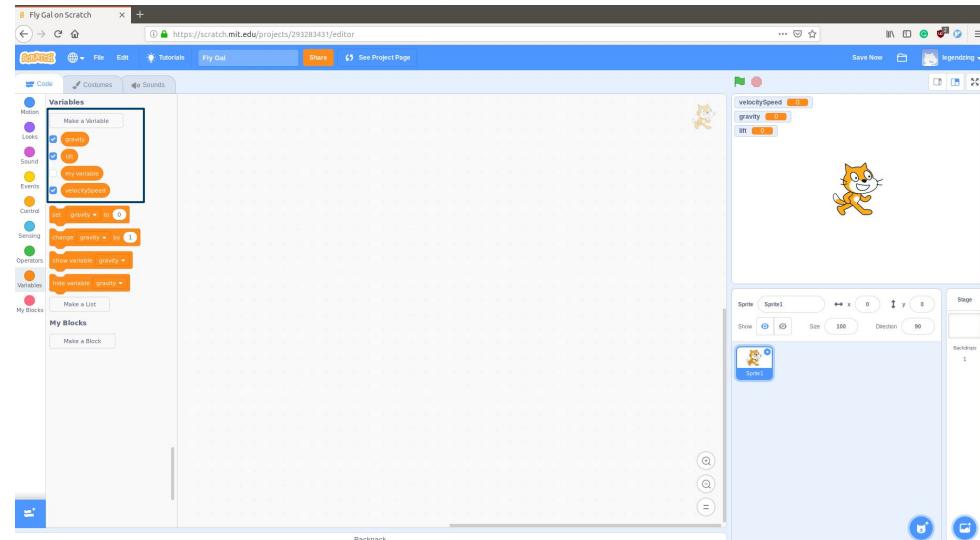
# Create Two More Variables

---

**velocitySpeed** will track  
the speed of our character.

We'll create two more  
variables, "gravity" and "lift"  
to set the world's gravity and  
our characters force against  
said gravity.

These variables will be  
for all sprites.

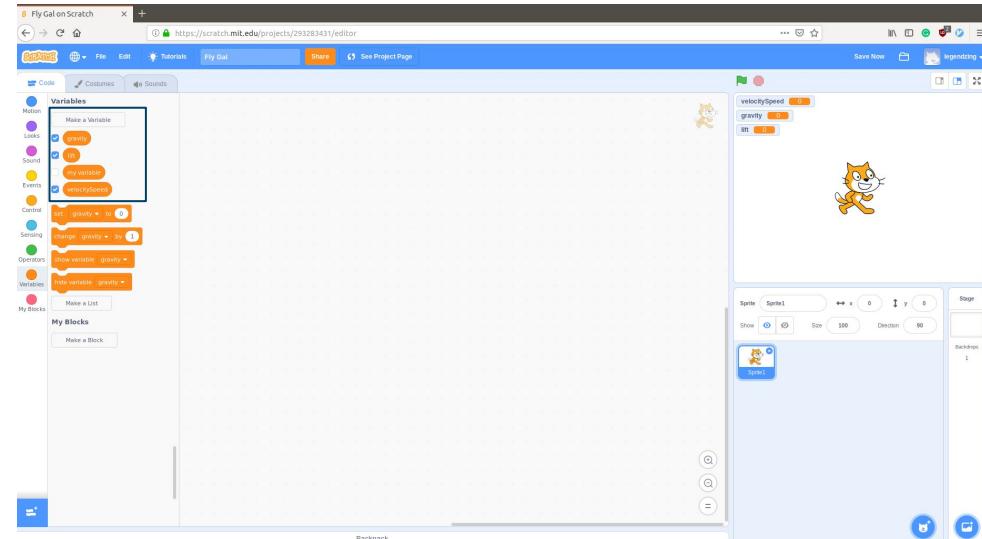


# Write our First Code Block

---

Let's write our first code block by dragging blocks of code to the Scripts Area.

We'll use the Control, Variables, Motion and Events categories to build this first block.



# Write our First Code Block

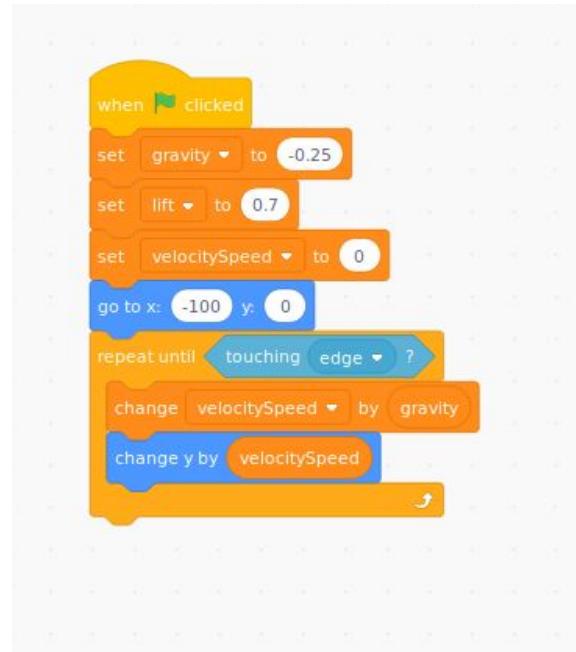
---

Our first block will establish the three variables that we have created and place our character on the screen.

The statements inside the loop cause the character to fall down along the y axis.

Be sure to drag the variables from the Variable store. Writing in Variables will not work!

Once you've built the block, run the script by clicking on the green flag.



# Run the Code

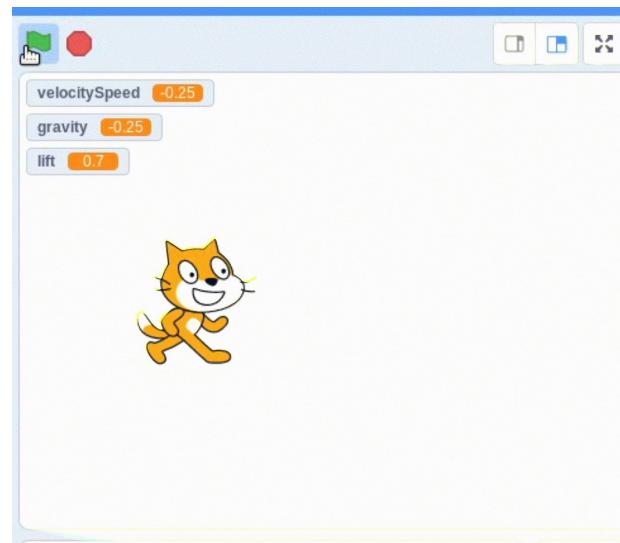
---

Our first block will establish the three variables that we have created and place our character on the screen.

