Sprite Art

Description

In this lesson students will learn how to change the appearance of a sprite and add animations. Through a jigsaw activity, individual students will learn multiple ways of designing sprites that are of different levels of difficulty. After their exploration, students will turn key their learning for other students.

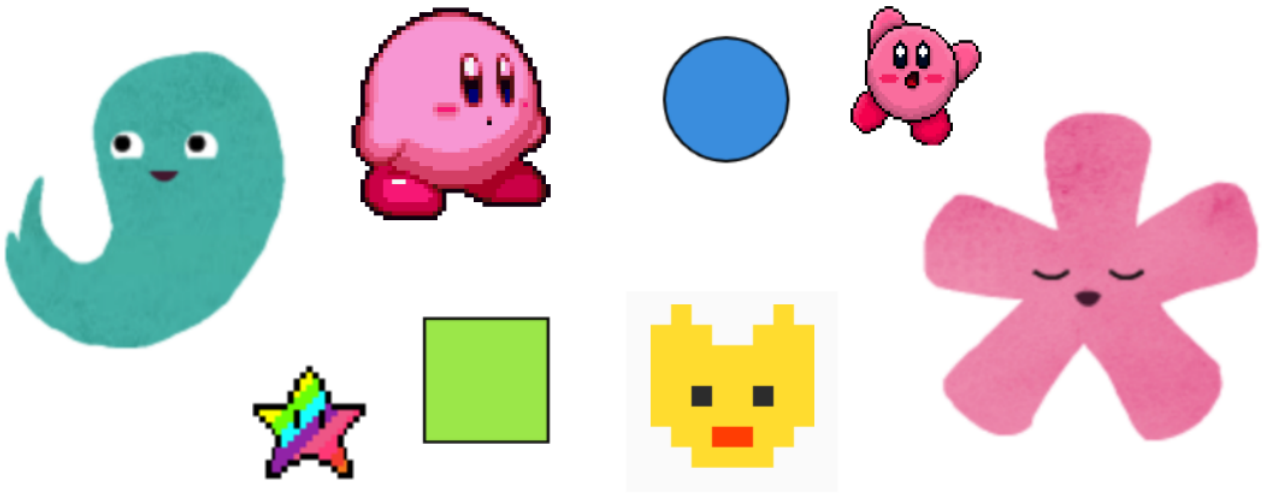
Objectives

- I can design a sprite using 'pixel art'.
- I can design a sprite using a png image.
- I can design a sprite using a sprite sheet animation.

Brain-Starter + Hook (5 min)

Prompt:
“The sprites we have made are cool, buttttttt they could be cooler.
What would you like to change about our sprites to make them more personalized?”

Turn && Talk: Which sprite would you use in a game? Explain WHY.



Lab Setup (5 min)

Either have Students identify their “Challenge Level” for the lab or assign students a challenge level. Try to encourage an equal number of all levels as this will help during the share out.

Mild - Using an image file as a sprite.
Medium - Creating a “8 Bit” Sprite.
Spicy - Sprite animation using a sprite sheet.

After students are grouped, have students move to sit with people working on the same challenge level.

Students should have a computer and sign-in to the p5 editor.

Lab Exploration (20-30 min)

Provide each group with the appropriate lab steps found below either digitally or as printouts for students to follow along with.

Mild - Using an image file as a sprite.

Starter Code | Final Code | Instructions

Medium - Creating a “8 Bit” Sprite.

Starter Code | Final Code | Instructions

Spicy - Sprite animation using a sprite sheet.

Starter Code | Final Code | Instructions

Jigsaw (10 min)

Regroup students in sets of 3, with a representative from each of the challenge levels. In these groups, students share their method.

Sentence Starters on a slide
“My method was to...”
“Here are the most important lines of code, this is how they work...”
“A ‘hang-up’ to watch out for is...”
“Do you have any questions?”

Students take turns teaching each other their method.

Debrief (10 min)

Say: “Today you saw three different methods for changing the appearance of a sprite. Turn and talk at your table about the different methods.”

What are the advantages of each method?
What are the disadvantages of each method?
Which method do you prefer?

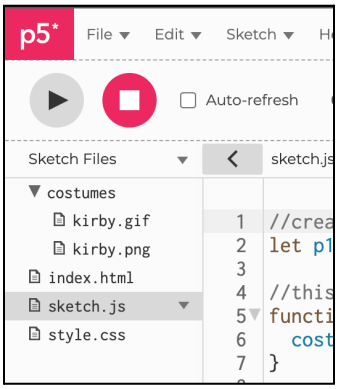
Share out some responses as a whole class.

Mild - Using an image file as a sprite.

We can use an image to change the appearance of a sprite!

Code it!

