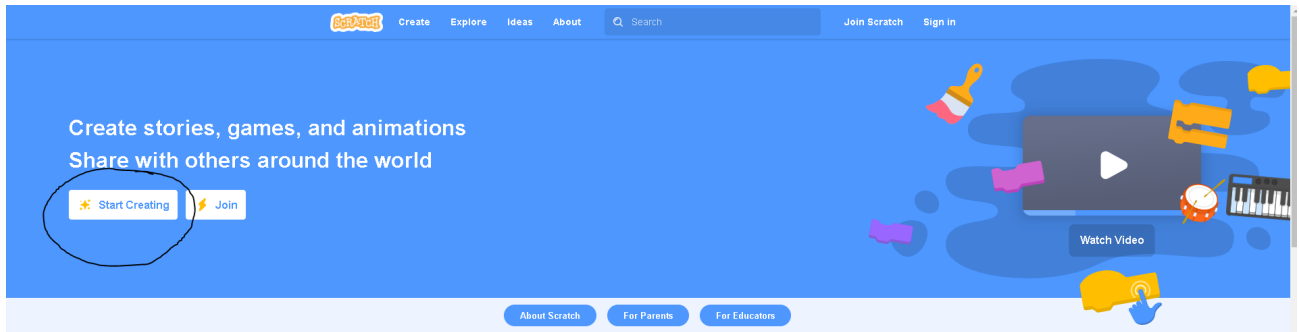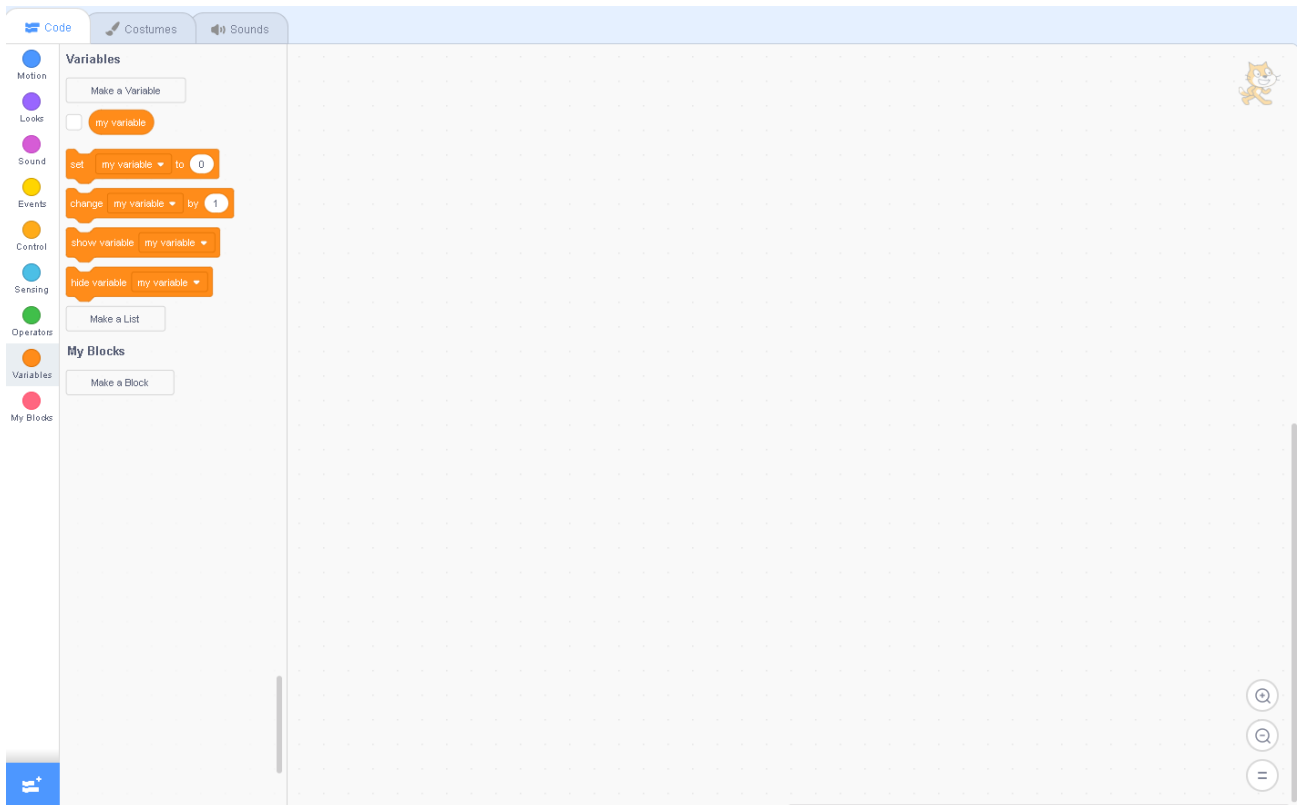# Idle Game Instructions (Scratch)

