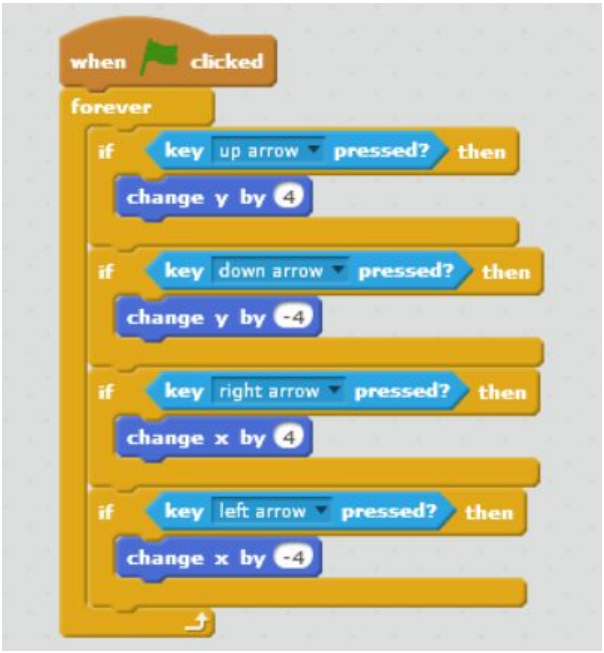
Lesson 2 - Maze Runner Game

1. Make your sprite move
    a. Learn about coordinates with the Coordinate Explorer program.
    b. Everyone can select their favorite sprite to play with or just stick with the cat.
        i. If they select a new one, get rid of the cat (right click sprite > delete *or* select the delete tool (scissors) above the stage and click the sprite).
    c. Add the following code to the sprite scripts space:

        i. This allows the sprite to move up by repeatedly checking if the up arrow is pressed and if so, changing the y-coordinate by +4.
    d. SAVE
    e. Now duplicate the **if** block to allow the sprite to move in the other directions.
        i. Right click the **if** block > duplicate.
        ii. Drag the new block below the existing **if** blocks inside the forever loop.
        iii. Change the **arrow** key pressed sensore to down/left/right.
        iv. Change the **change y by** to -4 for down, or replace with **change x by** 4 for right and **change x by** -4 for left.
        v. The code should look like the snippet below when done:



f. SAVE and RUN

2. Add the mazes
    a. Go to: https://github.com/jmatthewfrazier/ScratchClass
    b. Download the Mazes file
        i. You can do each maze individually or download a zip of the entire master branch as a zip.
    c. Add the mazes as a sprite.
        i. In the Sprite Box, select the **Upload sprite from file** button.
        ii. Find and select the maze1.png.
        iii. Now select the maze 1 sprite and go to its **Costumes** tab.
        iv. Select the **Upload sprite from file** button again (the one in the **Costumes** tab this time).
        v. Highlight the mazes 2-8 (maze1 is already in the program) and click **open**.
            1. This will add all the remaining mazes as costumes to the maze sprite.
            2. Fun Tip: you can draw your own mazes as sprites or find others online (online mazes should be .png files or your backdrop will get covered up).
    d. Add a backdrop.
        i. In the Backdrop box, click **Choose a backdrop from library** and select a backdrop for the game.
            1. Try to pick something light in color, otherwise it might be difficult to see the maze.
    e. Start up the maze.
        i. Select the maze sprite.
        ii. Go to its Scripts tab and add the following code:

            1. This code block starts the game on the maze1 costume and centers the maze in the middle of the screen.
    f. SAVE and RUN (You'll see that our character sprite can walk through walls! This is cool, but we don't want this feature).
3. Keep the sprite from moving through the walls.
    a. Add the following code under the **change y by** 4 snippet in the **if** up-arrow control block for the character sprite:

i. This means that if the sprite moved up and touched part of the maze, it will move back the distance it travelled up.
ii. It moves back so quick we can't see it. Instead, it looks like the sprite never moved at all.

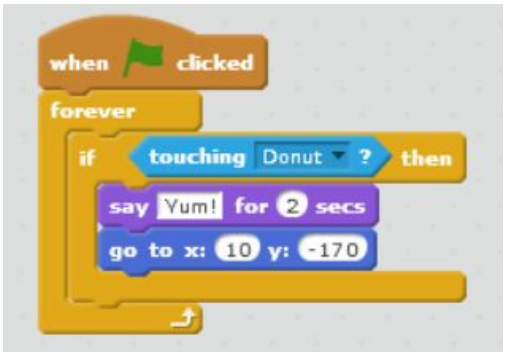
b. Duplicate the code for the other movement **if** blocks making changes to the **change *direction* by** snippets as necessary.
i. Your code should look like this:

c. Add final code to the character sprite.
    i. Our character is behind the maze. Let's move it in front.
        1. Add the **go to front** block (Looks) under the **when *flag* clicked** block and ahead of the **forever** loop.
    ii. Our sprite is too big. Let's set the size.
        1. Add the **Set size to *x*%** block (Looks) right under our last one.
        2. Set the size to 15%.
    iii. Our sprite isn't at the start. Let's move it there.
        1. Add the **Go to x: *x* y: *y*** block (Motion) under our last block.
        2. Set the x value to 10 and the y value to -170.
    iv. Our Sprite code should look like this:

d. SAVE and RUN

4. Add a goal to at the end of the maze.
    a. Create a new sprite.
        i. Click the **Choose sprite from library** in the Sprite box.
        ii. Choose a sprite (the **Things** tab is a good place to look).
    b. Add the following code to resize and position the goal sprite:

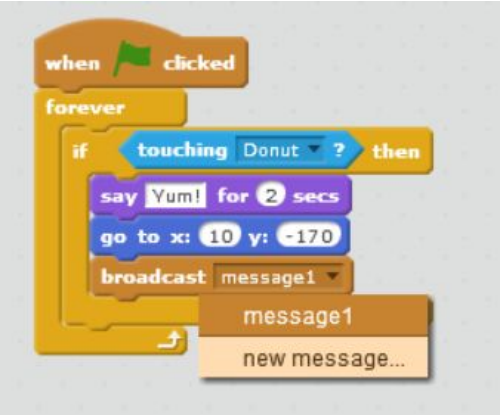
    c. Now we want to know when our character has reached the goal.
        i. Select your character sprite again.
        ii. Add the following script to your character's code:



1. This script is not attached to your already existing code under your character sprite.
2. It tests forever if your character touched your goal (a donut in my case), then your sprite says a win message (you can make this whatever you want), and moves your sprite back to the starting position.

iii. Add a **broadcast** block (Events).

1. Drag it under the **go to x: y:** block that you just added.
2. Select the dropdown menu in the **broadcast** block and select **new message**.



3. Type “next maze” in the text field and click **OK**.
4. **Broadcast** sends a message to all sprites in your game. Our message is saying that we are done with the current maze and are ready for the next one.

d. Add broadcast handling to the maze sprite.

i. Go to the maze sprite’s script area.

ii. Add the following code:

1. This code “listens” for the *next maze* message and when it hears it after your sprite completes a maze, it switches its costume to the next maze.

e. SAVE and RUN.

5. WERE DONE!!!

a. Other ideas:

i. Try creating your own mazes.

ii. Try creating a 2-player game.

iii. Try adding traps (spikes that pop up and down)