Final!!

The first idea for this project was to make a movement system for a character, and then just have it be a walking simulator/exploration game demo with no gameplay. However, as it was having a hard time running on my laptop, I scrapped it and remade the dinosaur game that you play on Google without wifi.

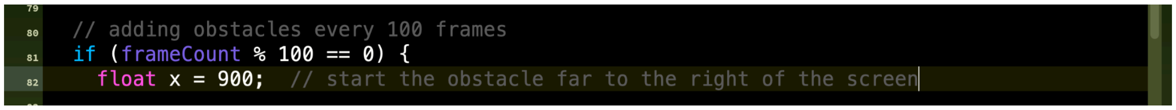
I also initially planned on making different sprites for when the character rectangle (named Lin) jumped or ducked, but dealing with collisions and hitboxes for the sprites was more labor-intensive than I first thought and therefore needed to be scrapped temporarily (will update over winter break!!).

So I essentially made the Google dinosaur game with an added "ducking" mechanic and a different theme planned for the future. The topObstacle and bottomObstacle classes are child/extended classes of the parent/base Obstacle class so they all have the same collision systems, but "spawn" at different positions – and, in the case of topObstacle, occasionally change shapes.

The first obstacle with this code was the scrolling background. I tried to do something similar to the Asteroids in-class exercise with rectangles, but I also had a background prepared already. I made a way to make the background scroll similar to the Asteroids in-class exercise. This made my computer run *extremely* slow, so I had to ask for help on how graphics buffers work. A kind stranger on Reddit helped me out with the actual code and it works!! The background scrolls and gives the illusion of movement. Yay!!!

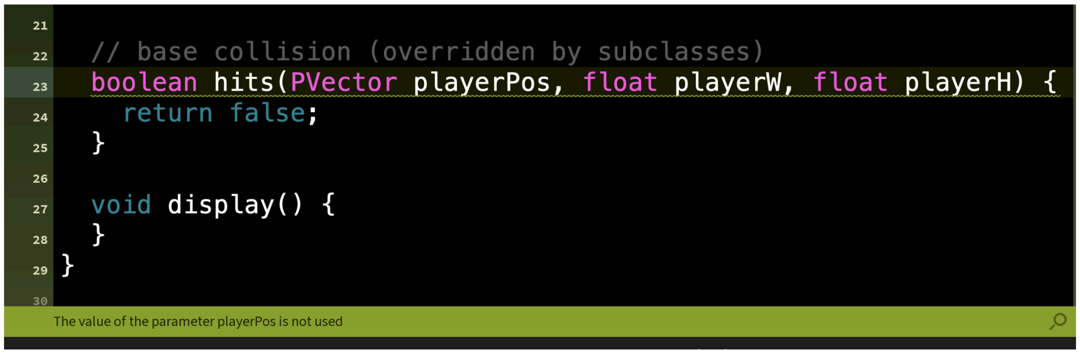
In terms of errors, these are two that stuck out.

The first issue wasn't technically an error, but it was interesting because I don't know exactly why it didn't work. I wanted the obstacles to spawn off the screen, so I set the variable to "width + 300." However, it didn't work; the obstacles kept spawning in the center-ish area of the screen. Just to be safe, I set it to "manually" spawn the obstacles at x 900 even though it isn't exactly resize safe (not like I plan on making this resized in the future, but I wish it worked the first way). Here's it fixed:



