

PURPLE MAI'A



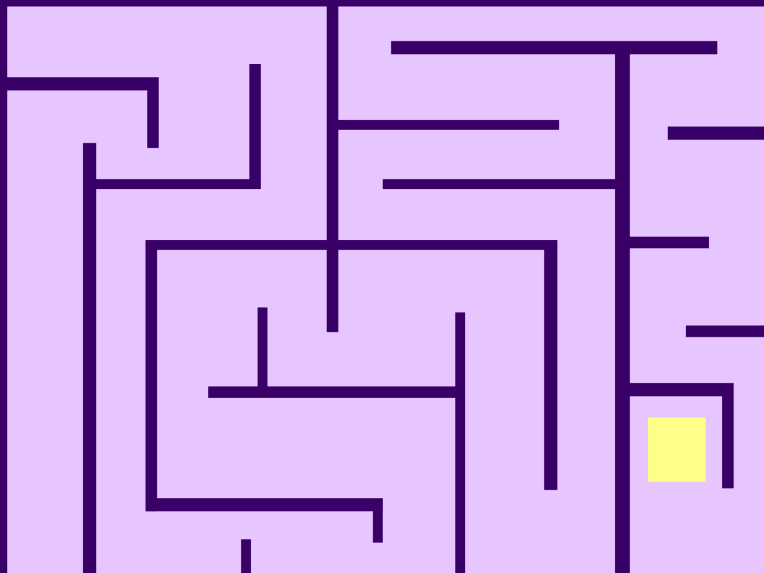
# MAZE GAME

**PART 1** Smooth Play Movement

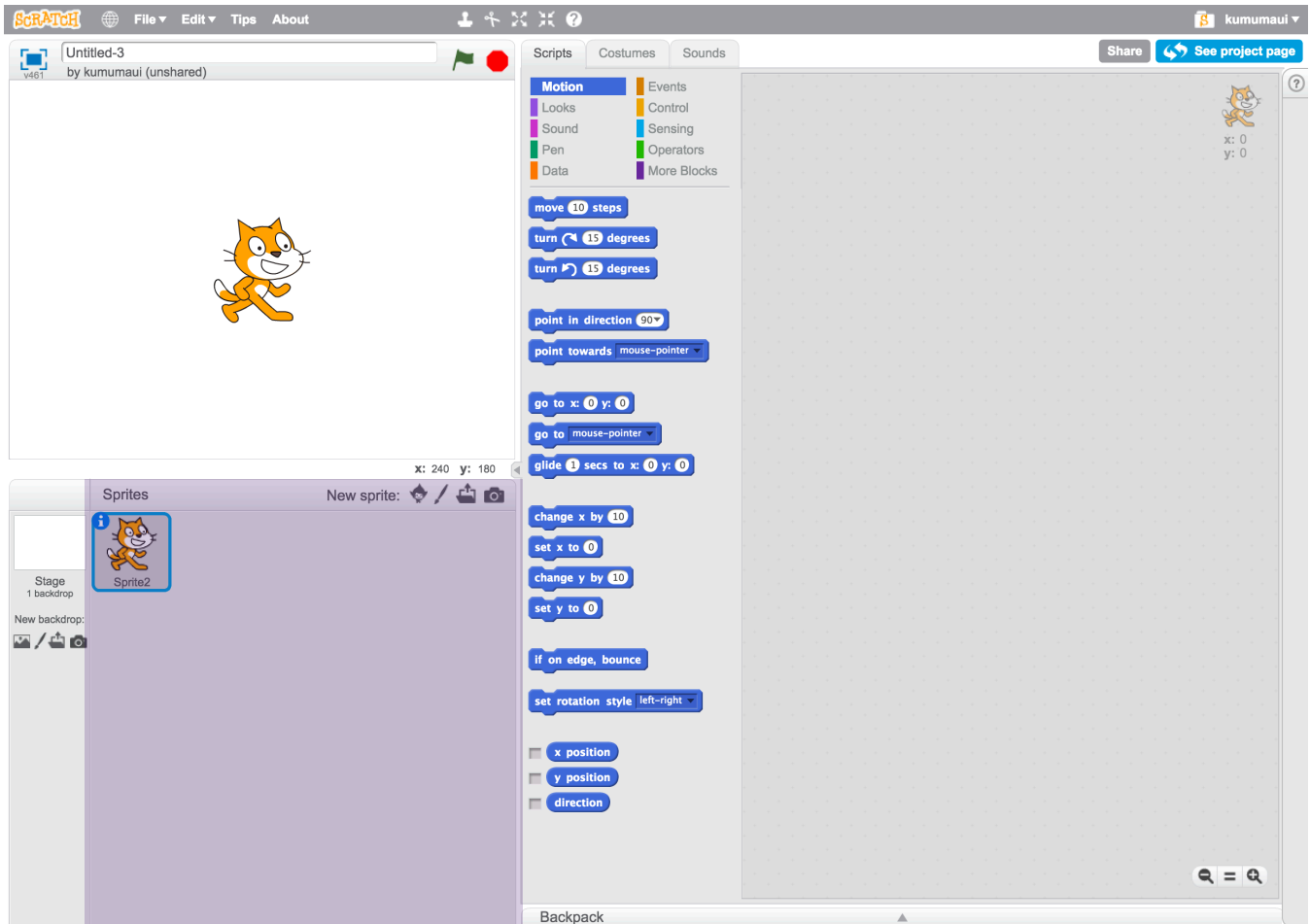
**PART 2** Graphic Design

**PART 3** Game Design

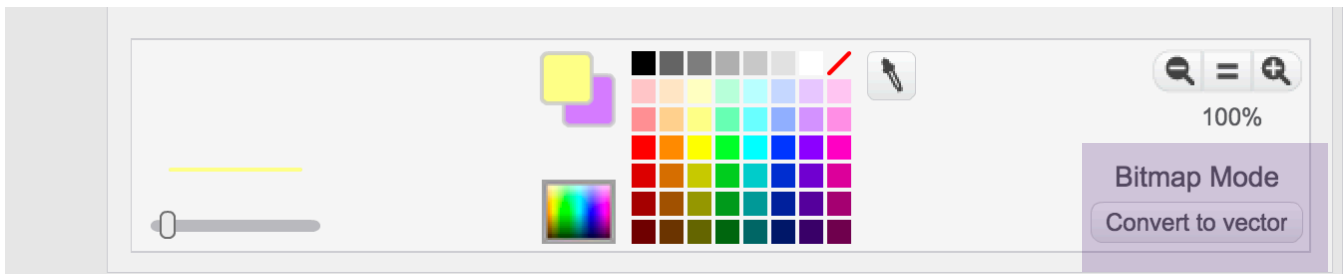
**PART 4** Adding Score



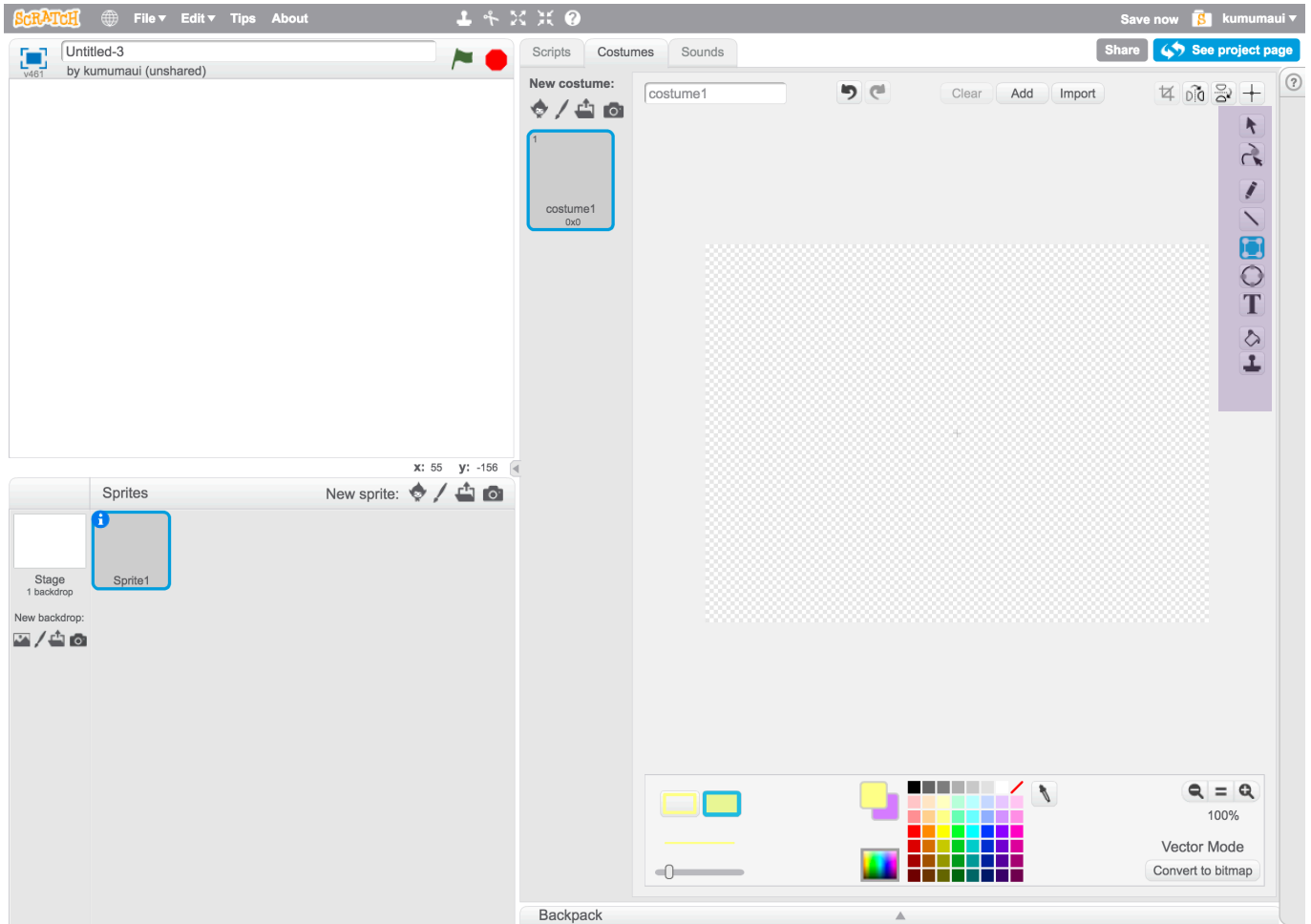
Delete Scratch (cat) by right clicking on the icon to get a dropdown.



Create a new backdrop, by clicking the **paintbrush** under New Sprite.

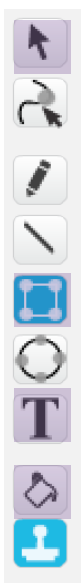