1. Visit scratch.mit.edu and clock Start Creating

   a.

2. Explore the layout and tabs

   a.

   b. Tab 1 Code: Contains all of your draggable elements plus workspace.

      i. Motion - Move objects, rotate or check for collisions

      ii. Looks - Output, change sprite "costumes", color effects, show and hide

      iii. Sound - Play sounds, start more than one sound, adjust pitch, volume

      iv. Events - When start clicked, when key pressed, when sprite clicked, when condition (> < >= <= !=), when message, and broadcast

      v. Control - Loops, delays, if/else/else if, for loops, stop, clone

      vi. Sensing - Collision or touching, color, input, key press, mouse position, mouse clicked, draggable, timer, date, username

      vii. Operators - Add, subtract, multiply, divide, random, check > < or = check && or || and !=, join, length, round, Math functions

      viii. Variables - Create variables, set, increment, show, hide, list (array)

      ix. My Blocks - Create your own methods and functions
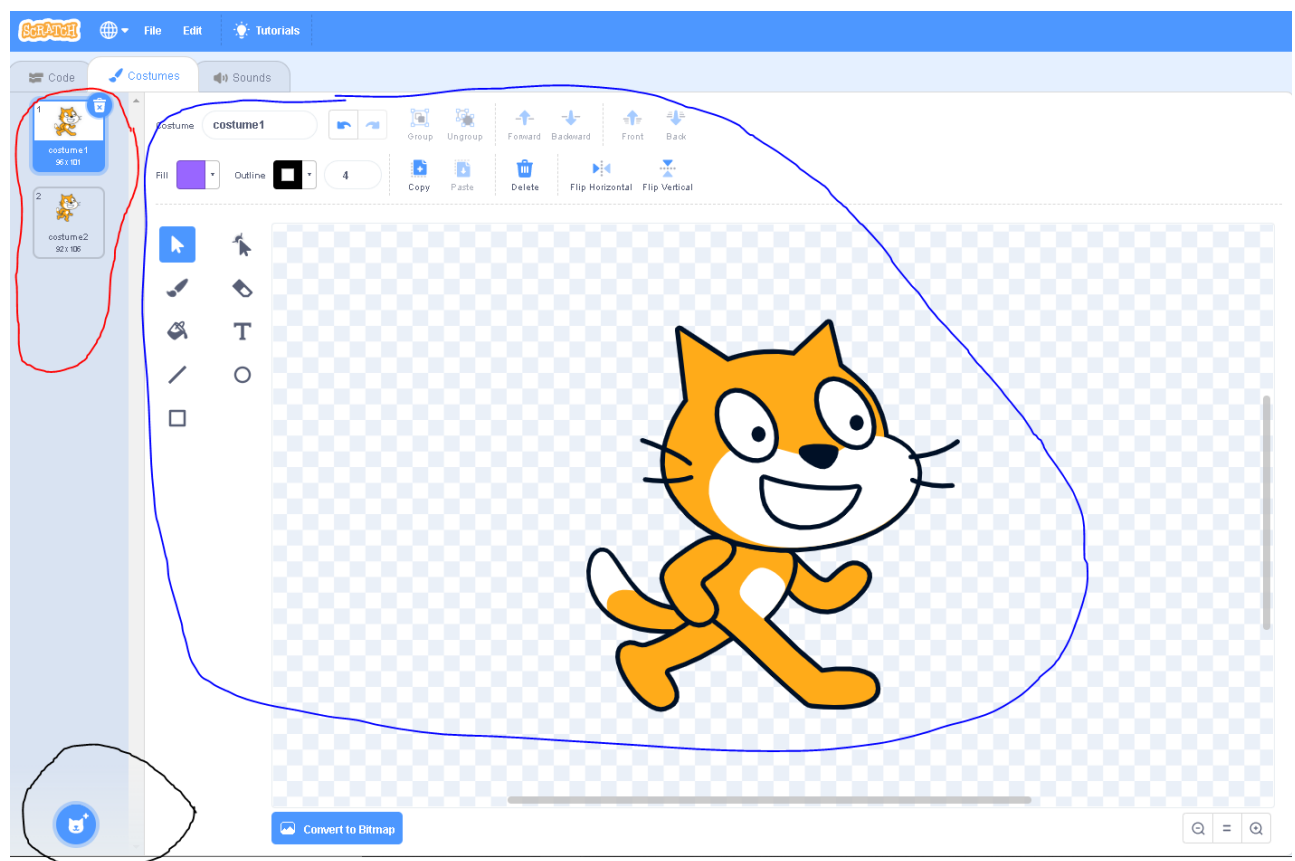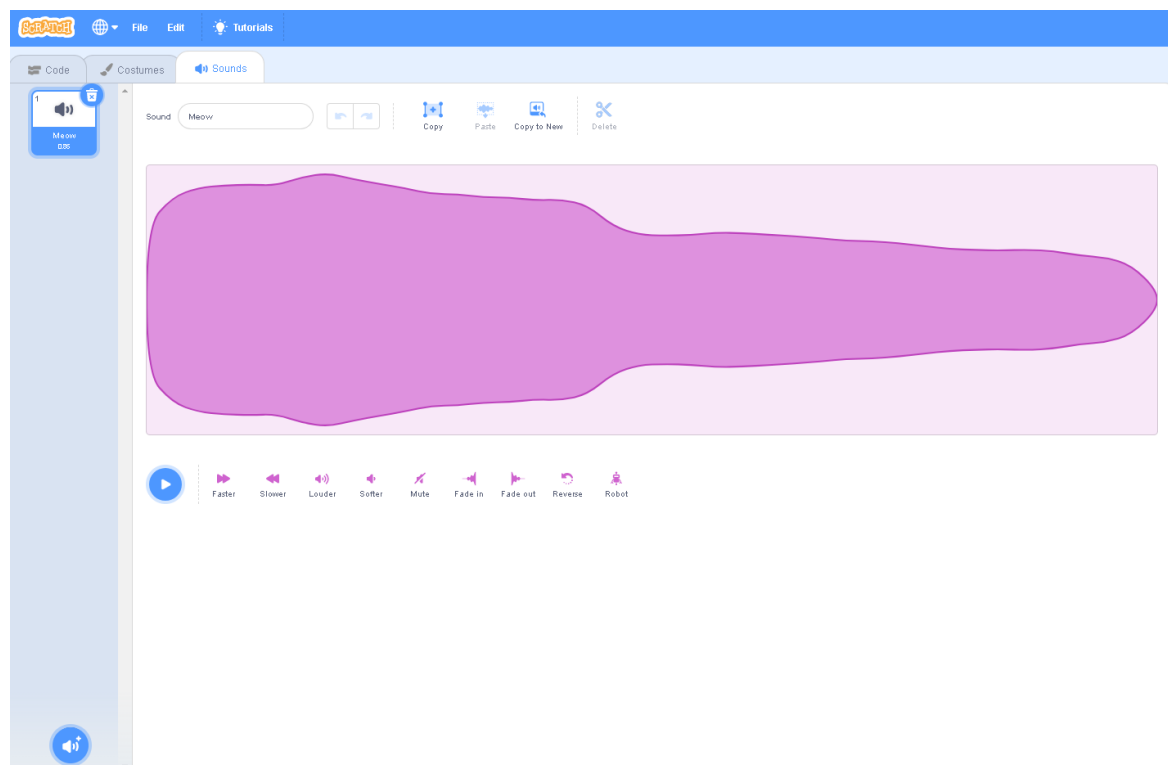
c.