[Starter Code]



1) Look over the starter code before moving on.

Some images have already been added for you into the p5 sketch's folder.

Here, we are preloading an image to use as a "costume" for a sprite.

```
function preload() {
  costume = loadImage('./costumes/kirby.png');
}
```

The `preload()` function is a part of p5, just like `setup()` and `draw()`. Preload runs before the rest of the program, it is common to preload images and fonts when working with p5.

2) Create a new sprite with the constructor. Add the "costume" image as a parameter.

```
//sets the sprite's appearance to be an image.
p1 = new Sprite(costume);
```

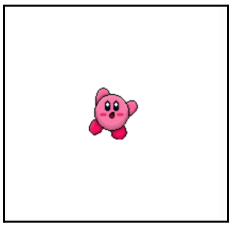
Run the code! You should see the sprite appear as a picture of Kirby!

Look in the console, you should see the size of the sprite printed out. What is it?

3) As you may have noticed, the sprite is really big! Way bigger than the canvas even.

We can resize the sprite using this command. Place it **before** you create the sprite.

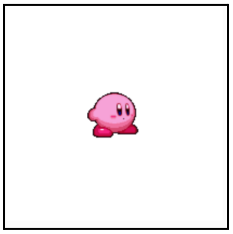
```
costume.resize(50,0)
```



Run the code! Now you should see all of Kirby.

What is p1's width?

4) Change to a gif! There is also a gif of Kirby in the sketch folder. Change the `loadImage` function to .gif instead of .png.



```
function preload() {
  costume = loadImage('./costumes/kirby.gif');
}
```

Run the Code! Now you should see a new Kirby! What is different?

What is p1's width?

Bonus!

If you still have time, try adding your own image for the sprite!

Option 1) Find an image online, it will be best if it has a transparent background.

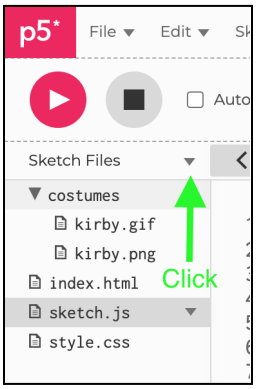
Option 2) Create your own sprite using a graphics program like Pixel Art, Pixlr, or Canva.

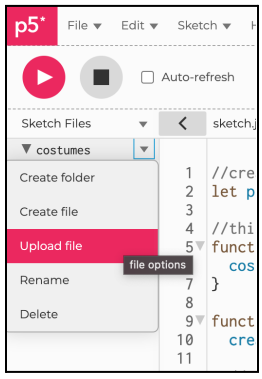
Make sure you have the image downloaded as a png, jpg, or gif.

Get it into p5!

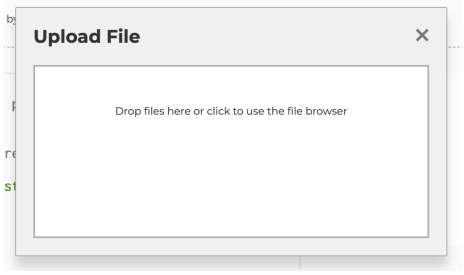
In order to use the sprite, you need to upload it to the p5 editor.

3) Open the sidebar to see the files.
4) Click the arrow next to the folder for the costumes.
5) Click “Upload File”





6) Drag a drop an image or double click to load an image saved on the computer.

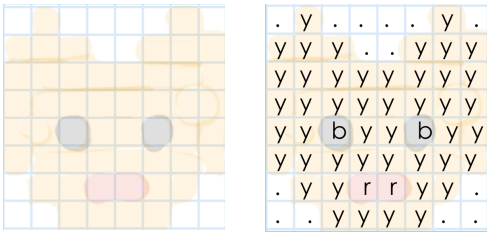


7) Make sure the filename matches the image you uploaded.

```
function preload() {
  costume = loadImage('./costumes/######.png);
}
```

8) Run it! You should now see your new image.

Medium - Creating a “8 Bit” Sprite



Design it!

1) Use the 8 by 8 grid to design a sprite.
2) Once you have your design, assign a letter to each cell.
    a) Put a . in empty cells.
    b) Put a letter to represent the color of the c, use “y” for yellow.

Code it!

[Starter Code]

1) Create a variable to hold your sprite. Name it what you like.

```
let cat;
```

2) In setup create a variable and assign it to a string of characters that matches your design.

