



Girls In STEAM 2022

Activity Book

Girls in STEAM 2022

Balloon Car Design Challenge

Description: Design your own car using supplies and see how far the car can go by using force from a balloon!

What you need:

- Piece of corrugated cardboard (~5cm rectangle)
- Long balloon
- 2 pipe cleaners
- 4 Styrofoam balls
- 4 bubble straws
- 6 regular straws
- 5 wooden skewers
- 15 popsicle sticks
- Tape
- Balloon pump (For Classrooms)
- Scissors

What you do:

1. Build a car using supplies you have! Some tips include using round objects as wheels and considering what parts of a real car are vital to your balloon powered one.
2. Find a way to attach the balloon to your car. Remember! Do NOT tie your balloon if you want to have multiple races with your new car.
3. Once you have your balloon in place, release the air and see how far your new balloon powered car can go!

Example of a more detailed prototype:

1. Cut one jumbo straw in half, these will be your front and back axels.
2. Place the skewers through the cut halves of each straw.
3. Pierce a foam ball at the end of each skewer to act as wheels. (The foam balls should also stop the skewer from falling out of the straw.)

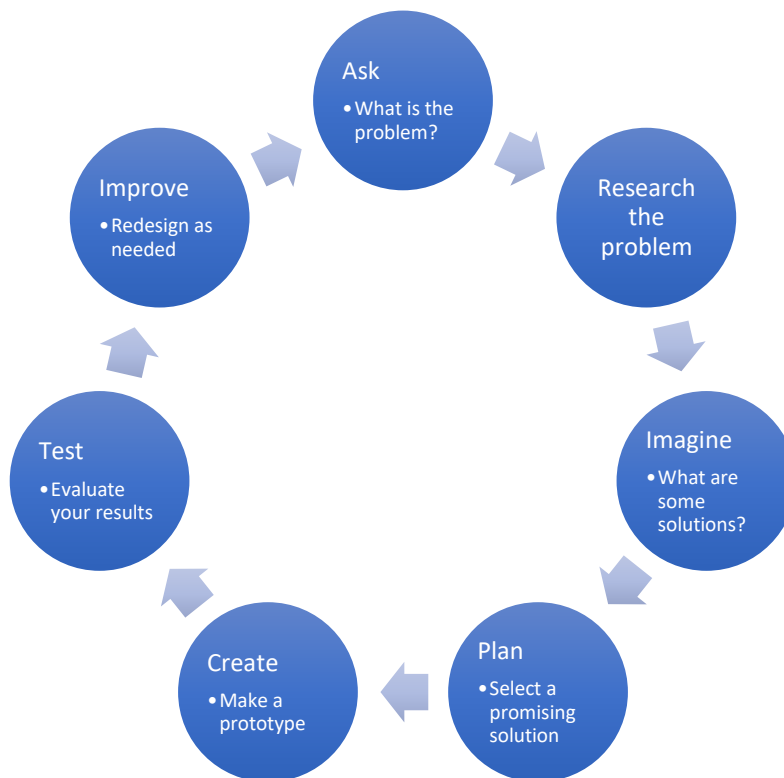
4. Tape one axle and wheel to the slim end of the cardboard and do the same with the opposite side. It should have a basic resemblance to a car.
5. Flip over the cardboard to have your axels and wheels touching the ground.
6. Find a way to attach the balloon to the top of your car. Do you tape the balloon while it is full? Or do you build something else that can hold the balloon in place? Remember! Do NOT tie your balloon if you want to have multiple races with your new car.
7. Once you have your balloon in place, release the air and see how far your new balloon powered car can go!
8. You can use the remaining supplies in your kit and experiment new ways to build your car and test different ways to create force.

Explanation:

These balloons powered cars are powered by the thrust created by air escaping the balloon. Think of this almost like your car's engine. Newton's third law of motion states that for every action there is an equal and opposite reaction. As air leaves the balloon it goes in one direction which causes the car to move forward in an opposite direction!

Engineers utilize scientific principles to develop useful technologies and solutions to problems that people experience in their lives. Prototyping and collaboration are tools that help engineers develop these solutions.

Here are some steps that engineers take to create effective solutions:



Taking It Further:

- Try to find new ways to use force instead of balloons or using two balloons
- Use other round objects to act as wheels
- Experiment your cars force on different terrains
- Challenge your friends to race your cars
- Challenge yourself to make your car go in straight line or even in circles
- Can you make a balloon car that travels for as long as possible (Time-wise!)

Balloon Car Design Challenge

Tips and tools:

- Use this empty space to draw out your designs.
- Majority of cars have a few basic components, such as, wheels, axels and a body.
- Predict how far the force will make the car travel.

Girls in STEAM 2022

Owl Pellet Dissection

Description: Take part in a real dissection using the tools provided to examine the bones, fur, and structure of an owl pellet.

What you will need:

- Owl pellet
- Gloves
- Forceps
- Toothpicks
- Paper plates
- Bone ID sheet

What you do:

1. Your owl pellet will be wrapped in foil. Unwrap it and place on the paper plate.
2. Observe the outside to see if you see any bones sticking out.
3. Use your forceps and toothpicks to slowly scrape away the excess material to reveal bones.
4. Use the bone ID sheet to help understand what bones you have. These owl pellets are from Barn Owls in Canada, the United States, and Northern Mexico. They may have some varying contents!
5. Once complete you can keep any of the bones you find and everything else can be thrown away
6. Wash hands after dissection.