Find the **"Convert to Vector"** button and click it



Draw a small (15x15) square. Make sure to choose your color and that this is not an outline.

### Tools we will use today

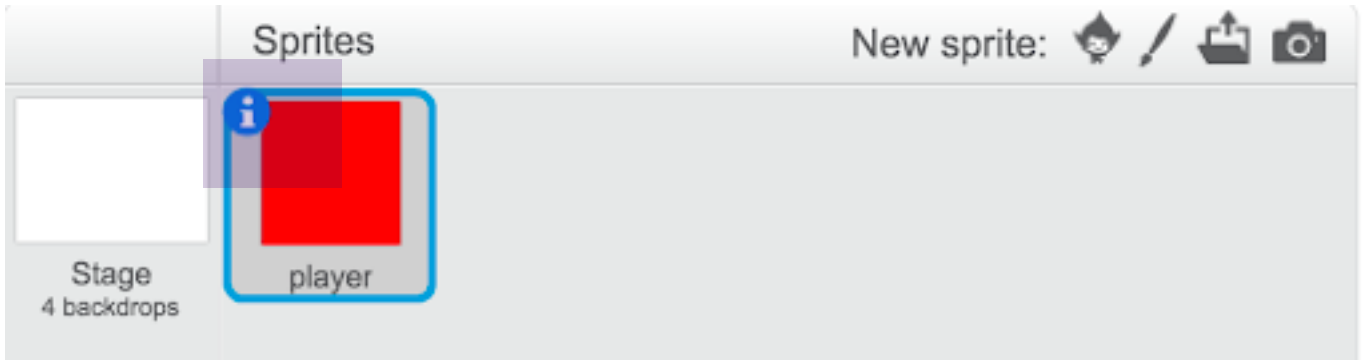


Click and resize vector objects

Draw rectangles. Hold down shift while dragging to draw a square

Free form text tool

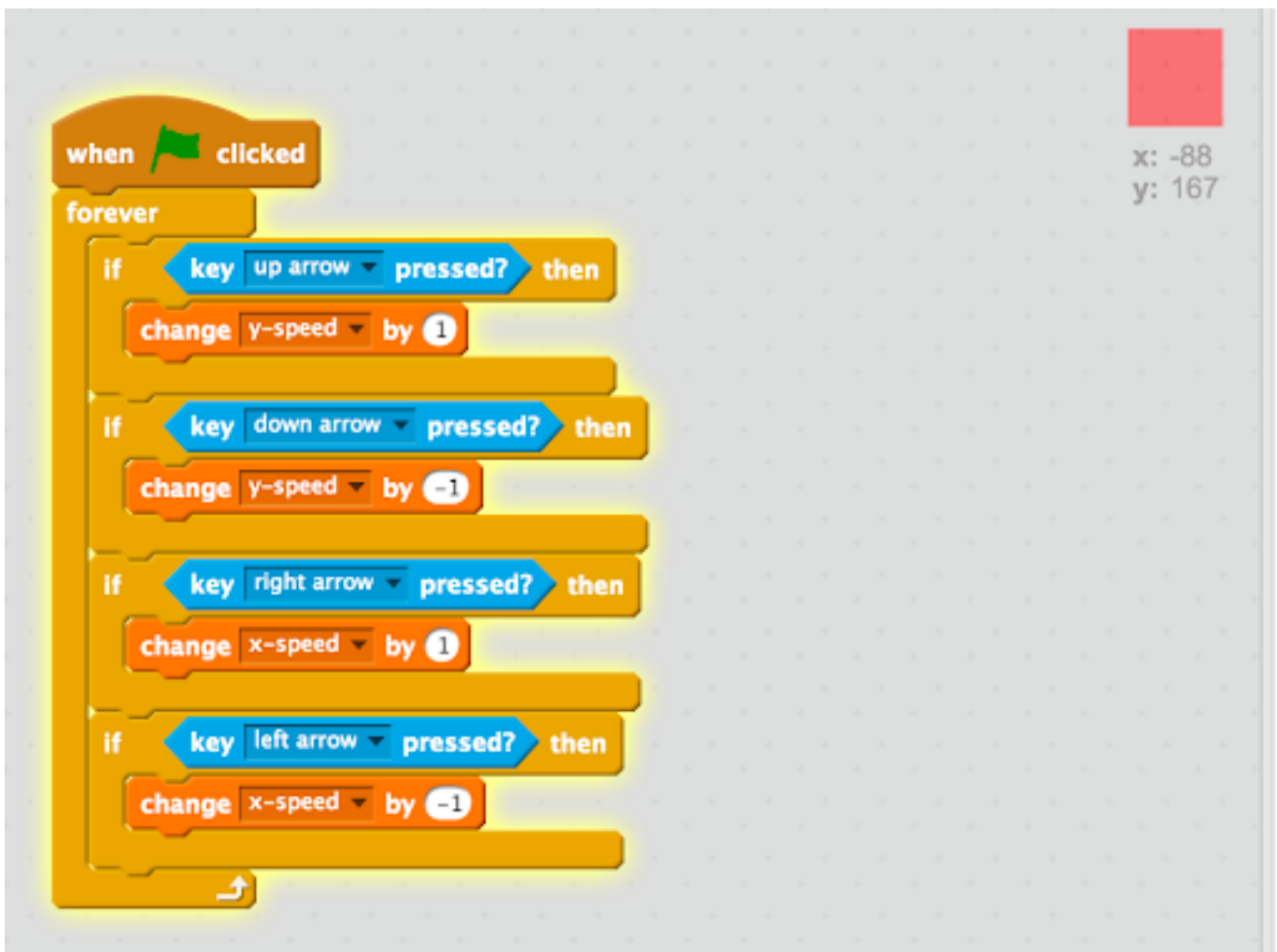
Change color of vector objects



Click the blue "i" to change the name of the sprite to **Player**.