The statements inside the loop cause the character to fall down along the y axis.

Be sure to drag the variables from the Variable store. Writing in Variables will not work!

Once you've built the block, run the script by clicking on the green flag.



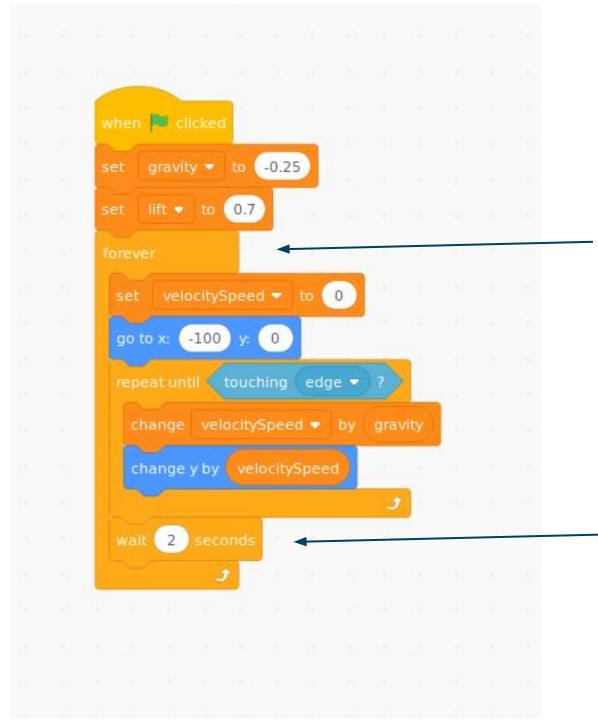
# Improve the Code

---

That's great for a single playthrough.  
Let's add a forever loop so we don't have to  
continuously press the green flag.

Add a `forever` loop after the `set lift to  
0.7` command block.

We'll also add a `wait 2 secs` block just  
before the forever block ends. This will  
provide a short delay after the character  
touches the screen's edge.



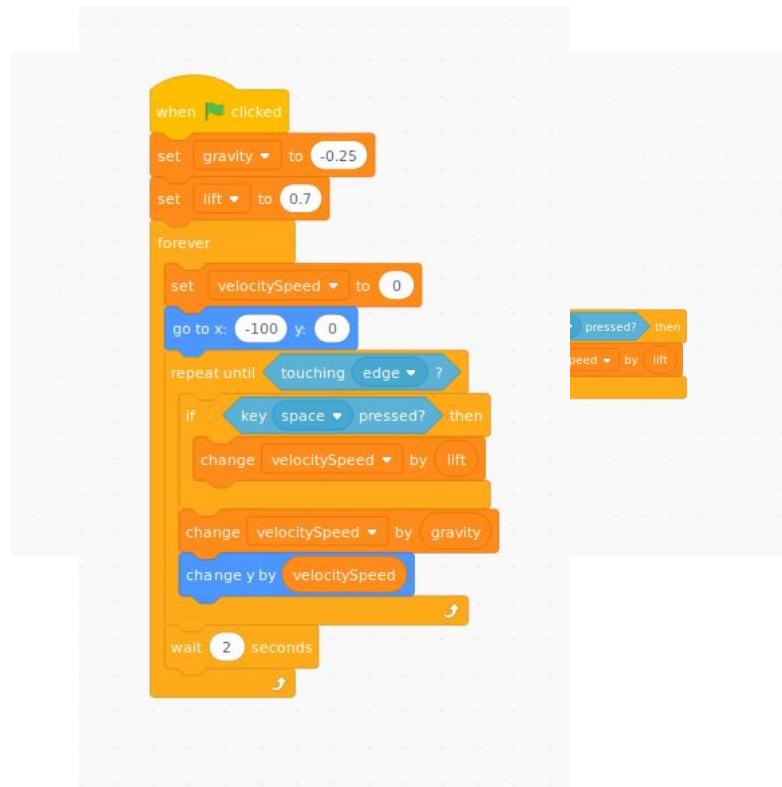
# Help Fly Gal Fly

---

This is a good start, but we want our character to do more than fall in a continuous loop.

Let's add functionality that will allow our fly gal to fly.

We'll create an if statement that will adjust the velocitySpeed when the space key is pressed.

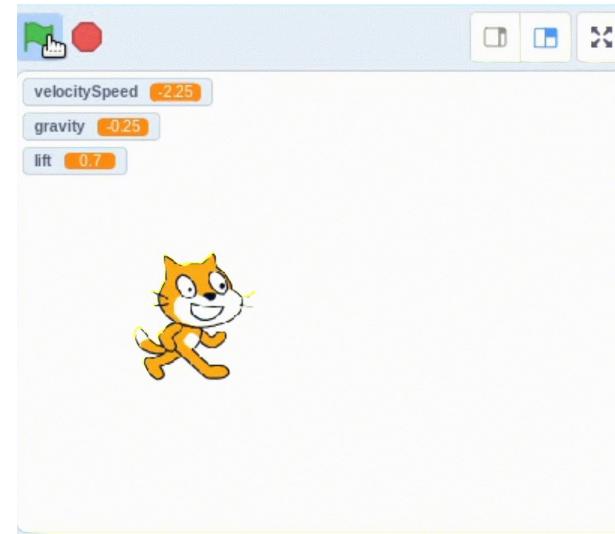


# Flying Fun

---

Now when we press the spacebar our character lifts so that we can keep her off of the edges by timing our key presses.

Once we're satisfied with the current functionality we will create a custom costume for our character.



# Create a New Character

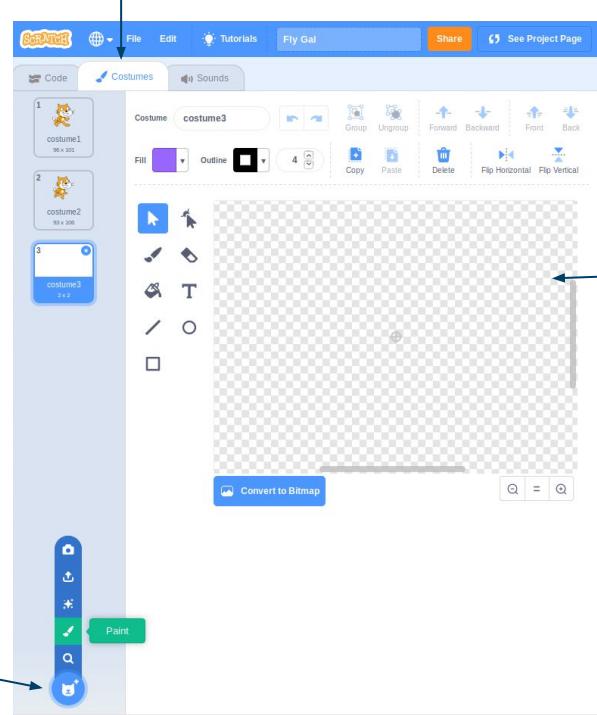
Let's create a new costume for our character! We can make our hero look anyway we want her to.