d. Tab 2 Costumes:

    i. Color Code:

        1. Red: All costumes. Costumes represent the frames of a moving sprite. In the example the cat has two poses.

        2. Blue: The design area. In this area you can edit, create and adjust your sprite frames or Costumes.

   3. Black: New, Search, Upload or Random costumes.

e.

f. Tab 3 Sounds: Upload and manage all sound effects, songs, etc
    i. Just like costumes you have a view of all sounds, the waveform that you can modify and then a button in the bottom left to upload, random, search or adjust sounds.

3. Edit costumes 1 and 2 with a custom graphic. Costume 1 should be a standard graphic (think button or bubble or balloon). Costume 2 should be a pressed button, popped bubble or balloon. You can use your own two custom graphics, but I will be using a balloon. See below:
   a. Costume1 should be renamed to BalloonNormal or some other useful name. Costume1 isn't helpful
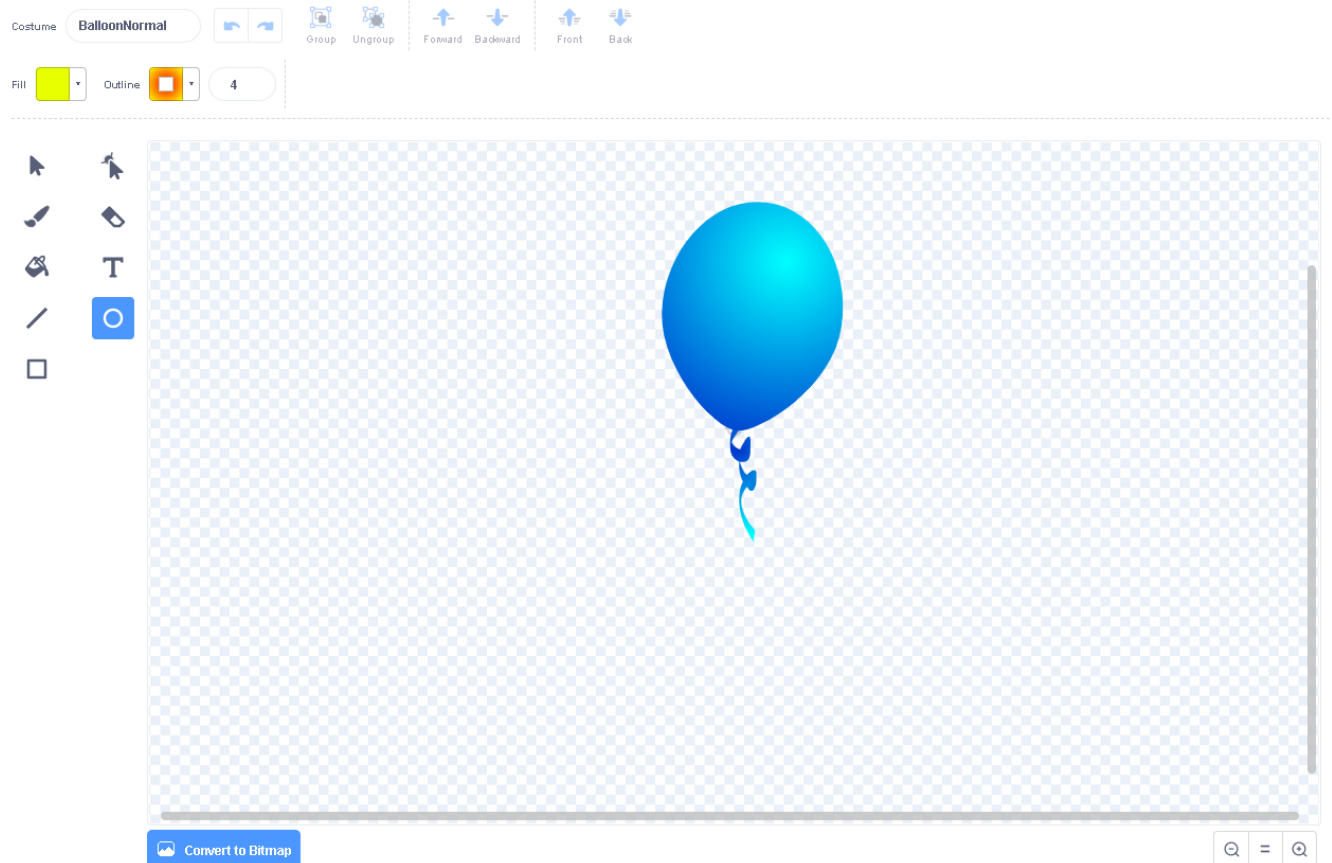
    b.
    c. Costume2 should be renamed to BalloonPopped or some other useful name. Costume2 isn't helpful, and you need to alter the graphic to show the change.In my case "pop" the balloon