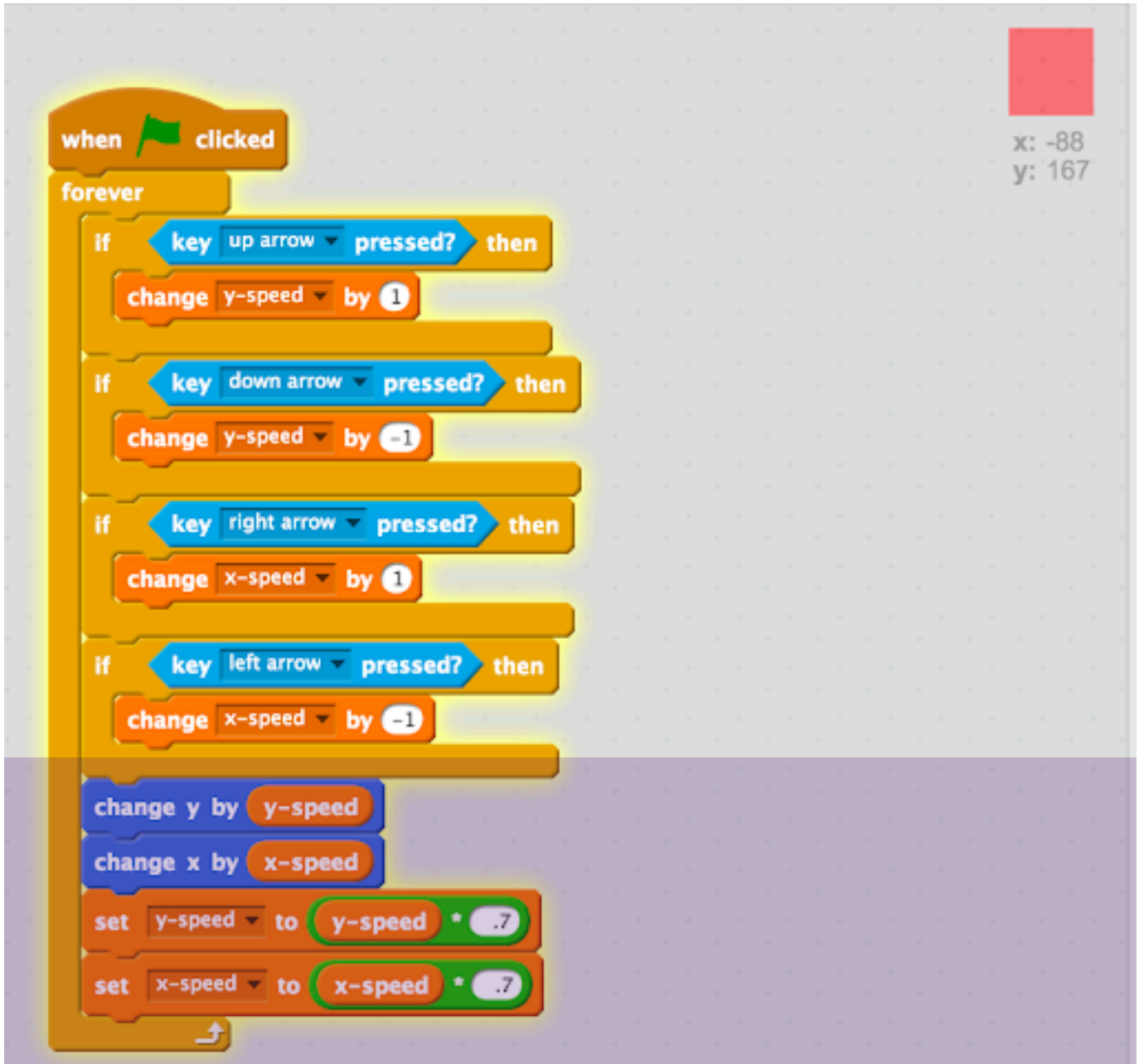
### Coding for moving our player

We create two variables for our speed, one for the x and one for the y. We do this so we can have smooth animation.



### A little more code

We need to make sure to set the speed of the variables and use a multiplier to make them smooth. Experiment with changing the numbers to see how it affects the movement.