Select the Costume Menu then hover over the cat-looking icon below the costumes and select the paintbrush icon.

This will create a new costume that we can customize. Let's name our new costume "fly-girl-1".

New Costume  
Menu

Costume Menu



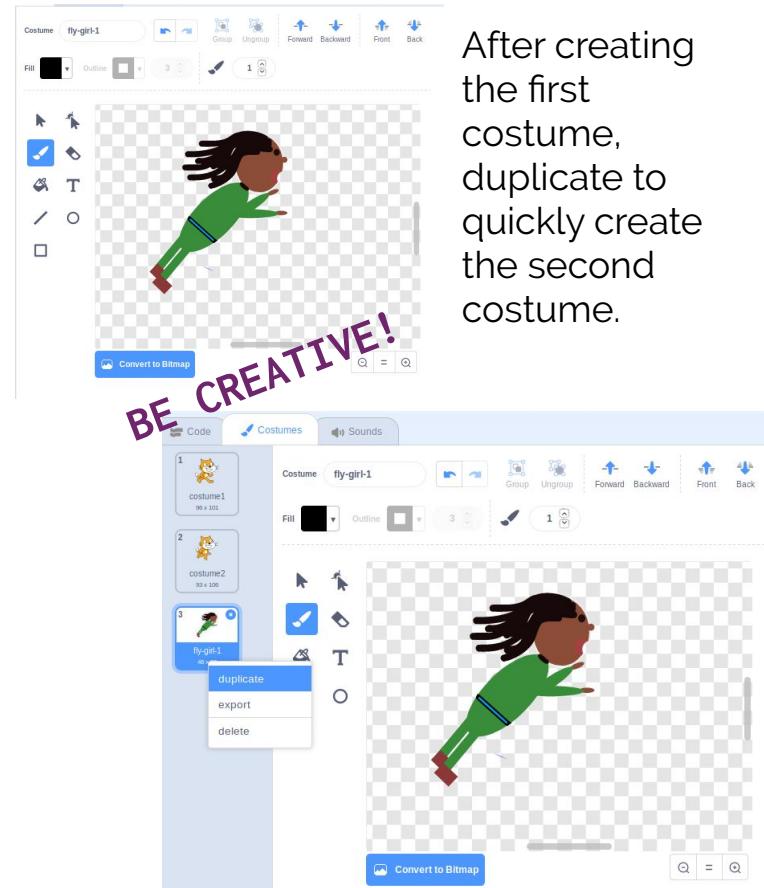
# Let's Draw!

---

We will create three costumes. One for when the girl is falling, another for when she is lifting and a third for when she hits a barrier--that's Game Over!

Experiment with the paintbrush, eraser, fill, line and shape buttons to make a character.

Avoid making the character too large. Pressing the magnifying button two times will help create a detailed character that isn't too big for our game.



After creating the first costume, duplicate to quickly create the second costume.

# Editing for Costume 2 Example

---

For the second and the shocked costumes, we do not need to reinvent the character.

Duplicate and edit to make a slightly different looking character.

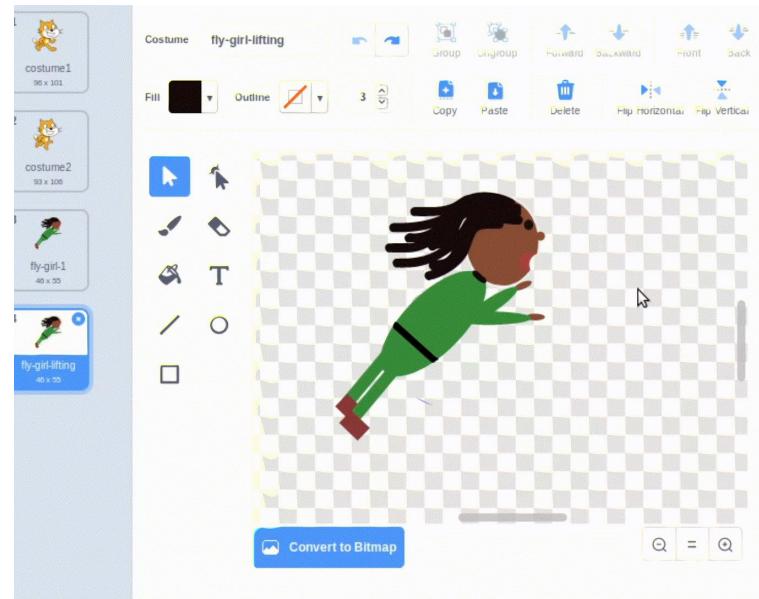
Here's an example of editing the character's hair to have a new look for lifting.

Remember to rename the costumes to:

fly-girl-1

fly-girl-lifting

shocked



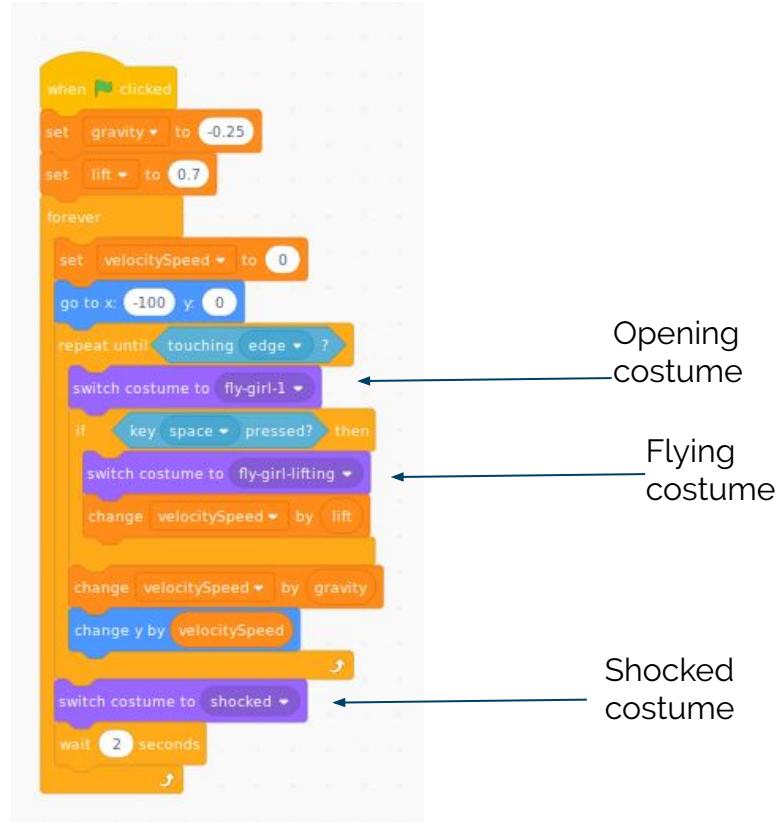
# Switching Costumes

Now that we have created our costumes, we'll add three new statements to tell the program when to use the various images.