Exploring the science:

Owl pellets are masses of bone, teeth, hair, feathers and exoskeletons of various animals preyed upon by raptors, (birds of prey), like our Great Horned Owl. Pellets are also made and regurgitated by hawks, eagles, and other raptors that swallow their prey whole or in small pieces. Owls feed early in the evening and regurgitate a single pellet approximately 20 hours after eating. Unlike snakes, the protein enzymes and strong acids which are found in the digestive tract of raptors do not digest the entire meal. After swallowing their food, it passes into the gizzard which grinds up their food in place of them having to chew. Next, it passes into the stomach. Animal tissues that are broken down by the acid in the stomach continue into the intestine for further digestion. The rest of the food that the owl can't digest is formed by the relatively weak stomach muscles into slimy, wet pellets made up of the undigested fur, bones,

feather etc. In this process even the most fragile bones are usually preserved unbroken. These leftovers are what we're going to examine!

The pellets have been sterilized, but could still contain bacteria from the owl's gut. In the wild, owls feed on many smaller animals such as birds, frogs, mice, moles, and squirrels. Larger owls, such as the Great Horned Owl, are also known to prey upon larger animals such as raccoons, rabbits, and even skunks. Regurgitated owl pellets are an important part of the ecosystem out in the wild. Many insects (especially larvae) and fungi will use the pellets as food and shelter. They also give scientists many hints about what, and where, owls eat.

Girls in STEAM 2022

Design Exercise

Activity Description: Design a new library for your school!

What you will need:

- Paper
- Pencil
- Eraser

Summary of Activity:

By using visualization exercises, participants will learn how to take their design ideas and develop it further!

Introduction Message:

Every design starts with a vision, so to begin the design of your new library, we are going to do a visualization exercise. Let's get started! First, we need to get into the right head space, so take your handout and a pen, find a comfortable place to sit or even lay down and when you are ready, close your eyes .

Make sure you are comfortable and take a few deep breaths.

We are about to visit the new school library that you spent the last year designing. It is rumoured to be an incredible space full of all of the things the most important clients, the students, asked for in the new design. It has things people have never seen before in a library because it's so much more than a library, it's the new hub of the school, the place where students want to be more than anywhere else.

The transformation is incredible and everyone is very excited! You have just arrived – have a slow look around, what do you see?

What to do:

- 1) Start the visualization exercise (5 senses): Start to think of what the new library for your school might look like. Utilize the Visualization Handout to help guide your thinking!

- 2) Write down anything else you want to make sure you remember about your design
- 3) Now that you have an idea of what your library will look like, draw out the design for your library!

Taking it Further:

- a. Measure your existing library, or the space in your school that you decide to renovate and create a to-scale “blueprint” for the design.
- b. Create a vision board for the library finding images online or in magazines to describe the quality of the space the textures, colours and patterns you might find there.
- c. You can download Google SketchUp (with permission of an adult or guardian) and experiment with real design software. Or, you can use the browser based (no download required) design software TinkerCAD, there are many online tutorials available to get you started.

Visualization Handout:

1. What was the first thing you saw when you visualized the space: How did the space make you feel?
2. Describe the colours and textures you saw in the space: What was the floor made of?
3. What were the walls made of?
4. Describe the light in the space?
5. What were the windows like?
6. What did it sound like?
7. What did it smell like?
8. What furniture was in the space?
9. What was the coolest thing you saw?
10. How does the space make you feel? Energetic? Calm? Focused?
11. What do you imagine people doing in that space?
12. What are people saying about the space?

Girls in STEAM 2022

Ocean Swim Coding

Description: Create your own 8-bit game and share it with a PyBadge.

What you will need:

- Laptop or desktop computer (tablets are not recommended for this activity)
- Internet Browser (Google Chrome recommended)
- Website access to Arcade MakeCode
 - <https://arcade.MakeCode.com/>
- 1x Adafruit PyBadge LC Board
 - Don't have a PyBadge? It's okay! Arcade MakeCode has a virtual testing tool that will allow you to still code and test your program!
- 1x USB Cable

Summary of Activity

Learn how to code an 8-bit game! In this game, coders will learn how to create sprites (or game characters) and how to make the sprite move. The object of the game is to get your fish sprite to make it through seaweed towers and to avoid toxic water bubbles. If your fish sprite touches a toxic water bubble, you lose one life and will be transported to a random location within your map. Try to get your fish to the end of the level to win.



**SASKATCHEWAN
SCIENCE CENTRE**

Real science. Real fun!

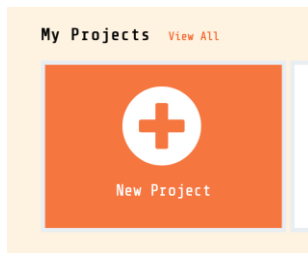


What to do:

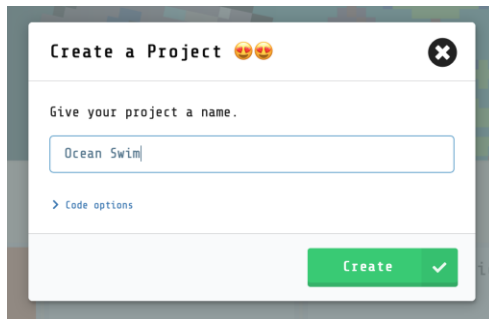
1. Go to <https://arcade.MakeCode.com/>

Note: You do not need to make an account to create games or download projects to your computer or PyBadge. Accounts are great if you want to share your work with the online community. Be sure to get permission from a parent or guardian before signing up.