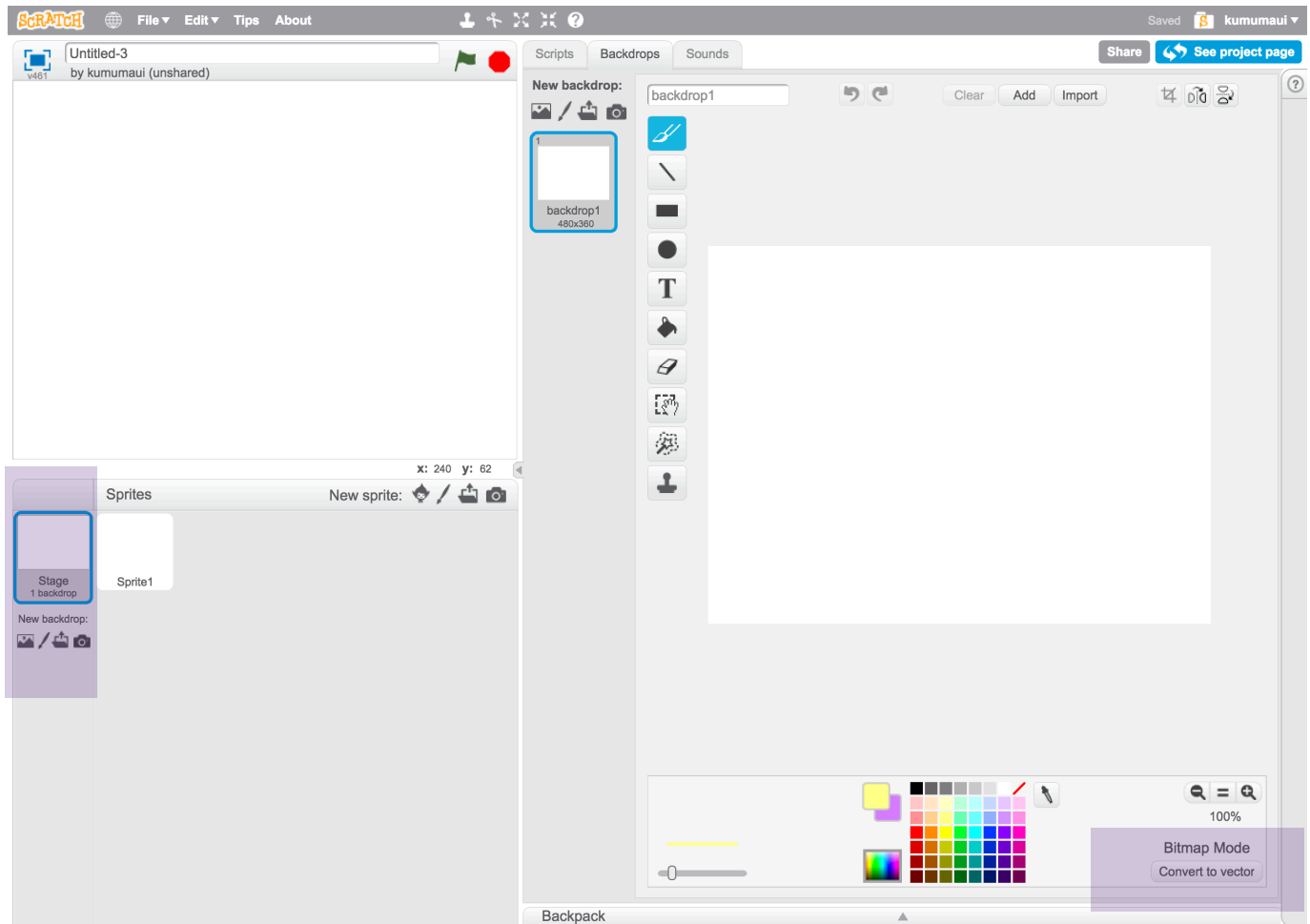



Smooth Play Movement


# COMPLETED

**We will design the backdrop to give the user a game experience. Remember to work in vector mode!**  
**We do this so the game looks good no matter what the size is!**

We will start with the backdrop. Click on the backdrop and then click “convert to vector”

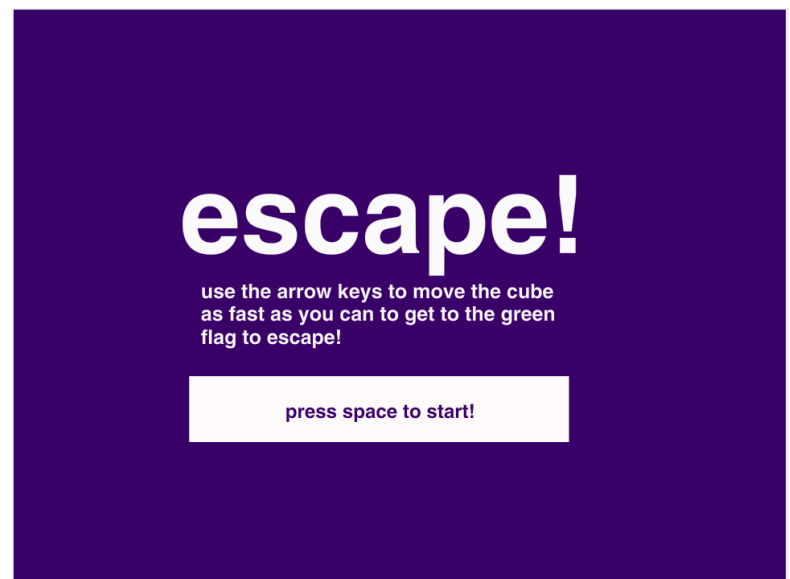


Use the  and click on white background. **Delete it.**

Use the  tool to draw rectangle with a bold color to cover the **whole backdrop!**

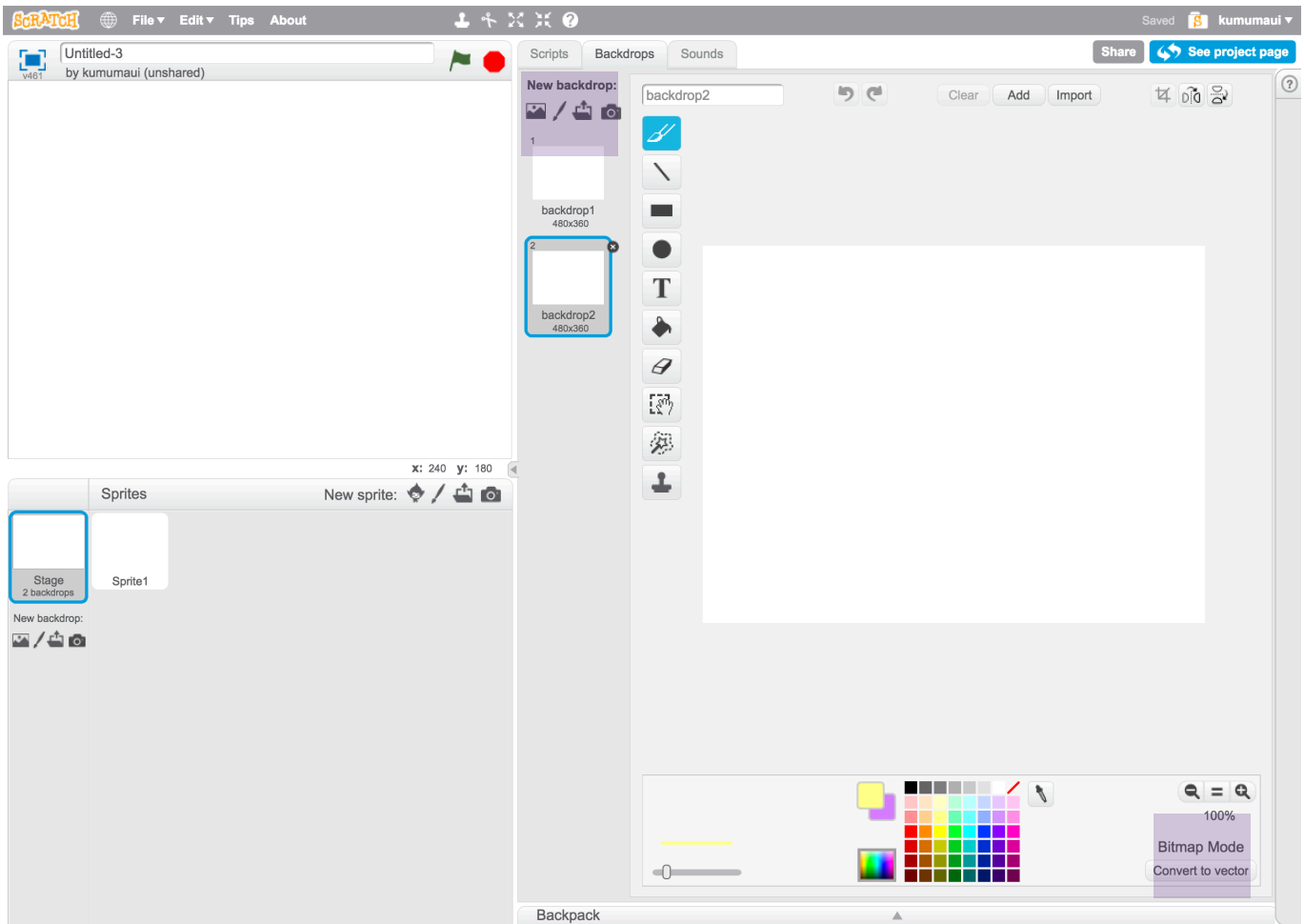
Use the  tool to add text

and the  to select and resize the text.