Click on the Code menu and add the "Switch costume to 'VARIABLE'" block after the "Repeat until touching edge," "if key space pressed? Then" and before the "wait 2 seconds" blocks.

The new blocks can be found in the "Looks" category.

(The two cat costumes can be deleted, we won't be using them)



# Test Your Program

---

At this point, we should have a prototype that changes the costumes when the space key is pressed, released and when the hero touches the edges of the screen.

Let's test it out for a few seconds to be sure everything works as expected.

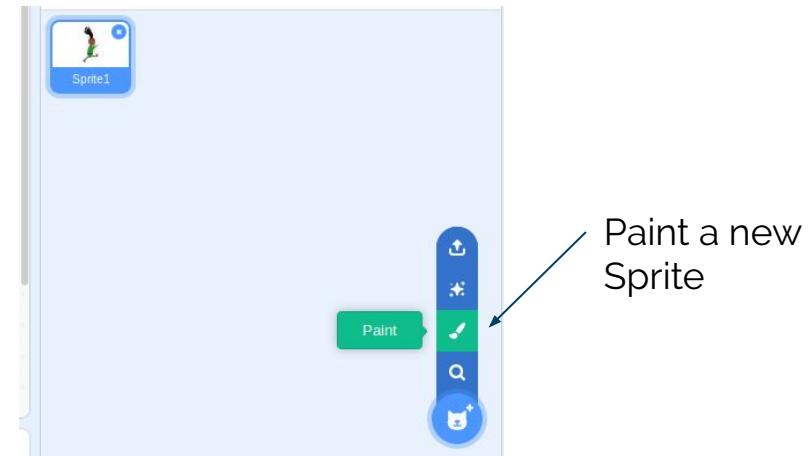
If it doesn't, that's okay. Bugs are a part of programming. Double check the variables, costume names and placement of each block.



# Add Barriers

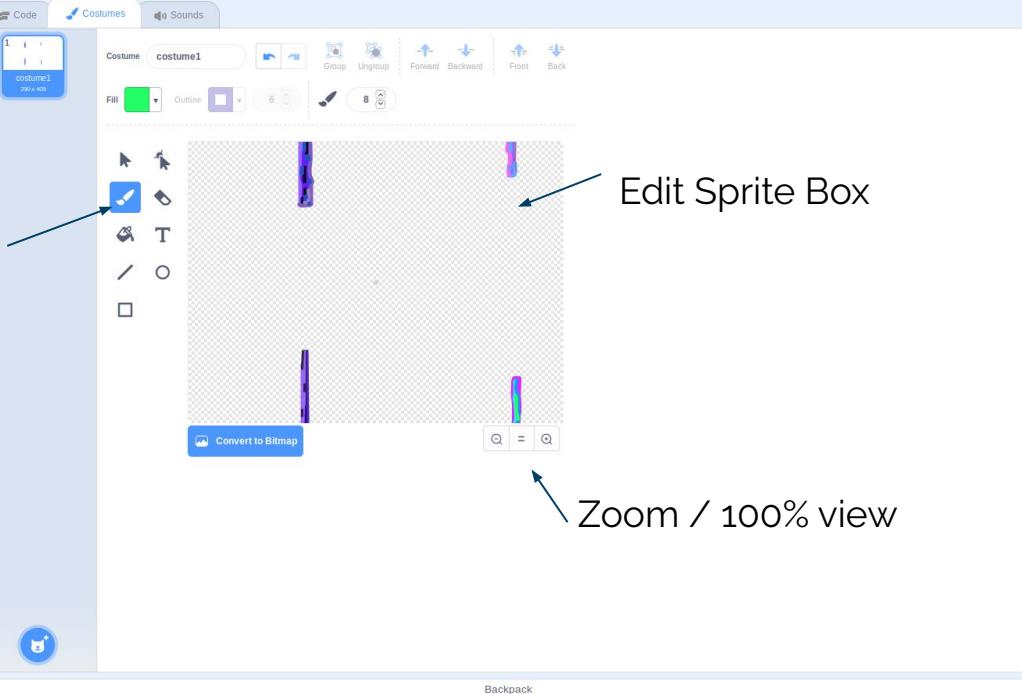
---

Let's add barriers for our Fly Gal to avoid. One set of barriers will be perfect. I drew two in a single sprite just for fun.