2. Click “Create New Project”

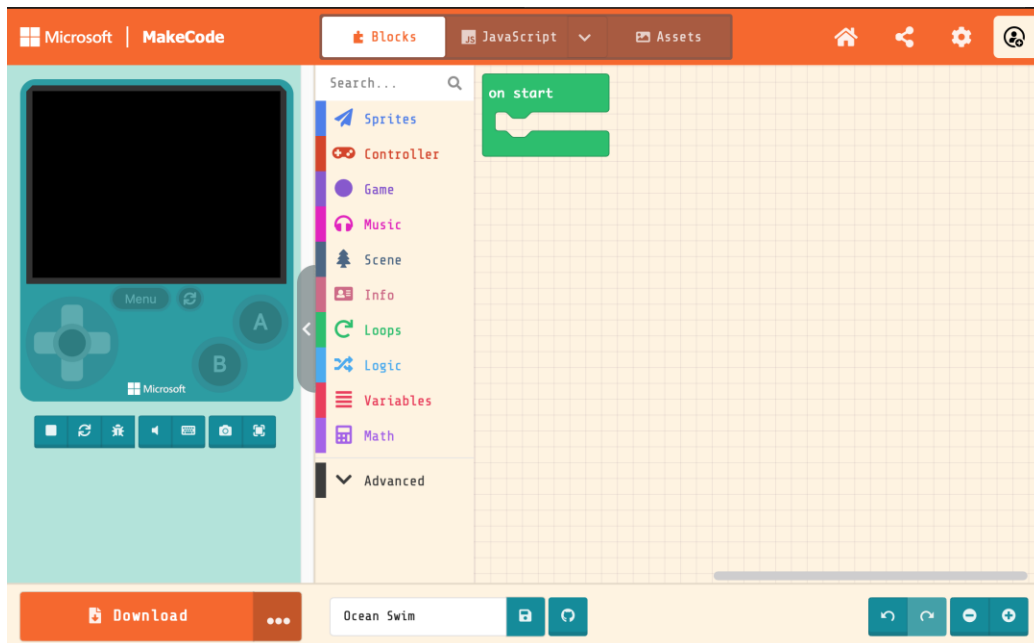


3. Name your project. We are calling ours “Ocean Swim.”



4. Connect your PyBadge to Arcade MakeCode by first plugging in the PyBadge to your computer. Next, go to the orange “download” button on the bottom left of your webpage and click the “...” button. Select choose hardware and select the “Adafruit PyBadge” which will be the second row on the left. Next click “connect device” option from the “...” button. Click the “Connect device” button again choose the device out of the popup window. If you are having troubles, look at the troubleshooting section later in the document.

5. To make your game, you will place code from the “Toolbox” menu onto the canvas. To begin your canvas will include a green “on start” block. To add code, click the category you are looking for, and just drag-and-drop the code onto the canvas. To make your code work properly, code “blocks” must link together – sort of like digital LEGO blocks. You will know they are connected properly when they have full colour. If the colour is dull, they are not connected properly. As you add code to your canvas, the mini game box on the left of the screen will change. Since the code isn’t completed yet, your mini game will not function properly at this point.



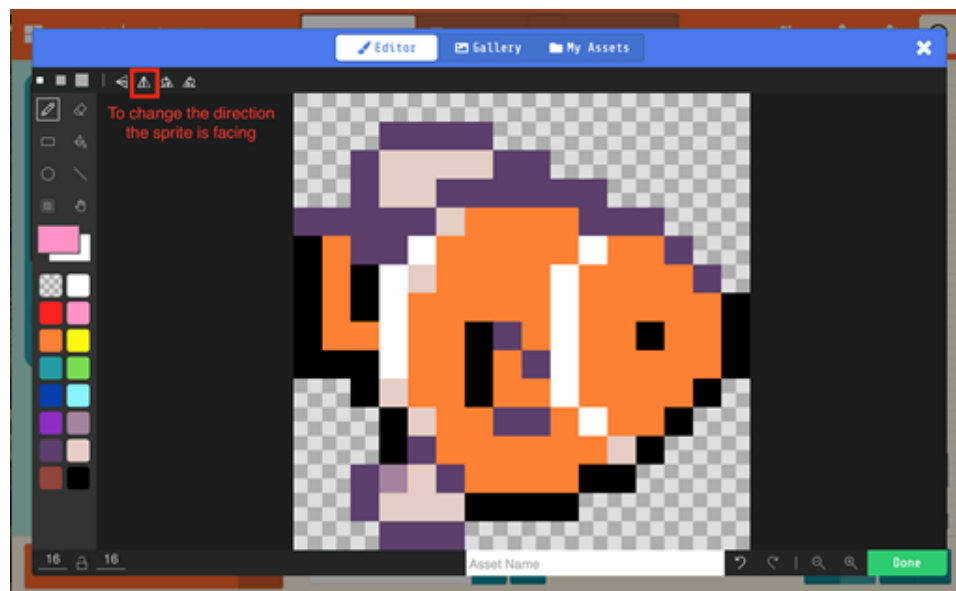
6. Click the blue ‘sprites’ toolbox tab. Drag and drop the “Set mySprite to sprite [] of kind Player” block inside of the green “on start” block. Double click the grey box to select your sprite.