You can use the `\n` to start a new line of the design.

```
let txt =
".y....y.\nyyy..yyy\nyyyyyyyyy\nyyyyyyyyy\nyybyybyy\nyyyyyyyy\n.yyrryy.\n..yyyy..";
```

Or you can use the ` character around the string to have each row be its own line of text.

```
  let txt = `
    .y....y.
    yyy..yyy
    yyyyyyyy
    yyyyyyyy
    yybyybyy
    yyyyyyyy
    .yyrryy.
    ..yyyy..
  `;
```

3) Next, create a variable named `img`. Assign it to the return of the `spriteArt` function.

```
let img = spriteArt(txt, 16);
```

4) Use the sprite constructor to create a new sprite. Add the img variable as a property of the constructor.

```
cat = new Sprite(img);
```

5) Run the code! You should see you design in the middle of the screen. Debug any errors that you have.

Remix and Extend

1) In the spriteArt function, change the second parameter to different numbers.

```
let img = spriteArt(txt, 8);
```

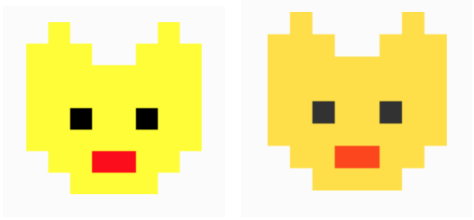
Run the code again, what changed? What do you think the number represents?

2) By default, p5 is using the standard colors, “y” became yellow, rgb(255,255,0). But we can create our own palette to use custom colors.

```
let palette = {
  'y': color(255, 225, 53),
  'b': color(50),
  'r': color(255, 69, 0),
};

//include the palette as a third parameter.
let img = spriteArt(str, 16, palette);
```

See the difference?



3) Right now the sprite is being created in the middle of the screen and doesn’t move. Using what you learned in the last lab, make the sprite start at a different location and use WASD to move the sprite around the screen.

4) Bonus! Design and code a second sprite!

Spicy - Animated Sprites

You can create sprites that are animated using multiple images. The starter code includes

Code It!

Starter Code

1) Look over the starter code before moving on.

Some images have already been added for you into the p5 sketch's folder.

The preload function is used to load all the images before the rest of the sketch runs.

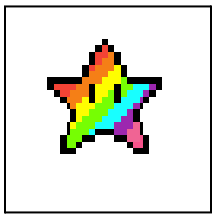
Here we are loading all the images and storing them in the `starAni` variable.

```
starAni = loadAnimation(
                      'assets/pixil-frame-0.png',
                      'assets/pixil-frame-1.png',
                      'assets/pixil-frame-2.png',
                      'assets/pixil-frame-3.png',
                      'assets/pixil-frame-4.png',
                      'assets/pixil-frame-5.png',
                      'assets/pixil-frame-6.png',
                      'assets/pixil-frame-7.png');
```

2) In setup, create a new sprite with the constructor. Add the "costume" image as a parameter.

```
//create a new sprite, add the star animation
star = createSprite(starAni);
```

Run it! You should see an animated sprite!



Remix and Extend

We have full control over the animation! There are many properties and methods that we can access. Find the full list here!

To access any of the properties or methods we must follow this structure: `spriteName.animation.property`

For example, `star.animation.frame` will return which frame the animation is on!

1) In setup, try changing the speed of the animation.

```
 star.animation.frameDelay = 10;
```

This will now update the animation once every 10 p5 frames.

How would you get the animation to change about once every second?

2) We stop the animation from playing use the `stop()` method.

```
star.animation.stop();
```

And get it to start again using the `play()` method.

```
star.animation.play();
```

Challenge!

In a previous lesson, you learned how to use keyboard inputs using the `kb` object. Your challenge is to use the keyboard to control the animation.

- [ ] If the player presses the p key, play the animation.
- [ ] If the player presses the s key, stop the animation.

Bonus!

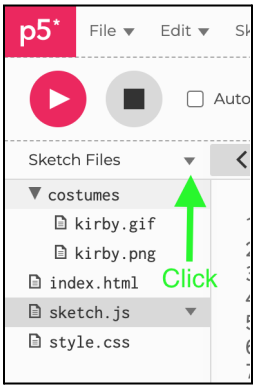
If you still have time, try adding your own animation!

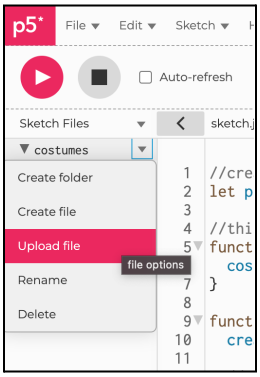
Create your own animation using Pixel Art. Export all the frames. Unzip the folder to see each individual picture.

Get it into p5!

In order to use the sprite, you need to upload it to the p5 editor.

1) Open the sidebar to see the files.
2) Click the arrow next to the folder for the costumes.
3) Click “Upload File”





4) Drag a drop an image or double click to load an image saved on the computer.
5) Change the `preload` function to have your new images.