# Painting a New Sprite

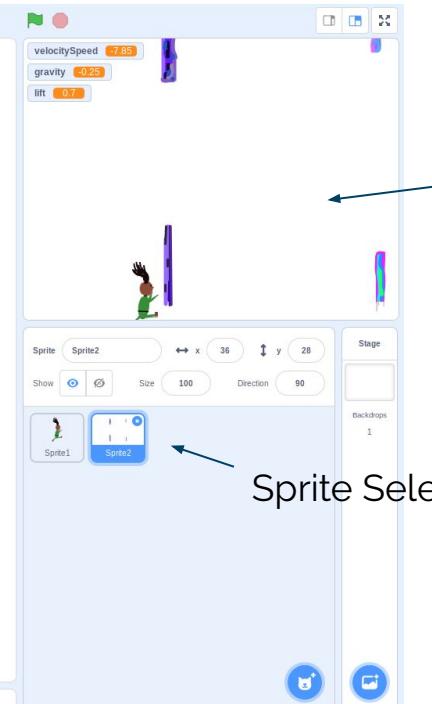
Costume Selector



Paint Tools

Edit Sprite Box

Zoom / 100% view



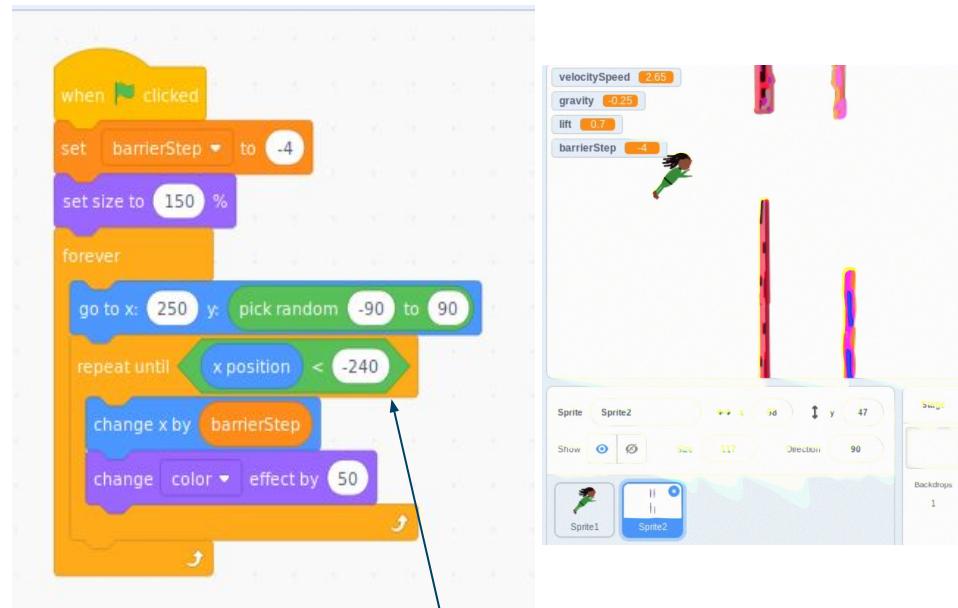
Preview Screen

Sprite Selector

# Animate the Barrier

Let's give the barrier an animation so that it moves from right to left across the screen.

Click on the 'Code' Tab for the new sprite to create a new code block.

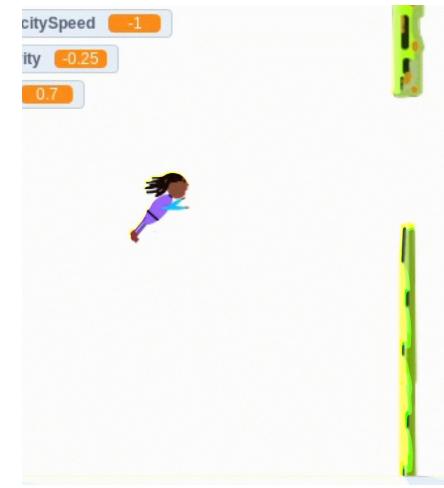
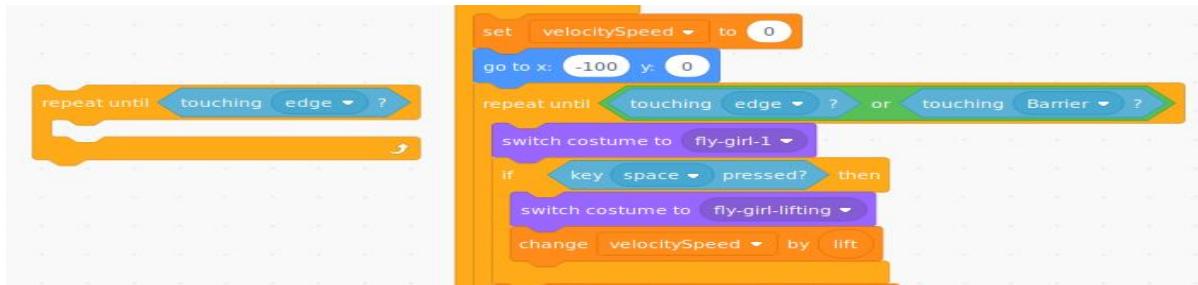


If the barrier sticks then reduce this by 10 until the barrier loops

# Collision Detector

You may notice that our Fly Gal can hit the barriers without consequence. But even our Fly Gal has limits to her powers. Let's add collision detection so the player needs to avoid the barriers.

Select Fly Gal's Code and replace "repeat until touching edge" with "repeat until touching edge OR touching Barrier"

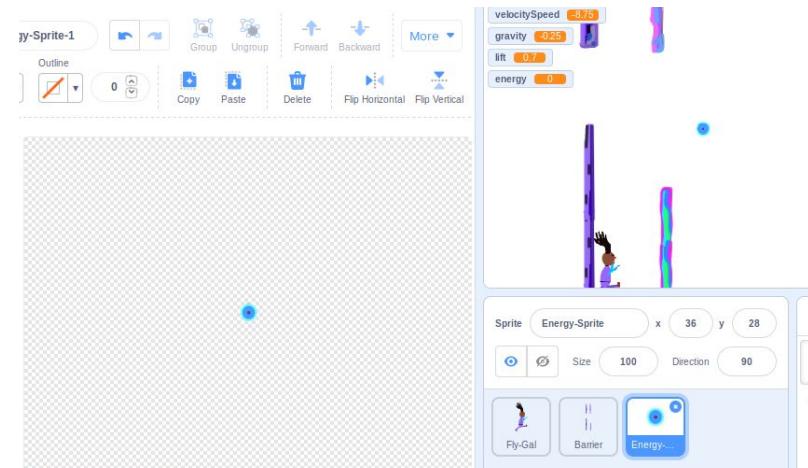


# Power up Fly Gal with Energy

This is shaping up to be a complete game.

Let's add another game element that requires Fly Gal to capture energy capsules in order to continue flying. This will add a new level of difficulty to your game.

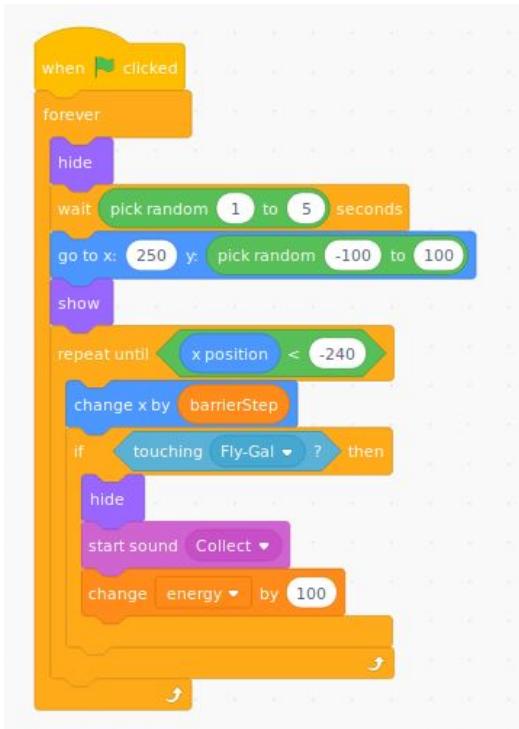
First, we'll create a new Variable called "energy" and set "energy" to equal 150 at the start of the game. Then we will create a new sprite called "Energy-Sprite" with the drawing editor.



# Power-up and Energize!

Let's add functionality that allows Fly Gal to collect the energy capsules and recharge her energy supplies.

Be sure the Energy-Sprite is selected and then click the code tab.