    d.

4. Create our first events. We will create 1 event to track when the user starts the game and a second event to track when the user clicks on the balloon.
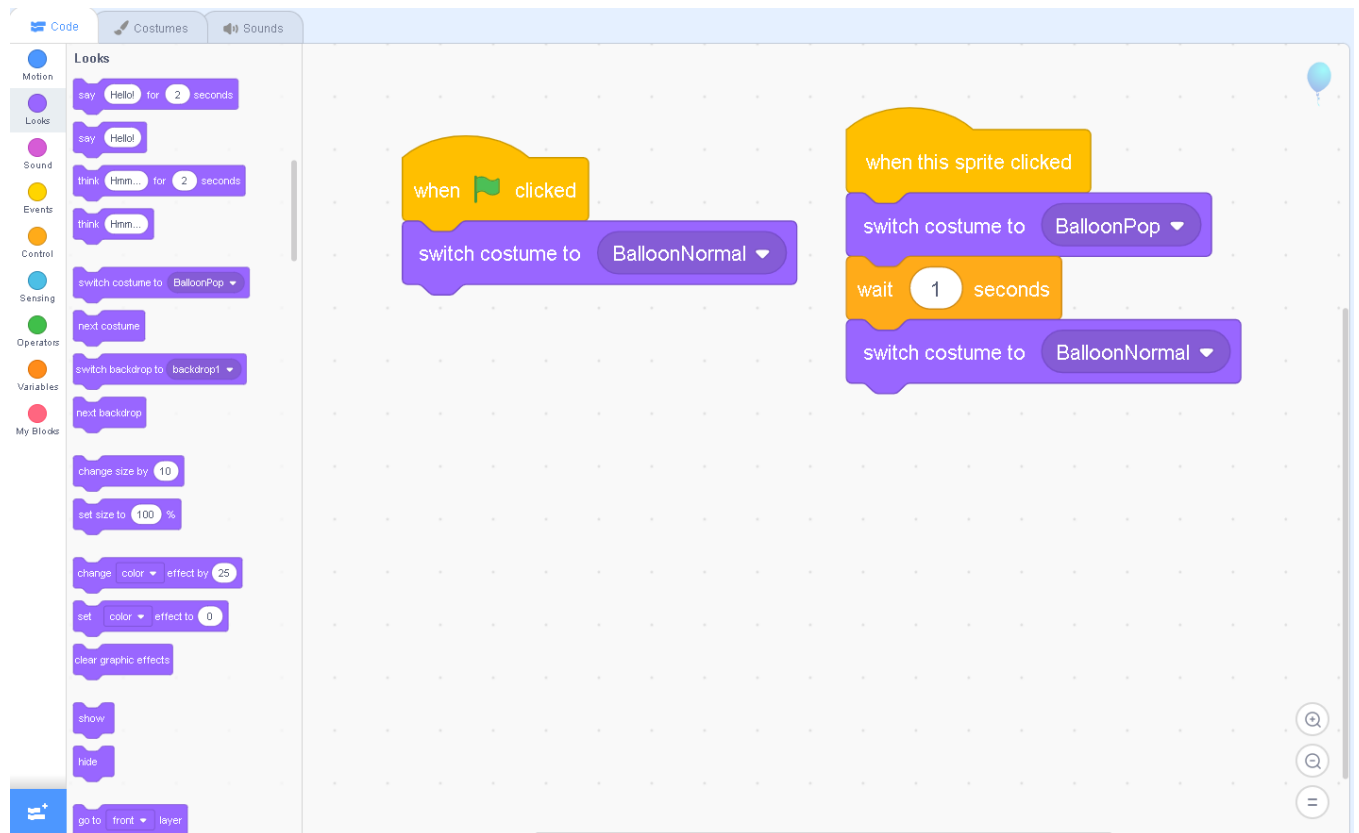
a.