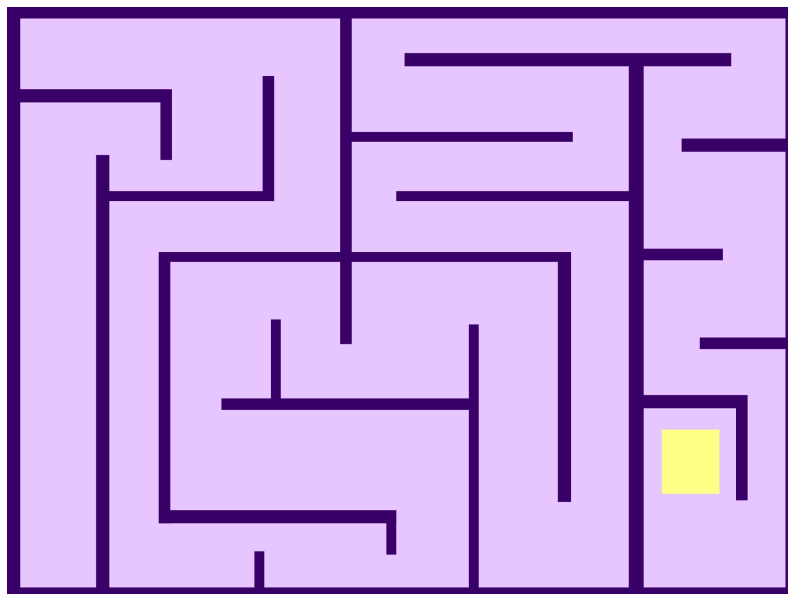
This is what it should look like!

## Design your maze on paper first!




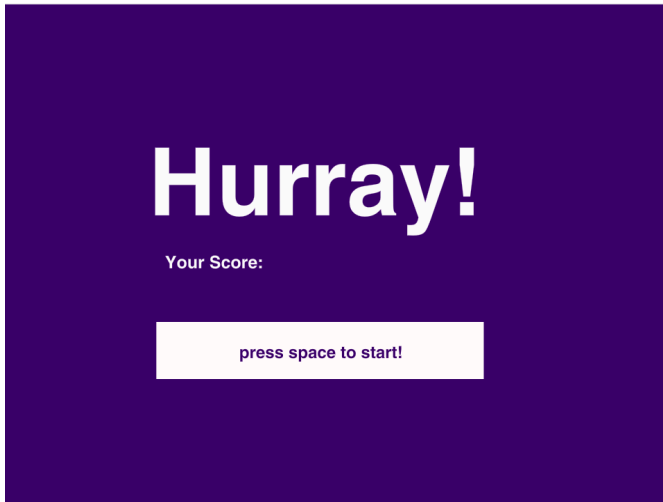
Now **create** a new **backdrop** and **convert it to vector**.

Use the  tool to put a border around the backdrop and continue making squares to complete your maze. Make your end of maze a **different color** and put at the end