7. For this example, we selected “Gallery” at the top and scrolled down to select the orange fish. We then reflected the image so it was facing the right side of the screen. You can choose any sprite you like.



If you have chosen another sprite and need to have it face a different direction, use the mirroring options highlighted in the image below to flip the direction it is facing.



Once you are happy with your sprite, click the “x” in the top right of the editing window.

8. Your sprite should now show up on the left game box. For the rest of the instructions we will refer to this as our fish, but it will be whatever sprite you chose. To get your sprite to move, you will need to go under the red “controller” toolbox tab. Select the “move mySprite with buttons +” block and drag it underneath of the “set mySprite to sprite of kind Player” block.

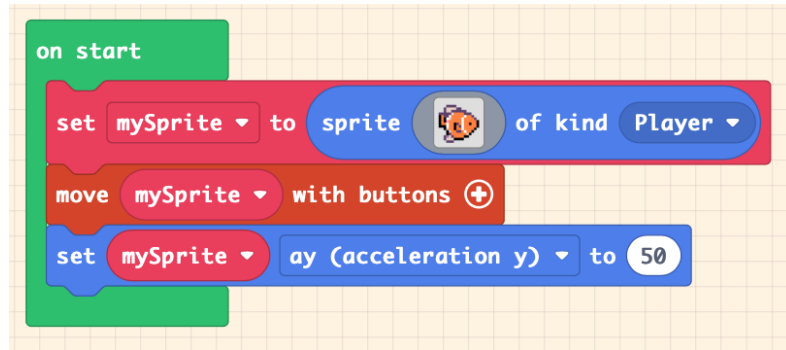
This line of code will let your sprite move with the arrow keys as well as the “W,A,S,D” keys. You can test this now.



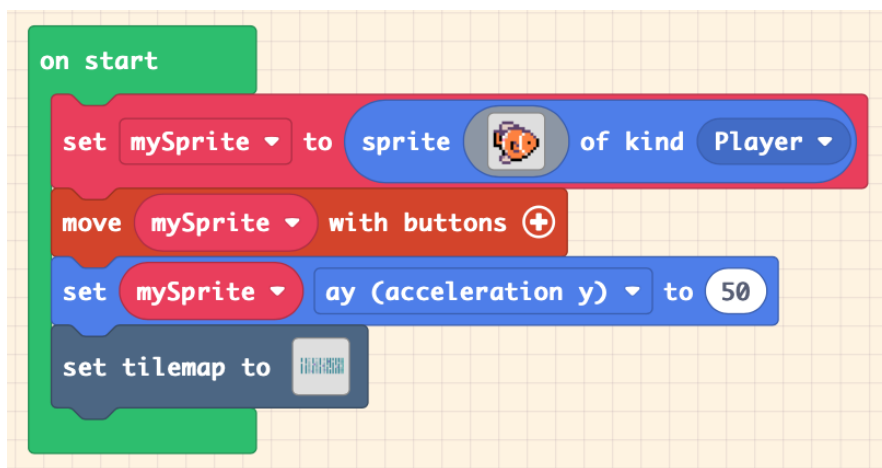
9. Our fish sprite can now move up, down, left and right. To get our fish sprite to look more like they are swimming in water, we need to change two things:
 - a. Click the “Plus” button on the “move mySprite with buttons” block. In the text box beside “vy” change it to say “0.”
 - b. Next, we will add a block to change the way our sprite accelerates horizontally, or on on the y-axis (up and down). Choose the “set mySprite x to 0” block from the “sprites” tab on the toolbox tab. And put it below your current blocks.

Then, use the drop-down menu that currently says “x” and select “ay (acceleration y).” We need to change the movement speed to give it a burst of motion on the y-axis, to appear like it is swimming, when we hit the left or right, or A or D keys. 50 is a good place to start.

To make the movements faster or slower, try changing the number to a speed that you prefer to have your sprite swim at. This can change how easy or difficult your game is.