Hide the energy for a random time

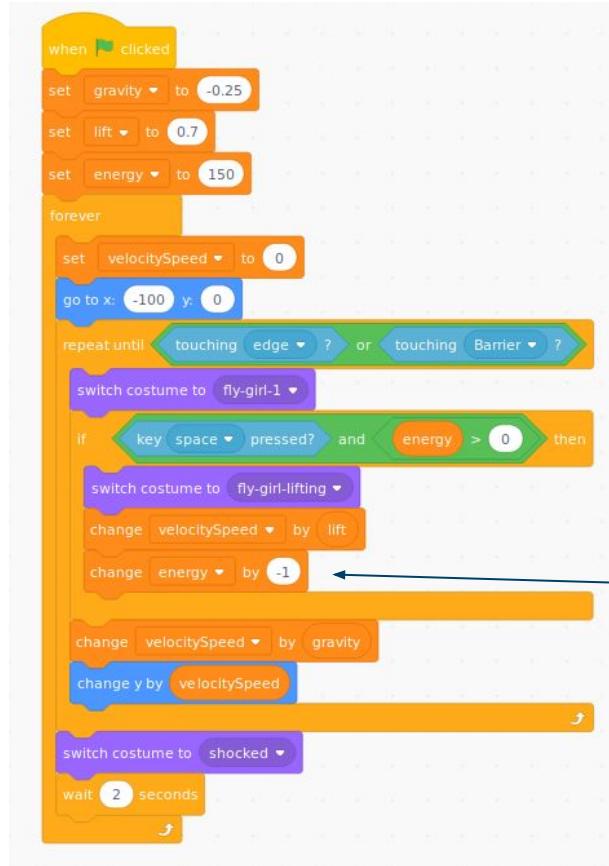
Then show it somewhere on the screen until it's either moved past the screen or Fly Gal collects it.

Make a sound when Fly Gal collects the energy and change the "energy" variable by 100.

# Burning Energy

This is great, but Fly Gal's energy only goes up, never down. That's only fun for so long. Let's refactor the code such that Fly Gal expends energy when flying and falls if she runs out.

Be sure to select the Fly Gal Sprite to refactor her codebase.



Add a conditional such that Fly Gal can fly as long as she has energy

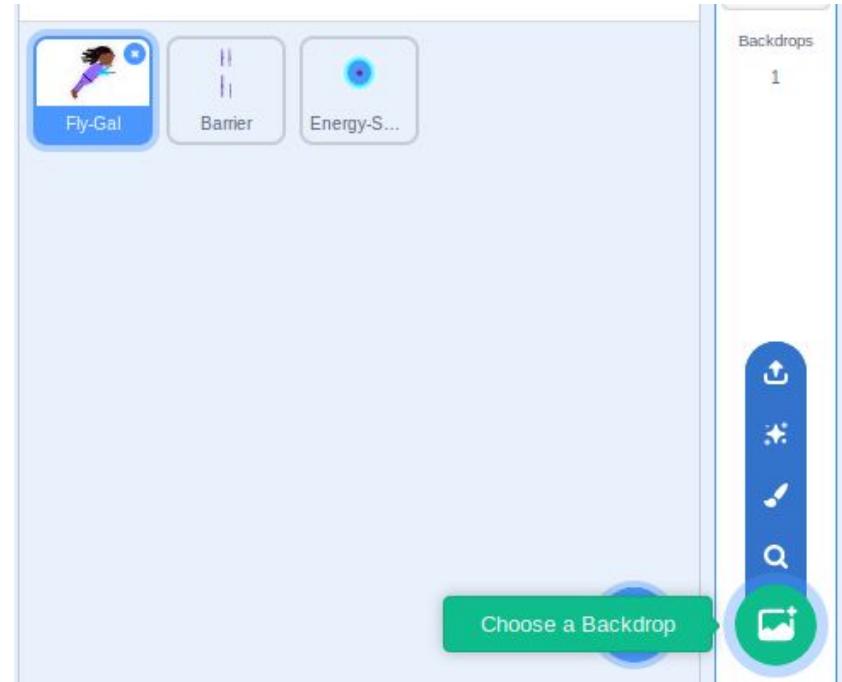
Update energy by -1 point while the space key is pressed

# Add Scenery

---

Our game mechanics should be fully operational now. Let's add some scenery so Fly Gal has more than a white background to fly against.

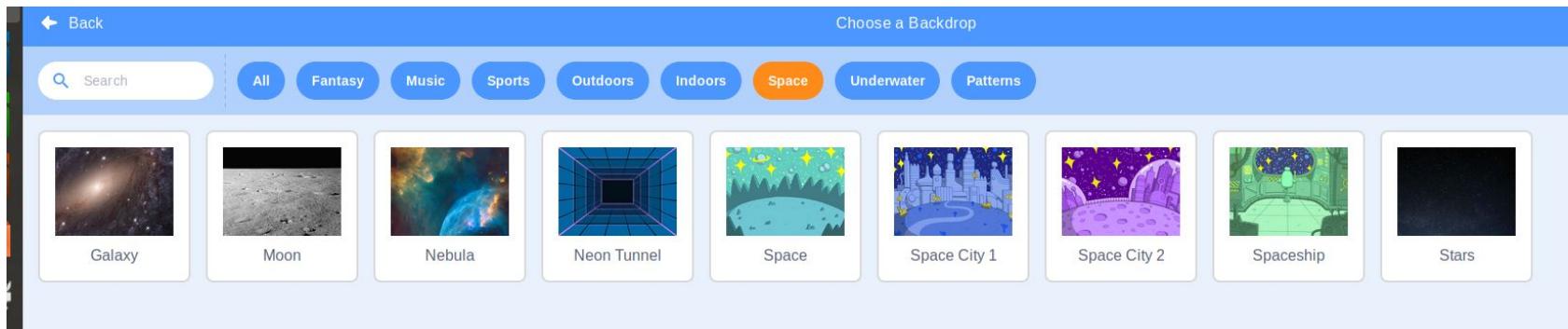
In the bottom right corner, click the "Choose a background" button.



# Choose a Backdrop

---

Now we can choose any backdrop of our choosing. Since Fly Gal is out of this world, consider a Space themed backdrop.



# Make the Backdrop Interactive

---

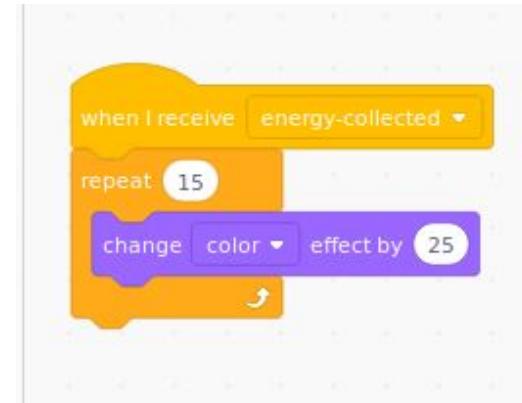
We can code the backdrop to do fun things, too! Let's add functionality such that the color changes every time an energy capsule is collected. This will require adding code to the Energy sprite and the new Backdrop.