b. You will need pieces from Events, Looks and Control to make these two sections of code.
c. Test your code! Does the sprite reset to your normal sprite when you click the green start flag? Does the sprite change when you click it and then reset to the normal sprite after 1 second?
    i.   If yes Move on to step 5!
    ii.   If no, Check your costume names and make sure they match your blocks. If your costumes have different names you will need to change the code blocks to match your names.

5. Add some sound effects when the sprite is clicked. By default scratch gives us Meow… Lame! Instead you can add any mp3, wav sound of your choice or use one from hitting the button in the bottom left of the sounds tab and browse the library of sounds. I found one called Pop which is perfect for the balloon!
   a. Edit your code block for sprite click to include the sound when clicked. See below.
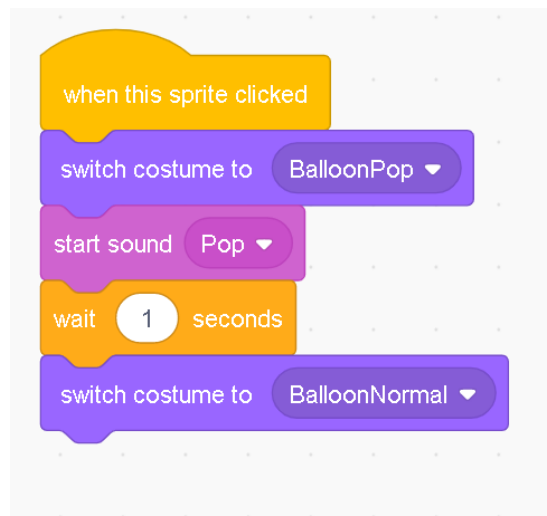
    i.
    ii.   Test! Does the sound play? Does the sound lag the game? If yes then you used play sound until done! This locks up the game! Do not use it!