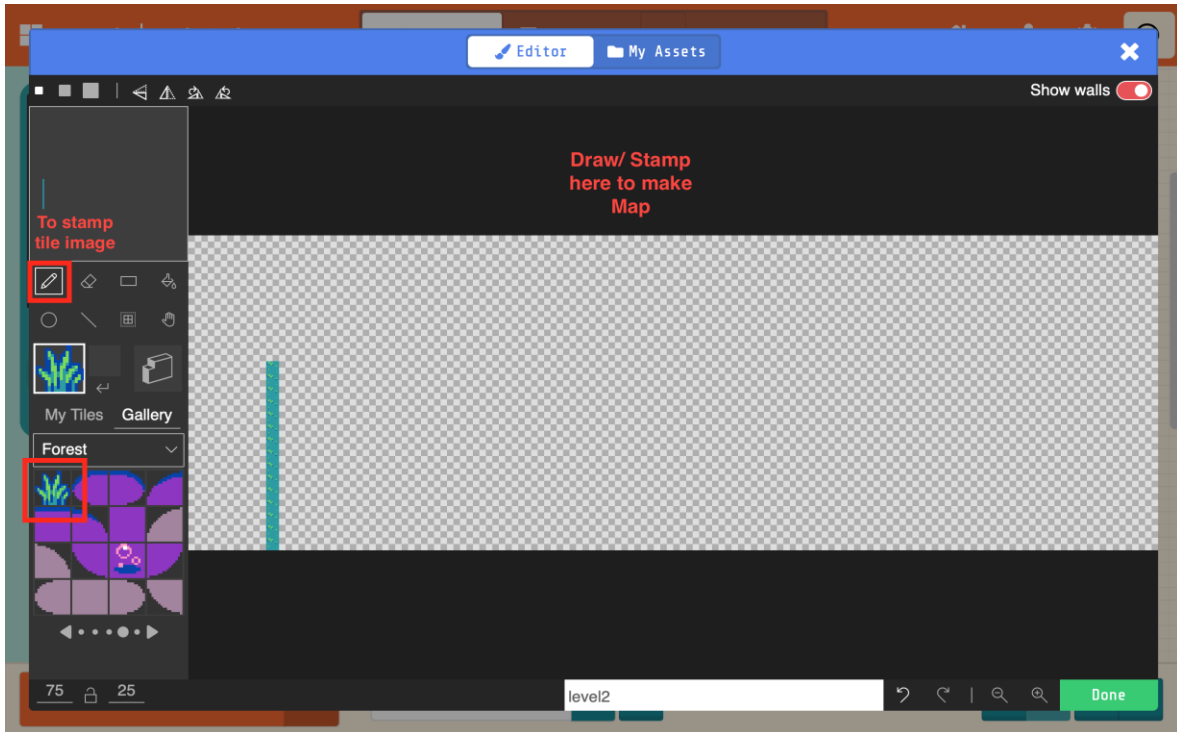


10. Next, we need to create the level tile map, or the background and obstacles our fish sprite will be traveling through. Select the “set tilemap to []” block from the dark blue “scene” option from the toolbox menu.

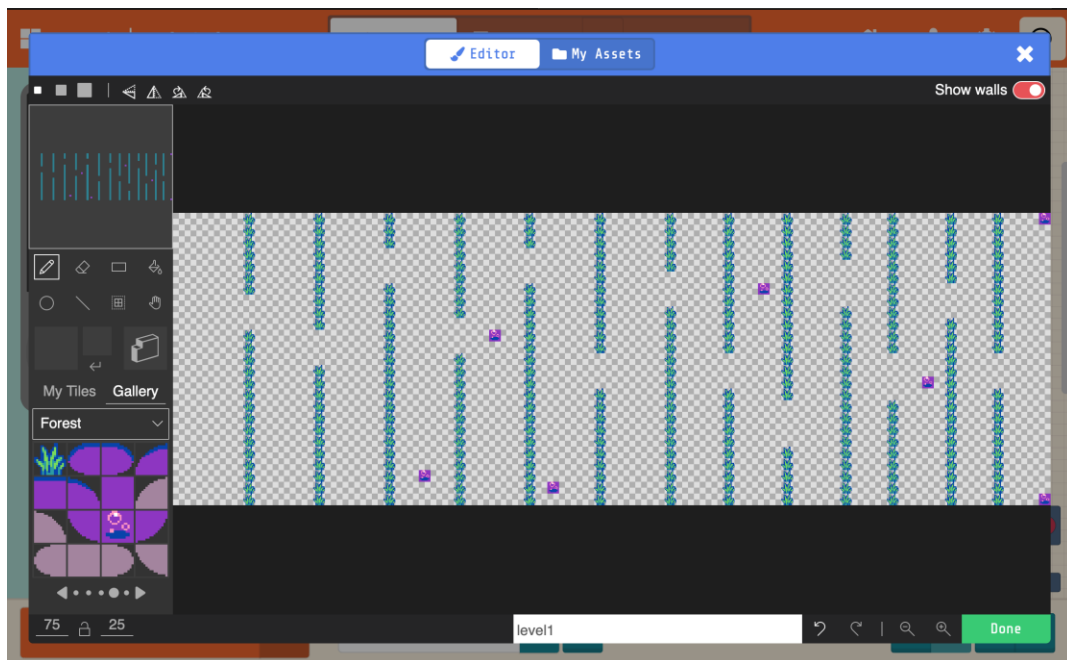


When creating your tile map, make sure to set the size of your map to 75 x 25 to give a lot of swimming and obstacle room. You can choose different terrain tiles in the bottom left corner. There are many options you can choose from. For this example, we will be using the seaweed tile.

When creating your seaweed towers and obstacles, you want to make sure to have enough physical space for your fish sprite to comfortably move. Sprites are typically the same size as one “tile.”



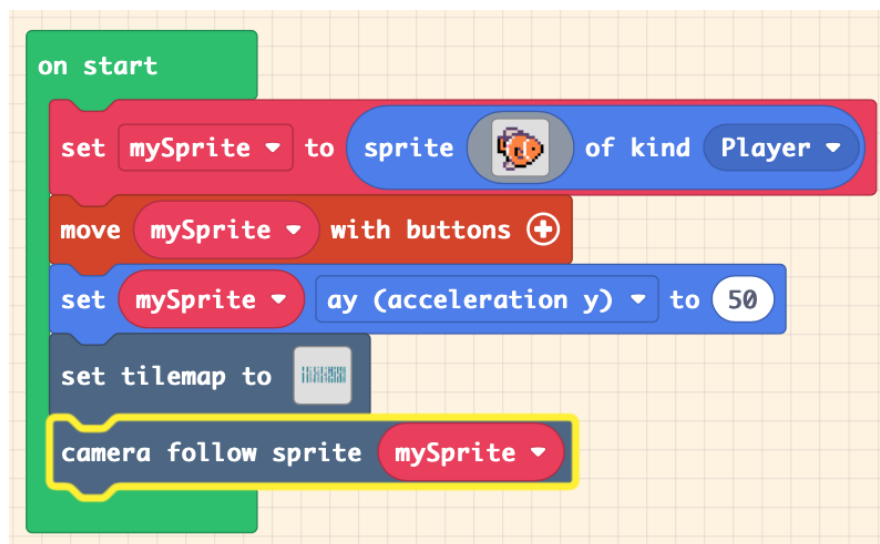
After adding in your seaweed towers, add in a different tile type of tile, like the purple bubbles to use as a “hit space,” or a space that affects your sprite if it touches it, otherwise known as an “overlap.” Place the new tile randomly, more random tiles can make your game more difficult.