**Duplicate background 1 twice and change text for win & lose**

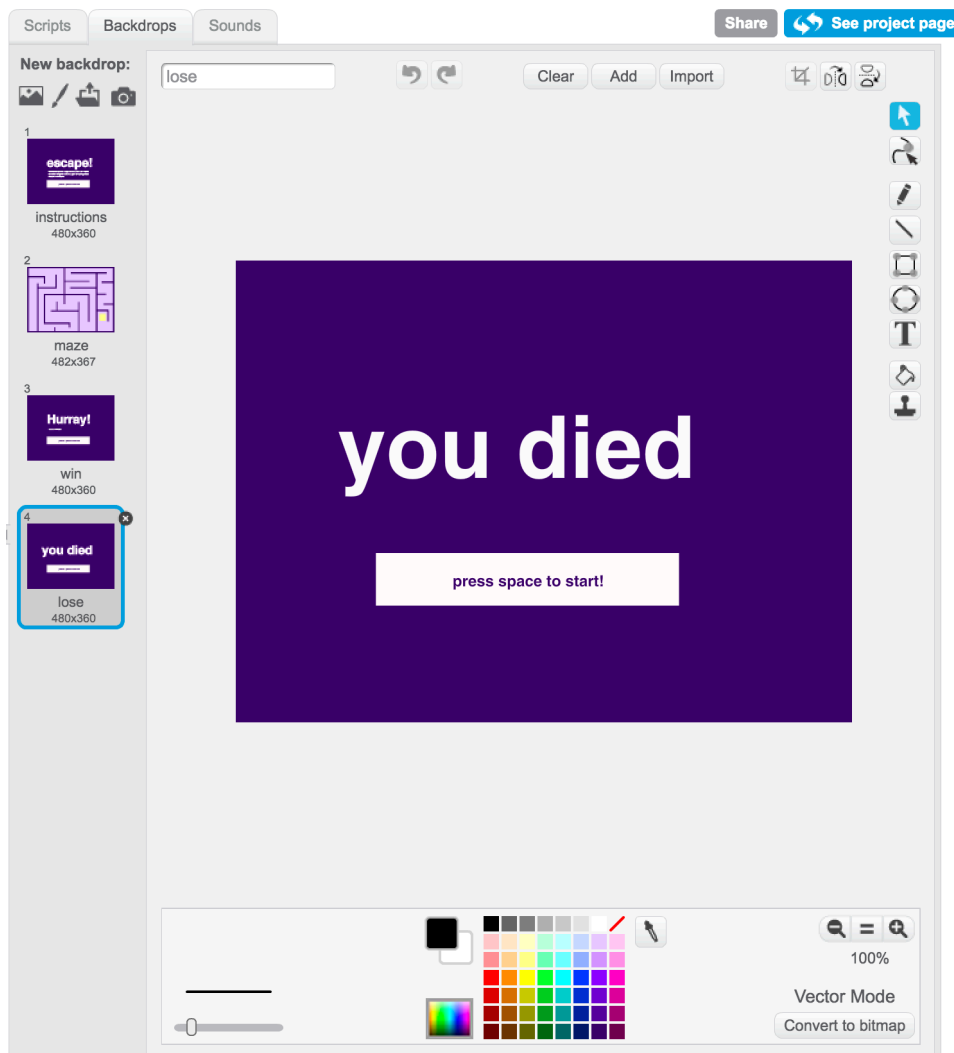
Use the  tool to edit the text




WIN

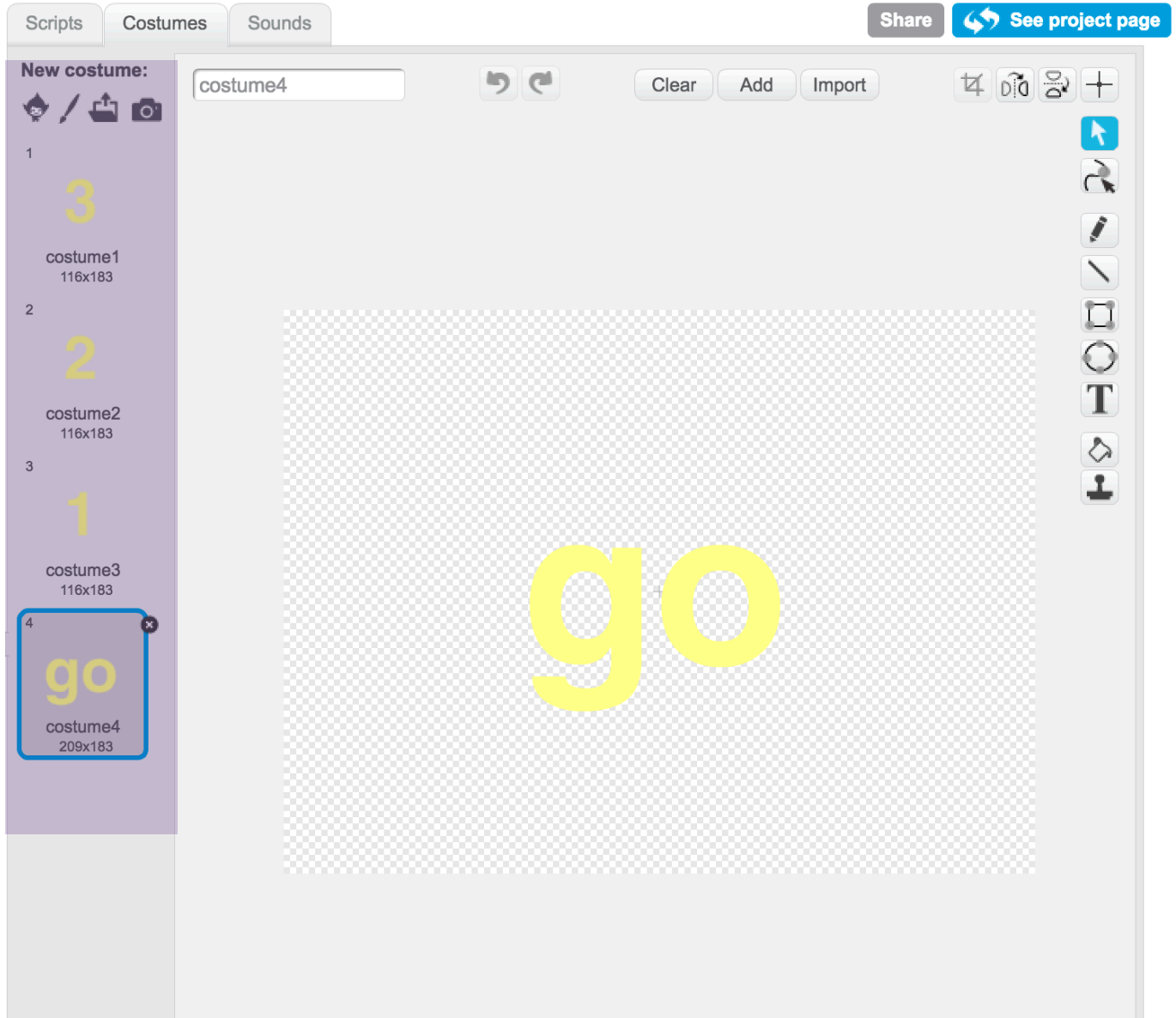


LOSE





Create a new sprite and call it **count down**. **Convert to vector**. Now use the  and make costumes for **3, 2, 1, go!** in that order. Also make sure the text is centered!



GRAPHIC DESIGN

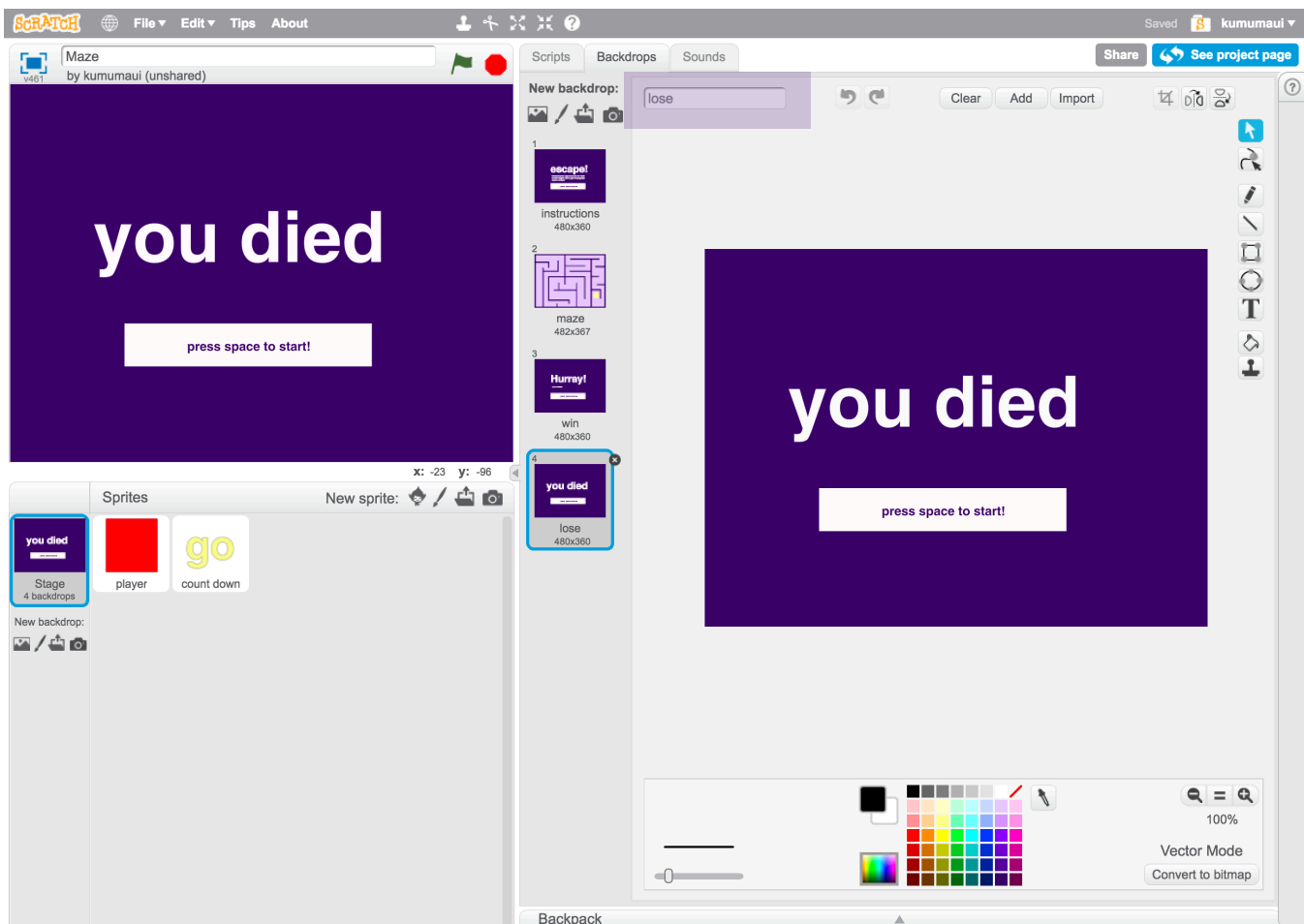
# COMPLETED

## We are going to control the game with a variable and add color detection!

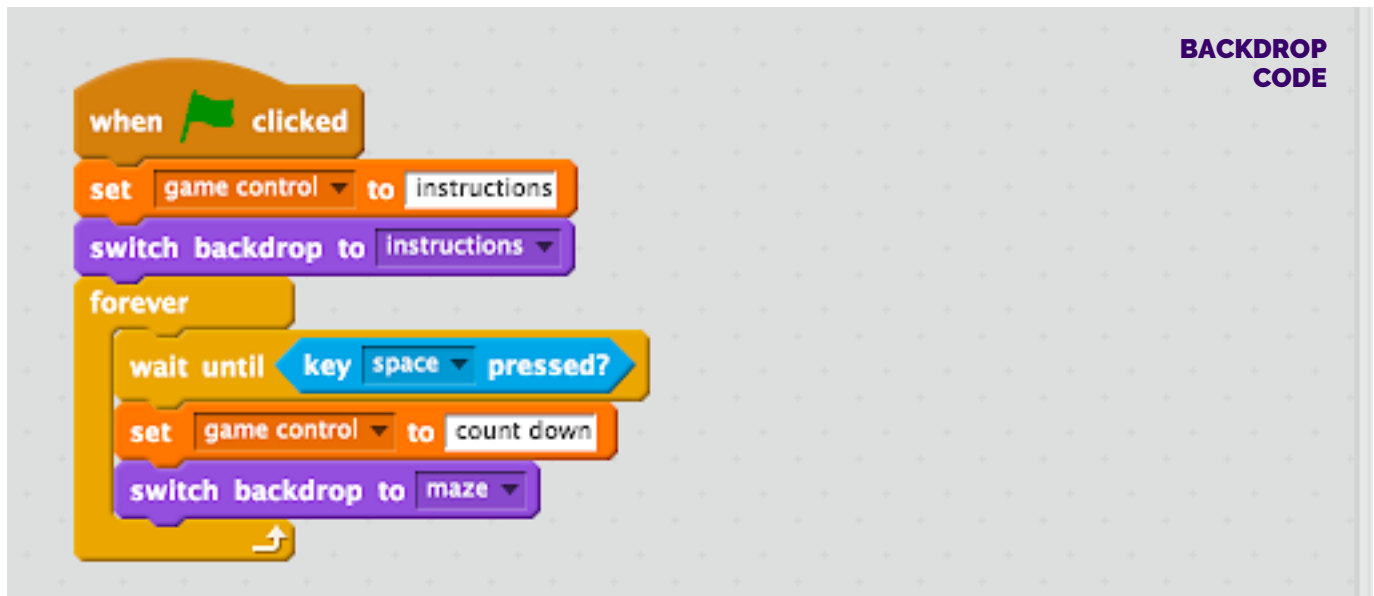
First create a variable called game control and then go over to the **backdrops** to start coding