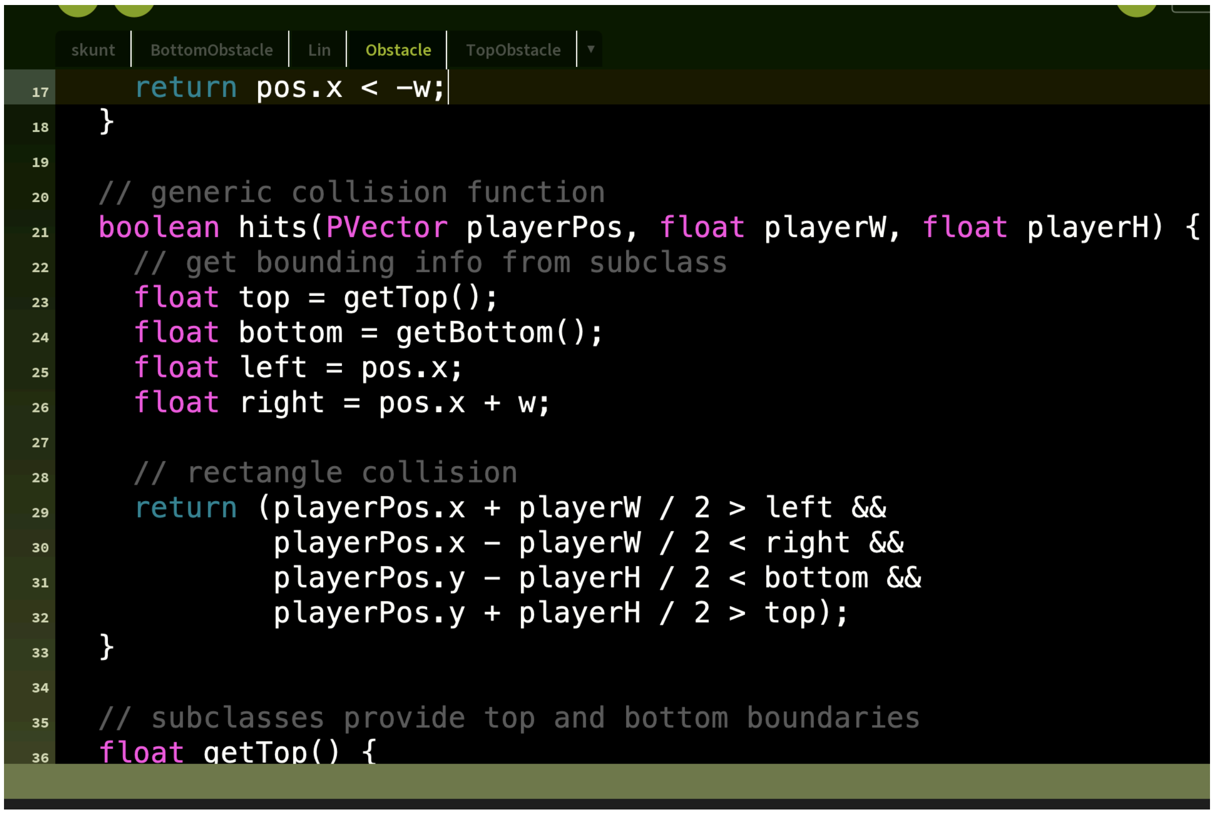
```
// adding obstacles every 100 frames
if (frameCount % 100 == 0) {
  float x = 900;  // start the obstacle far to the right of the screen
```

The second one isn't really an error, it was technically *an* error – had little squiggly lines under the code – but it still ran. It was related to the collision system I had, and I'm assuming the computer was just upset that I never used the playerPos parameter, but it could run just fine even though I never explicitly used that parameter. It also stopped working when I took the parameter out, so I brute-force-fixed the code (also thanks to the kind stranger) to use the playerPos parameter just to get the little squiggly error line out so it looked a little nicer.



```

  // base collision (overridden by subclasses)
  boolean hits(PVector playerPos, float playerW, float playerH) {
    return false;
  }

  void display() {
  }
}
```

The value of the parameter playerPos is not used

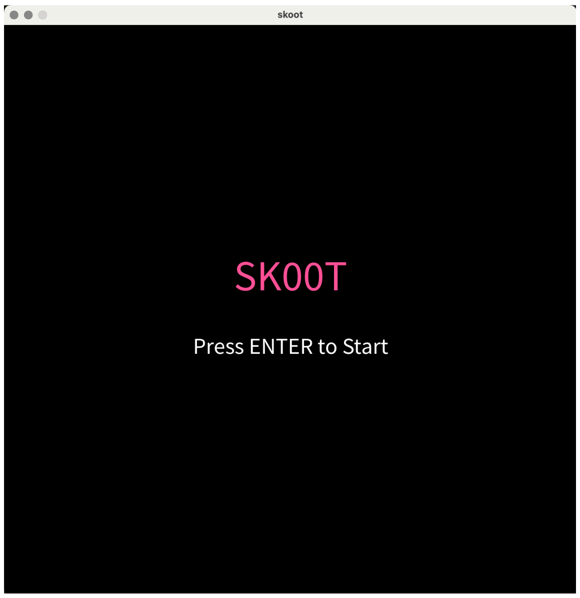


```
    return pos.x < -w;
  }

  // generic collision function
  boolean hits(PVector playerPos, float playerW, float playerH) {
    // get bounding info from subclass
    float top = getTop();
    float bottom = getBottom();
    float left = pos.x;
    float right = pos.x + w;

    // rectangle collision
    return (playerPos.x + playerW / 2 > left &&
            playerPos.x - playerW / 2 < right &&
            playerPos.y - playerH / 2 < bottom &&
            playerPos.y + playerH / 2 > top);
  }

  // subclasses provide top and bottom boundaries
  float getTop() {
```

After fixing those issues, going back and renaming some variables, and fixing the subclasses, I have a working demo for this!! I named the game Skoot because it's like Skate but only in theme, and it basically uses the same gameplay mechanic that the dinosaur Google game uses. Depending on the score you get when you get a game over, the little comment says different things. I've enjoyed playing with the speed and jump variables on my own, and am excited to possibly make this more developed over time just as a side-project.