Find the "broadcast message" block in the events category. Create a new message called "energy-collected." Broadcast a message when a capsule is collected. The backdrop will receive this message and change color effect.

Energy-Sprite Code



Backdrop Code

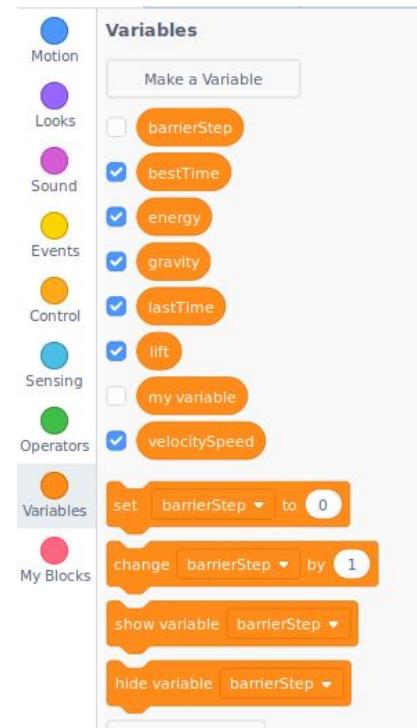


# Run Time

---

Let's add a competitive element to our game by tracking how long a player can fly before falling or hitting a barrier.

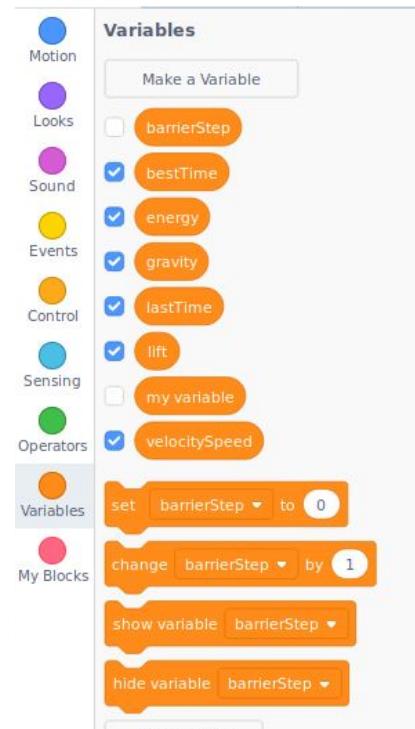
Create two new variables: "bestTime" and "lastTime"



# Clean Up

---

While we're here, we can clean up our game a little bit by unchecking the "gravity," "lift," and "velocitySpeed" variables.

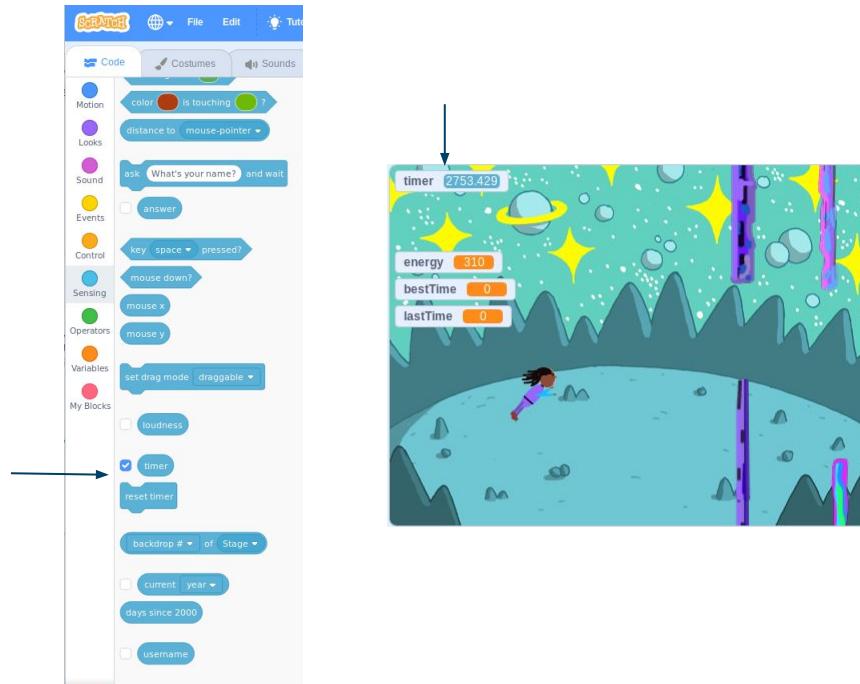


# Add a Timer View

---

Find the “Sensing” category and check the “timer” box so we can see how long Fly Gal flies in real time.

A timer box will appear in the game view.



# Organize Info Boxes

---

Did you know we can place the information boxes anywhere in our view? We can! Let's clean them up so there are two boxes in each upper corner.



# Turn Back Time

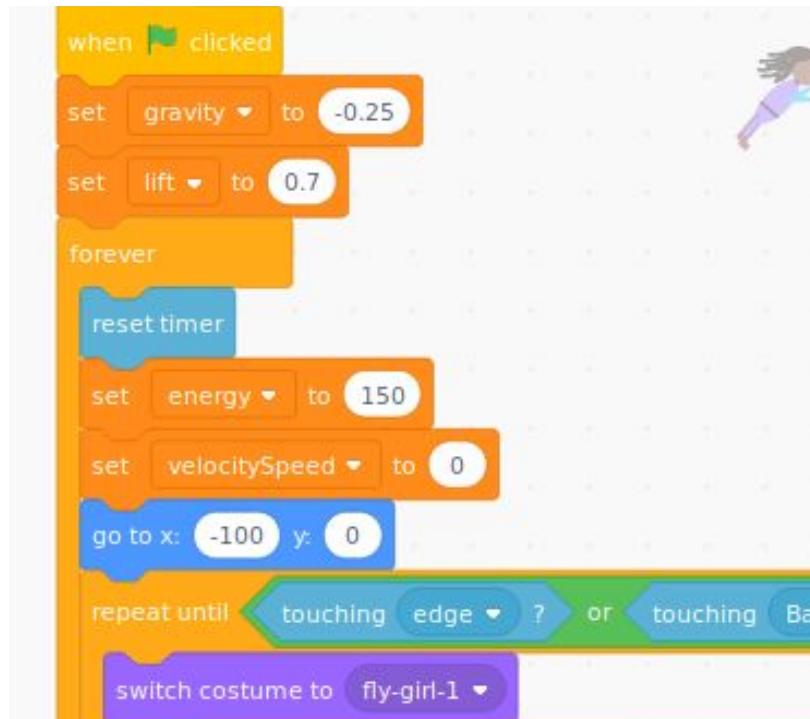
Let's add a reset timer block that will turn the clock to 0.0 at the start of each Fly Gal game. Then we'll create functionality to measure the last recorded time and the best time ever.

```
when green flag clicked
  set [gravity v] to [-0.25]
  set [lift v] to 0.7
  set [energy v] to 150
  forever
    reset timer
    set [velocitySpeed v] to 0
    go to x: -100 y: 0
    repeat until [touching edge? or touching Barrier?]
      switch costume to [fly-girl-1 v]
      if [key space v pressed? and energy > 0] then
        switch costume to [fly-girl-lifting v]
        change [velocitySpeed v] by [lift v]
        change [energy v] by [-1]
        change [velocitySpeed v] by [gravity v]
        change y by [velocitySpeed v]
      end
      switch costume to [shocked v]
      set [lastTime v] to [timer v]
      if [lastTime > bestTime] then
        set [bestTime v] to [lastTime v]
      end
      wait [2] seconds
    end
  end
```