11. You will notice that your fish sprite swims right through the seaweed tower. To turn tile maps solid for sprites, we need to draw walls. To draw walls, make sure your pencil tool is selected as well as the “draw wall” tool. Stamp/drag the wall tool over the seaweed tiles until they are all red. Now your fish sprite has its obstacles.



12. If you have tested your tilemap out, you may have noticed that your fish sprite disappears. To get the camera to follow the sprite, we must go to the “Scene” tab in the toolbox menu and grab the line of code called “camera follow sprite mySprite”

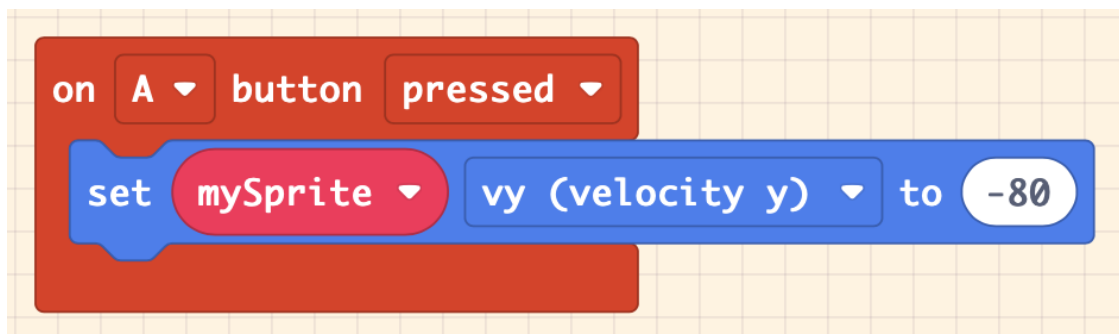


13. Next, we want to change the velocity of the sprite, as this will enhance the motion of our fish sprite to make the swimming look more natural.

To do this, drag the “on A button pressed” block out from the “controller” section of the toolbox. This goes separately from your “on start” blocks as it only occurs when you give it information or press a button.

Then, you will get the “set mySprite x to 0” block from “Sprites.” Use the dropdown menu to change “x” to “vy (velocity y)” and change the textbox to say “-80.” You will notice this change of velocity in your fish sprite by pressing the A button, or space key on your keyboard. By using the number -80, we see that the fish sprite moves up over quickly. Changing the ‘vy velocity’ number will change how fast your fish sprite can sprint.

Test different numbers - this is a good visual representation of Acceleration and Velocity!



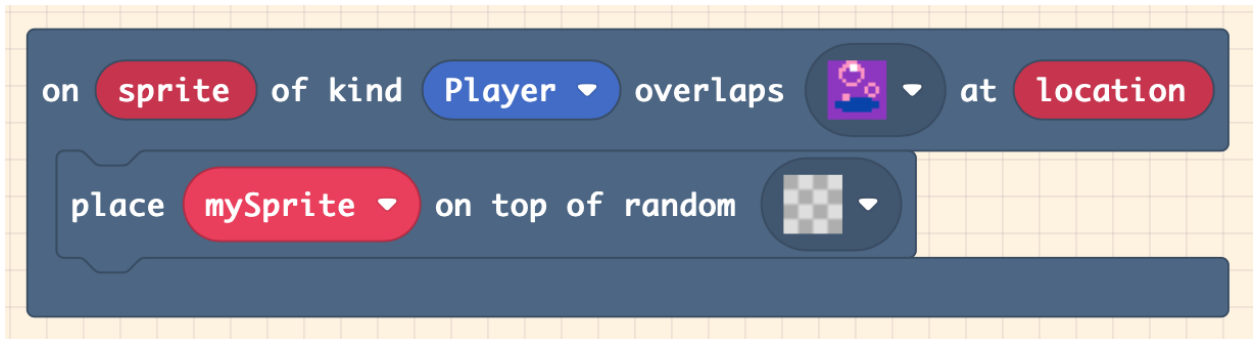
14. To make the game more interesting, we are going to add a new element to our game. We are going to use the second tile map we added earlier to create an obstacle for our fish sprite. If our fish sprite comes into contact with the obstacle, we want something to happen.

In this activity, we are going to make the fish sprite lose a LIFE if it touches the obstacle. To do this, we are going to need to code something called an “overlap”, this means our sprite will touch or overlap with the second map tile. Grab the “on sprite of kind Player overlaps [] at location from the “Scene” category.