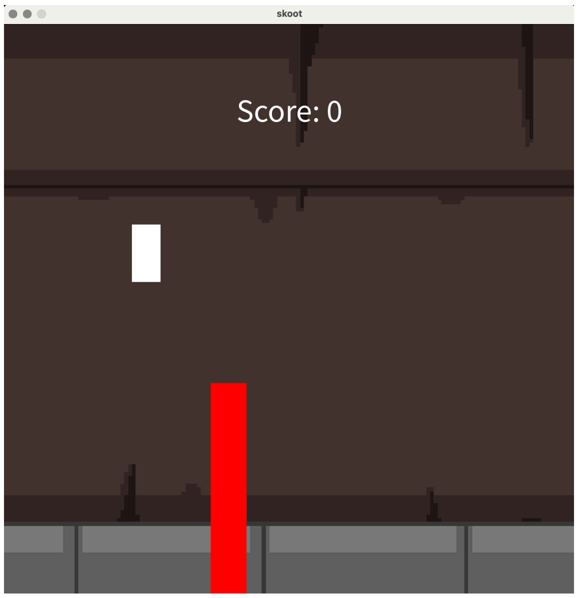
Hopefully over this upcoming winter/later spring break I'll make the differing topObstacle shapes, all the bottomObstacle shapes, and also Lin textured with graphics and make the program run smoothly.

Below are some screenshots of what I have currently, very excited to work on this more in the future!!

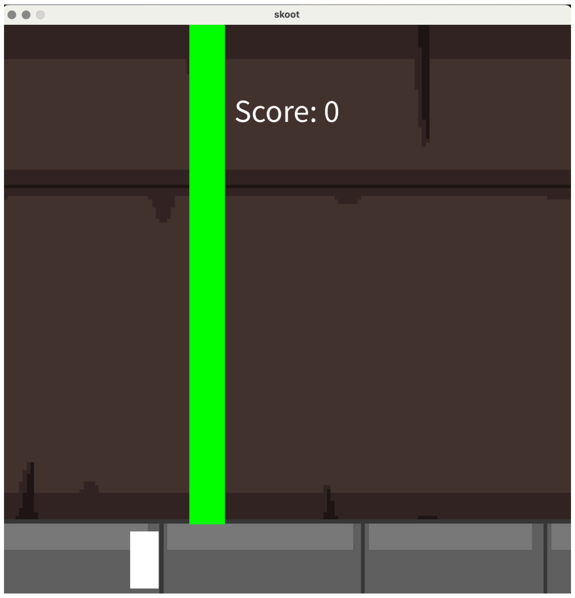


SKOOT

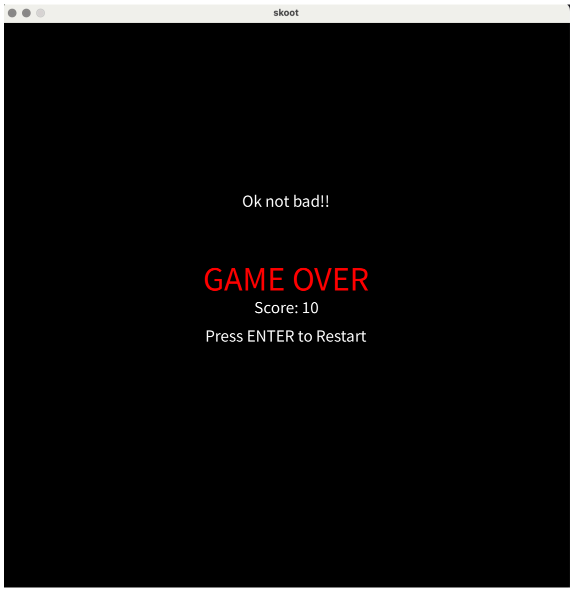
Press ENTER to Start



Score: 0



Score: 0

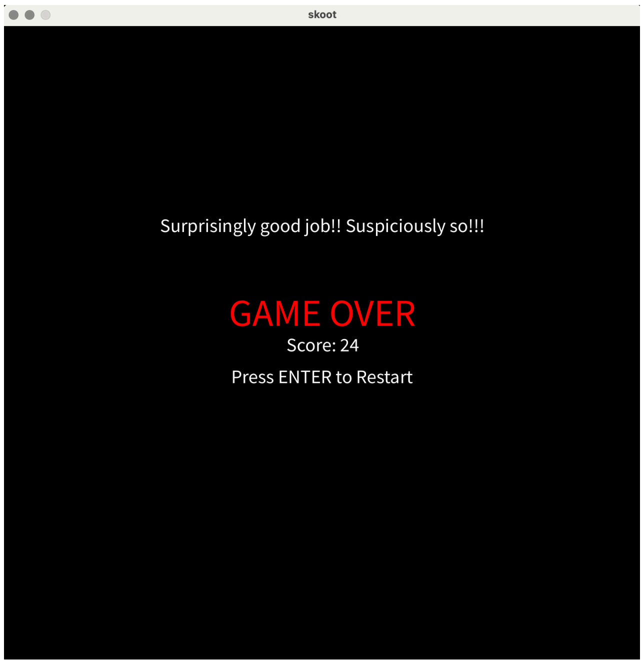


Ok not bad!!

GAME OVER

Score: 10

Press ENTER to Restart



Surprisingly good job!! Suspiciously so!!!

GAME OVER

Score: 24

Press ENTER to Restart