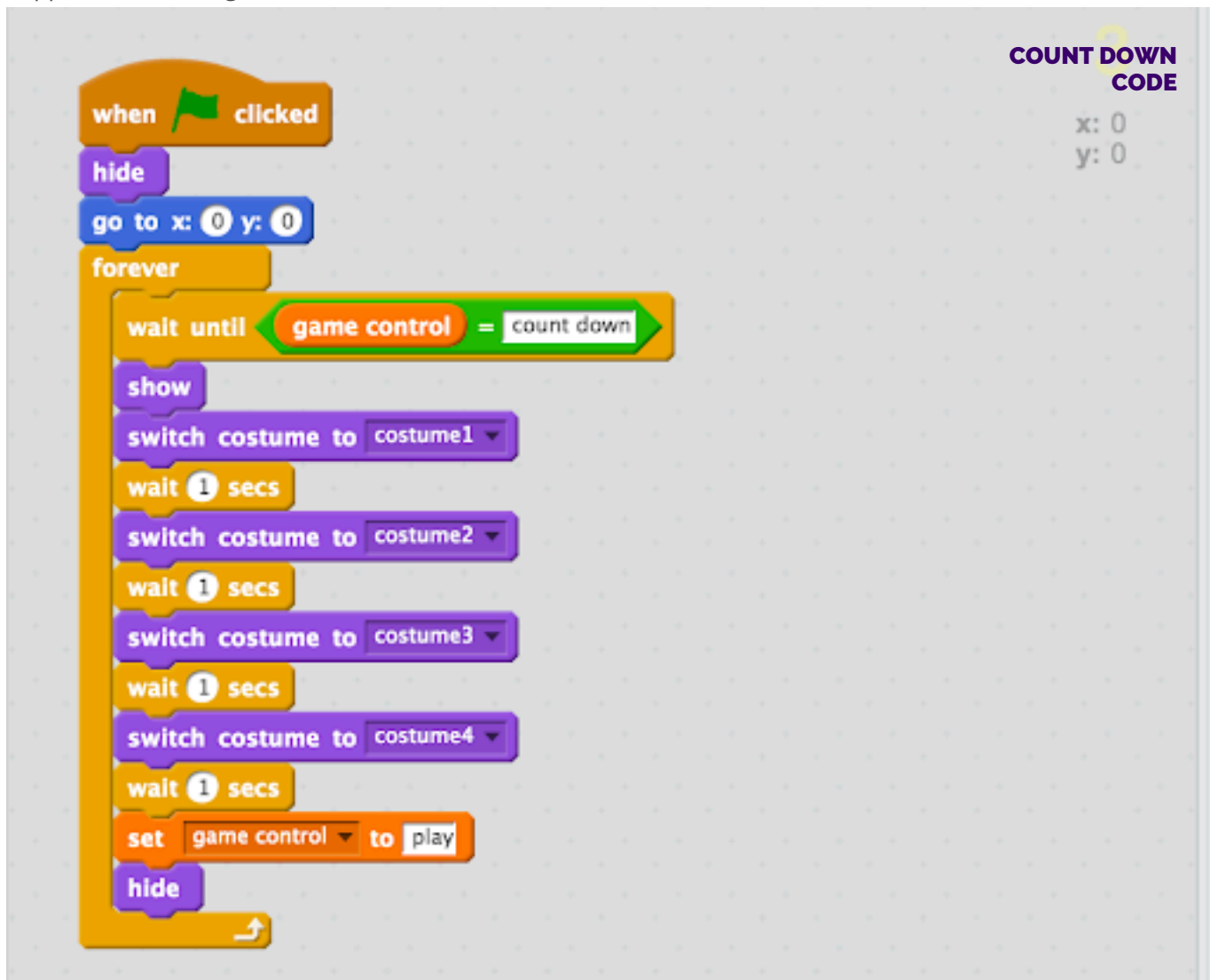
But the names of the backdrops are not very intuitive so lets rename them to **instructions**, **maze**, **win**, **lose**.



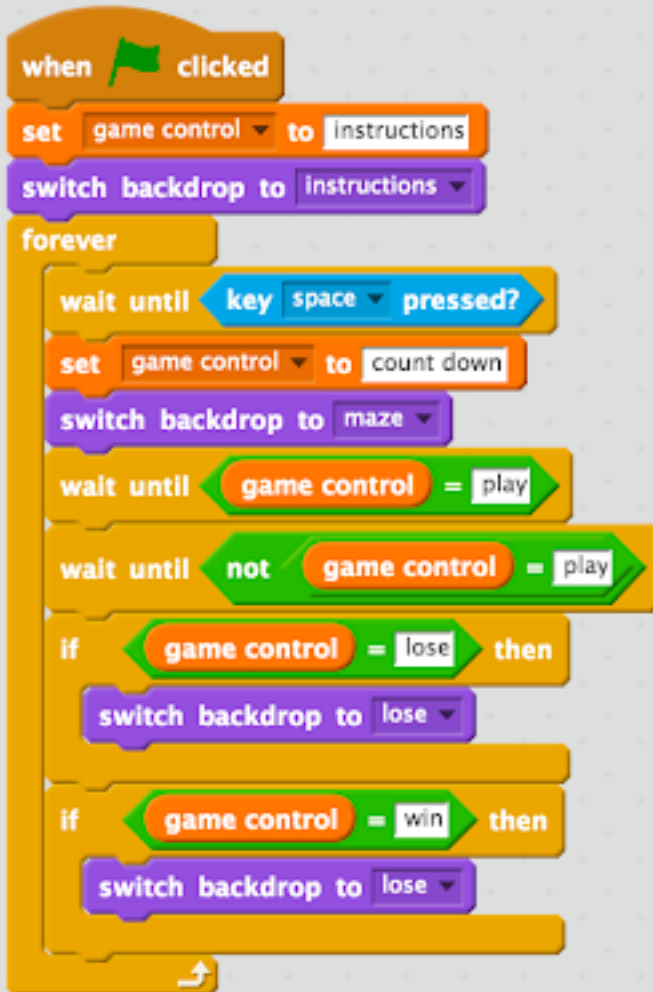
We want the player to be able to start the game, so we are going to add the code to switch the **backdrop** on space bar click.



Awesome! Now switch over to the **count down sprite** and add the code to make the countdown happen to start the game.