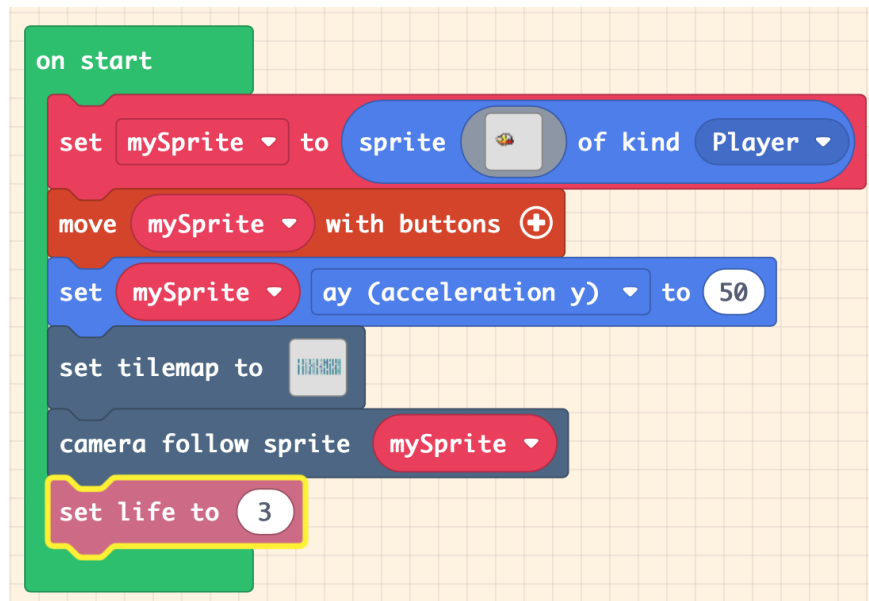
In this example, we have used purple bubbles as our ‘hit box’ overlap tile. So, any time our fish sprite touches this specific tile, we lose a life and get teleported to a different location on our current map location. This means we will teleport in the map area where our fish sprite is

currently in, placed away from the purple bubble tile. To get this block of code, go to the dark blue “scene” tab and select “place mySprite on top of random [grey squares]”



Make sure that the random square drop down is selected to the empty tile space (grey squares) as selecting a seaweed or bubble tile will cause a glitch in your game.

15. Now that we know that something will happen when our fish sprite swims into the purple bubbles tile, we need to tell the rest of the game what is going to happen. Our fish sprite will lose a life every time it happens. But before we can do that, we need to add lives to our game. To do that, we are going back to our green “on start” block. On the toolbox tab, go to the dark pink option ‘info’ and select “set life to 3”, this will automatically update your game to have 3 lives every time it starts.

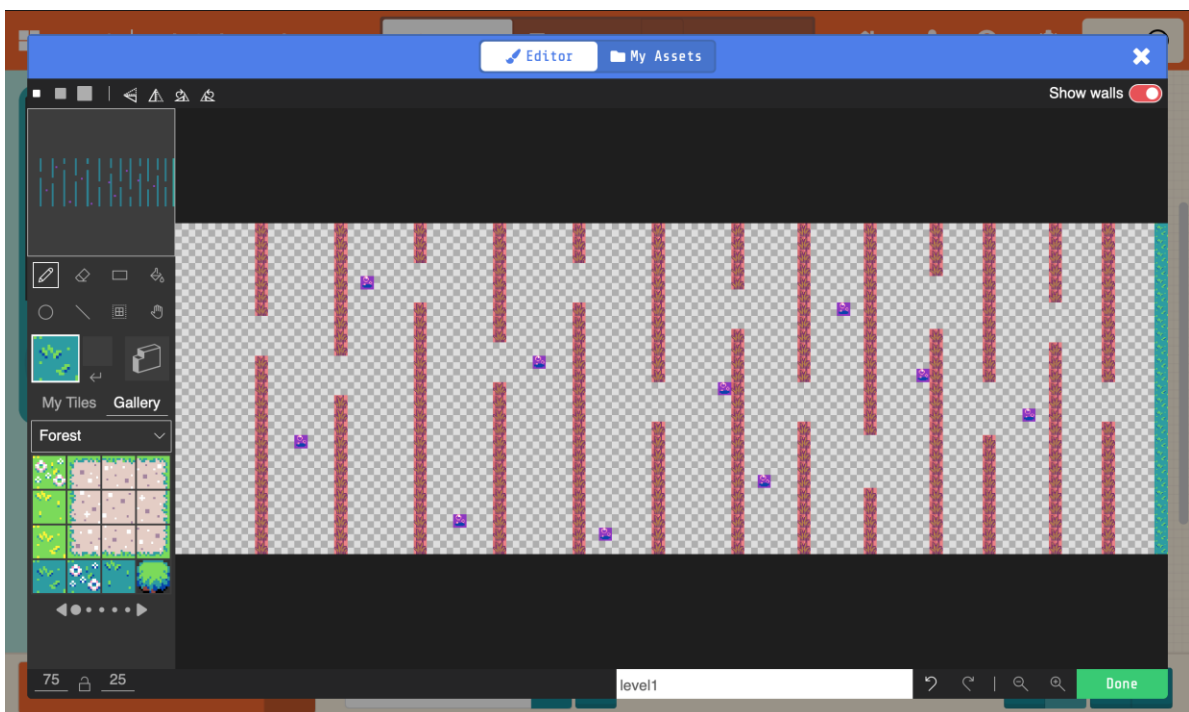


16. Since we have added lives to our game, now we need to tell the code that we want it to be taken away if the fish sprite hits the purple bubble tile. We need to go back into the dark pink “info” tab in the toolbox and select “change life by -1” and put it in the sprite overlap box from earlier.