# Fix the Bug

---

Did you notice that we have a bug? When Fly Gal touches an edge or barrier her energy doesn't reset. Let's move the "set energy to 150" block to be inside the forever loop, just below "reset timer."

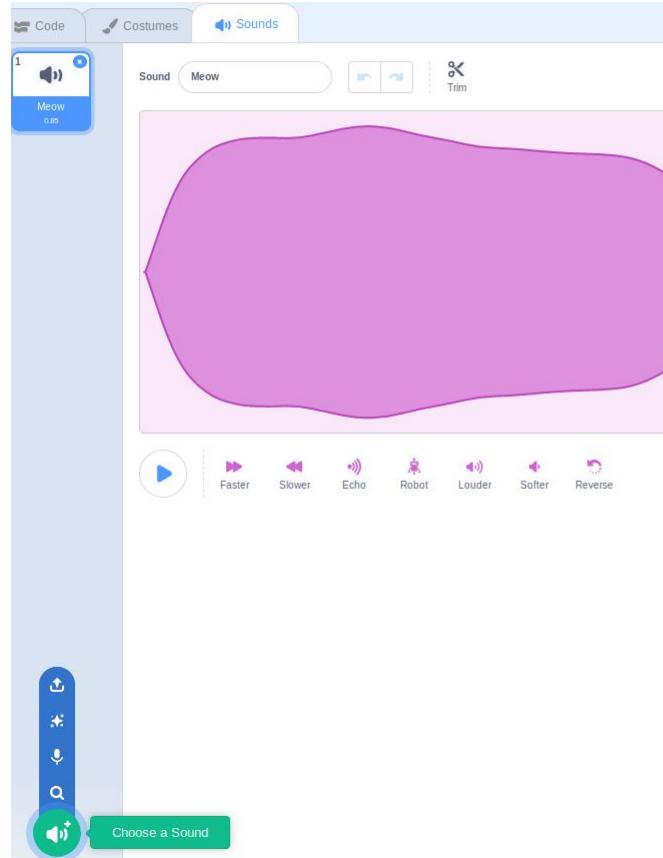


# Make Some Noise

---

Great work so far! Aside from collecting energy, our game doesn't make any noise. Let's add music that plays during the game and a sound effect for when Fly Gal touches an edge or barrier.

To add sound options select the Sounds tab and then click the Speaker icon on the lower left of the screen.



# Add Sounds to Code

---

Sounds will need to be added to different sprites. Think about what sounds each sprite should make and when for the proper placement of the new sound code blocks.

Play with the different options and have fun!



We can add a play sound block in a forever loop on the background to create game music

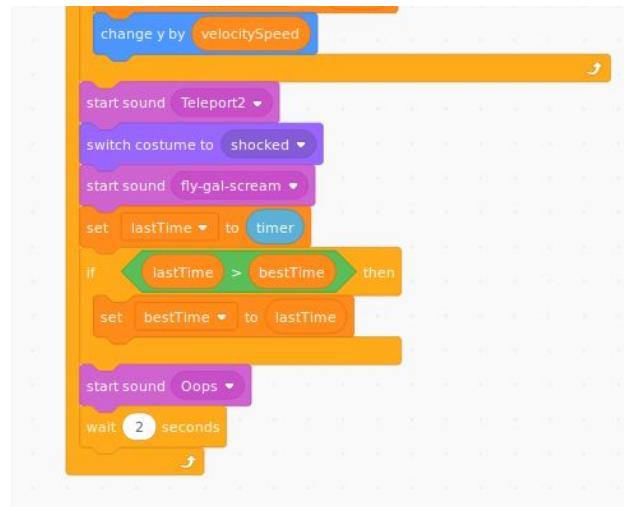


We can add sounds to the energy sprite. This adds a new level of interaction between the player and game elements.

# Experiment with Sound

---

Where else can you add sound to your game? Let's add a sound effect for when Fly Gal touches a barrier or an edge.



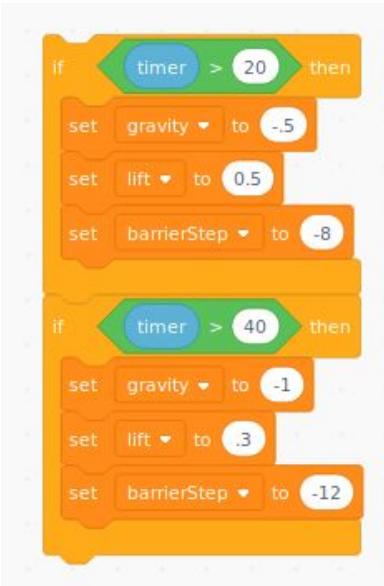
You can add one or many sounds to create a layered effect of noises to help give more meaning to an event.

# What Else Can You Do?

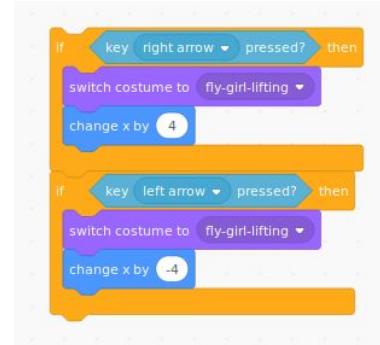
---

What if Fly Gal could also move left and right? What would that code block look like?

We could also add layers of difficulty to the game that increased with the passage of time.



Play with the variables to create a game that is fun and challenging but not too brutal to play!



# Code Your Creativity

---

Try new things. What happens when you add a new block of code to a sprite? Does it break the code? Debug and try a new approach.

Be sure to take a moment to have some fun with your game, too.

What will happen if you placed a block of code like this in the energy sprite's code space?