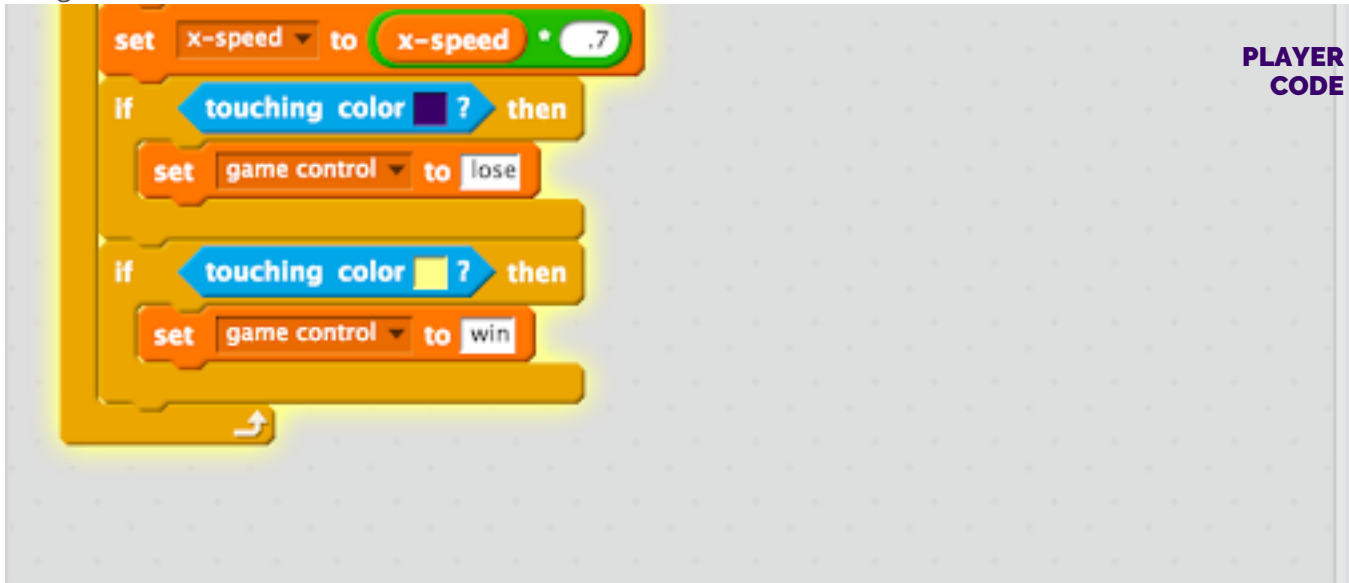
Back to the **Backdrops**, where we want to display the backdrops, based upon whether the player wins or loses

**BACKDROP  
CODE**

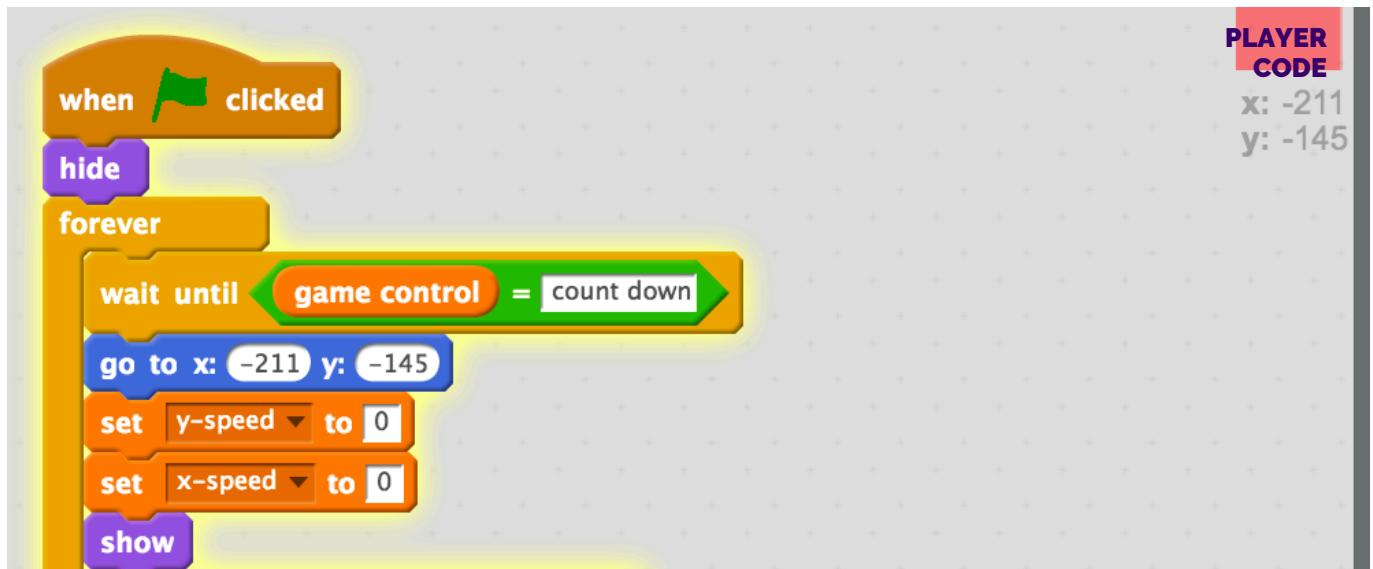
# TEST

to make sure that your countdown is working and that you can start moving after the countdown.

Now we want to move onto making sure the game ends when the **player** hits the wall or the flag.



Now we are going to hide the **player** until the game begins and appear during the countdown. Additionally we want to position the player to the start of the maze.



Not done yet! We need to add the player control to a variable to play when the game starts.  
Here is the final code!

The image displays two sections of Scratch code. The left section is a large script starting with 'when green flag clicked', followed by 'hide', a 'forever' loop containing 'wait until game control = count down', 'go to x: -211 y: -145', 'set y-speed to 0', 'set x-speed to 0', 'show', 'wait until game control = play', and a 'repeat until not game control = play' loop with four 'if' statements for arrow key presses (up, down, right, left) that change y-speed and x-speed by 1 or -1. The right section is a smaller script starting with 'change y by y-speed', 'change x by x-speed', 'set y-speed to y-speed \* .7', 'set x-speed to x-speed \* .7', two 'if touching color' blocks (red and yellow) that set 'game control' to 'lose' or 'win', and ends with 'hide'.

# FINAL PLAYER CODE

It has to match!


**TEST**  
to make sure that  
you can win and  
lose

GAME DESIGN

# COMPLETED

We are going to set the score to be how many seconds it takes a player to beat the maze! We are creating a timer that increases the score every second.


**BACKDROP CODE**



```

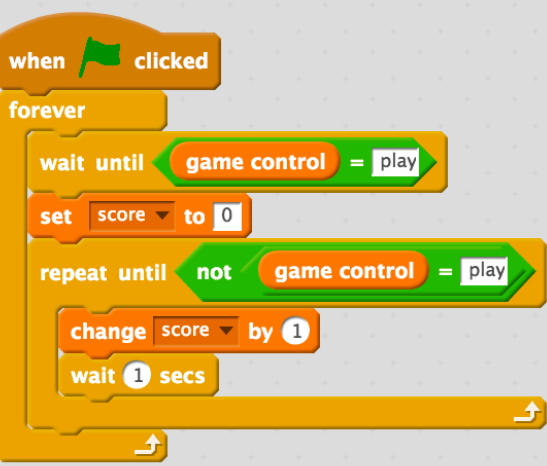
when green flag clicked
  forever loop
    wait until game control = play
    set score to 0
    repeat until not game control = play
      change score by 1
      wait 1 secs
  
```

Now we add code to hide and show the score on the right screen.



```

when green flag clicked
  set game control to Instructions
  switch backdrop to Instructions
  hide variable score
  forever loop
    wait until key space pressed?
    set game control to count down
    hide variable score
    switch backdrop to maze
    wait until game control = play
    wait until not game control = play
    if game control = lose then
      switch backdrop to lose
      play sound gong
    if game control = win then
      switch backdrop to win
      show variable score
      play sound water drop
  
```



```

when green flag clicked
  forever loop
    wait until game control = play
    set score to 0
    repeat until not game control = play
      change score by 1
      wait 1 secs
  
```



ENJOY YOUR

# COMPLETED GAME

**onus points for adding sound!**