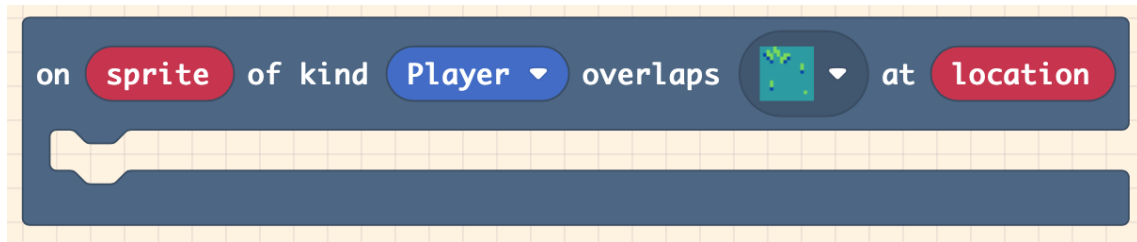


17. It is important for us to define how we win the game in our program. To do this, we are going to use a similar code to how you lose a life by overlapping a map tile. Our code should be working and our tile map is done, so we can add a finish line for our fish sprite to win the game.

To start, we need to add the new tile to the game, just like before when we were designing our tile map. For this example, we used a teal and green tile block so it was different enough from the other obstacles.



18. To code the finish line, go to the toolbox under the dark blue “scene” tab and select “on sprite of kind Player overlaps [] at location.” So, once our fish sprite swims into the new obstacle at the end of the tile map, the game is finished. This block of code only needs to have the tile changed. To do that, double click the grey square and select the tile that you just added.



19. Finally, to finish and win the game. To do that we need to go under the purple “game” tab in our toolbox and select “game over <lose> +,” be sure to click and swipe this button from the grey “lose” to green “win.”



20. To send your game to your PyBadge, click the download button on the bottom left of the screen. Once it transfers, you have a fully playable game on a super cool piece of hardware.

Troubleshooting:

Help! We are having problems with the code!

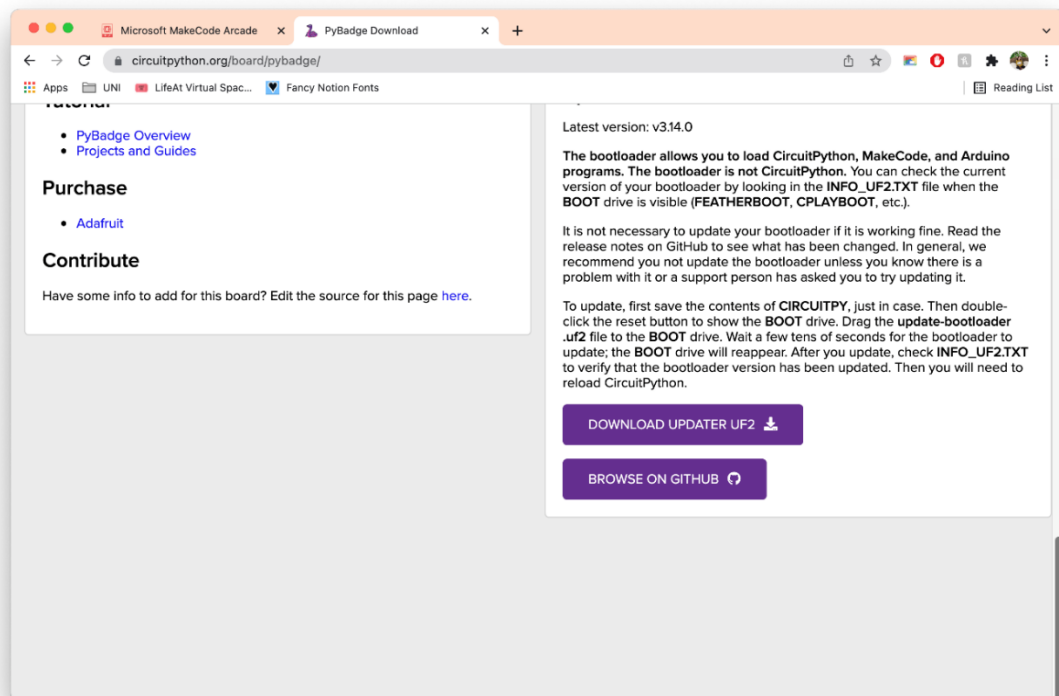
If you are having difficulties with the code, the full game code can be found here:
https://MakeCode.com/_iLtesfDTvAxu

When I try to download my program on Arcade MakeCode, the PyBadge isn't showing up!

If your PyBadge isn't showing up as an option to connect to on Arcade MakeCode, it may have out of date software.

To update your PyBadge software and to ensure you can upload from Arcade MakeCode, you will need to update the software. To do so you will need to:

1. Ensure that your PyBadge is plugged into your device correctly.
2. With the PyBadge still connected, reset your PyBadge by double pressing the "reset" button on the back top left of the PyBadge.
3. Download the update from <https://circuitpython.org/board/pybadge/>. Scroll down to the bottom of the page and select "Download Updater UF2".



4. Once the PyBadge has a screen that says “PyBadge” with a green and blue background, drag and drop the downloaded file onto the USB device found in your “My Computer” page and it should restart itself.
5. Once the device is restarted, you should be ready to upload code from Arcade MakeCode!