6. Track the number of clicks by the user!
    a. We will do several things here.
    b. Create a variable for user clicks and for pointMultiplier
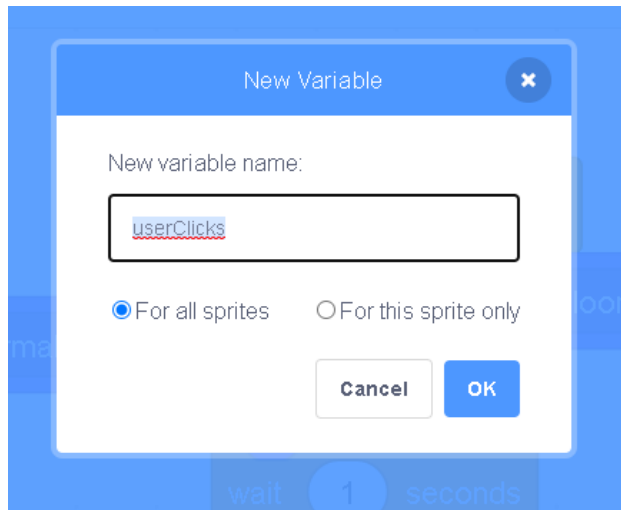
        i.
    c. Create an overall game loop that prints out the user clicks total and add clicks when sprite is clicked
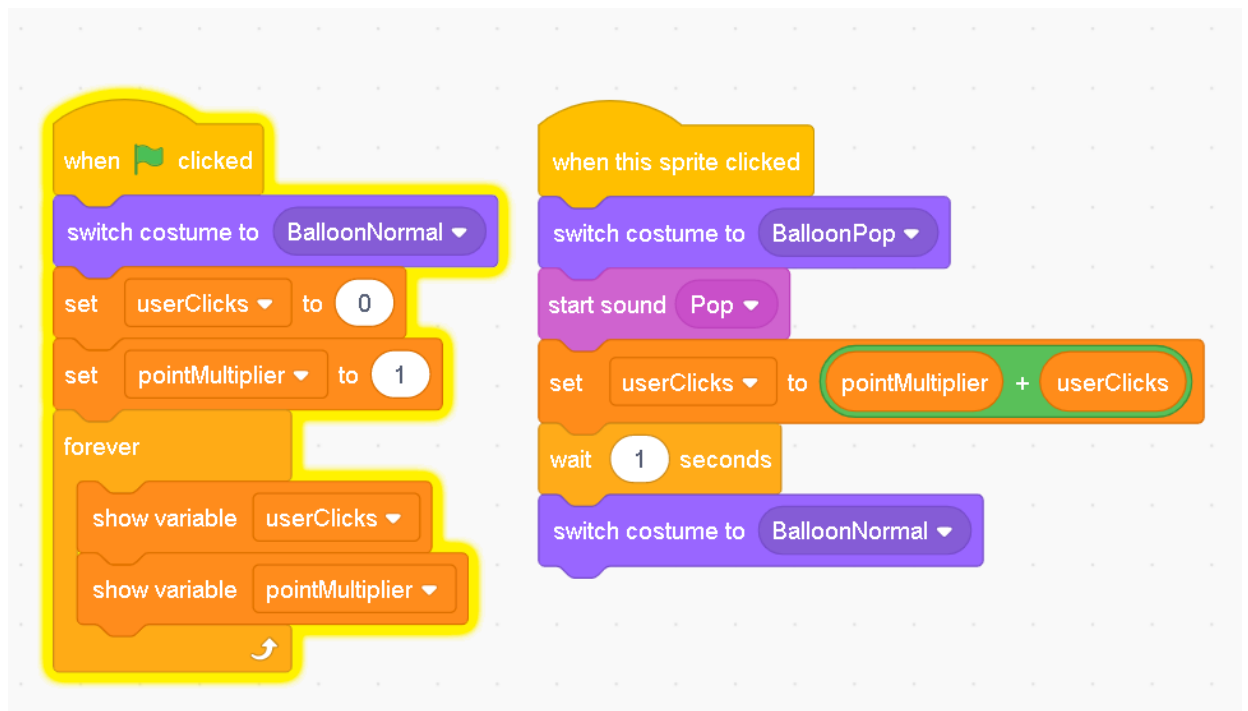
        i.
    d. Add some conditions to adjust the pointMultiplier to add to amount per click as the user progresses in the game.
        i.   See the example below. Note… this is where scratch gets messy!

```
when [flag] clicked
switch costume to (BalloonNormal ▼)
set (userClicks ▼) to (0)
set (pointMultiplier ▼) to (0)
forever
    if < (userClicks) < (50) > then
        set (pointMultiplier ▼) to (1)
    else
        if < (userClicks) < (200) > then
            set (pointMultiplier ▼) to (2)
        else
            if < (userClicks) < (1000) > then
                set (pointMultiplier ▼) to (3)
            else
                if < (userClicks) < (10000) > then
                    set (pointMultiplier ▼) to (4)
                else
                    set (pointMultiplier ▼) to (5)
    show variable (userClicks ▼)
    show variable (pointMultiplier ▼)
```

ii.

iii. Test! If it works as expected take a screenshot of your entire screen or call me over and show me. If you call me over just hit submit on schoology, Otherwise submit the screenshot