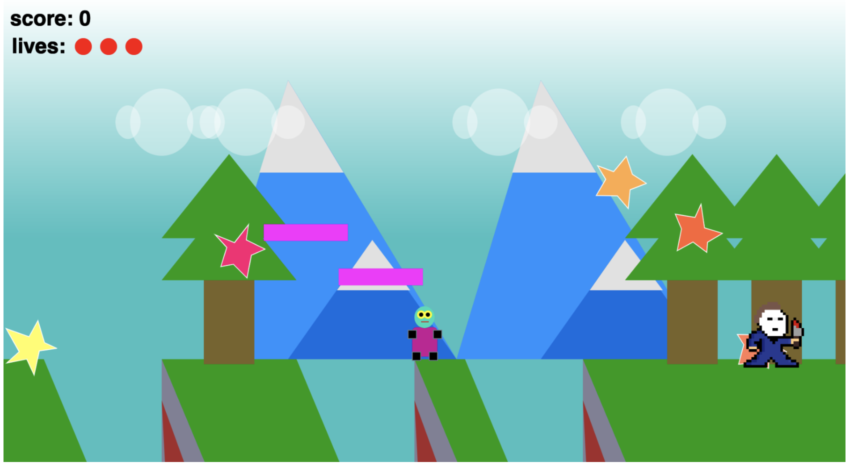
Alexandre Vieira

Subject: Introduction to Programming I
User_id: dd7be94b19b121f06f3cf4db221d09a3

Game Project

March 08, 2023

Game Overview



The game is a college assignment where I used p5.js to create the scenario, character, mechanics, and other elements to interact with. I explored some tools learned and added some bits of creativity. Some extra mechanics were added, like the flying mode, and random platforms and enemies' positions. I bet the randomness and new fly mode will be the killing points of this game. Hope you enjoy it!

Implement Extensions

I added some new elements that were not required for the completion of the game, just for the sake of practice, and to make the game unique.

1. **Fly mode (new mechanics)**: Based on the high level of difficulty the randomness brings to the game, I added a function to fly, using double jump. By simply modifying the enemies' and platforms' positions.

2. **Enemy’s speed increases when the score goes up (new mechanics):** The enemy gets obsessed with your progress and will move faster when you get more collectibles. Watch out!
3. **Randomness**: To add more the game more challenging, I added these features to **enemies** and **platforms**, so every time the game starts, you will have a unique scenario to play. I did not use a random position for the collectible since it could be in a strange position, like over the canyon.
4. **Added movement to the flagpole and collectibles**: To catch the player’s eye, the stars are frenetically moving and the flagpole moves as if there is a wind blowing it.
5. **Sound and atmosphere**:
    a. Added *eerie atmospheric* music to the background.
    b. Implemented *footsteps* while walking in the x-axis, once the character stops, it triggers a **sound.stop()** method. I really liked learning this part.
    c. *Different dying sounds*: Added different sounds for dying events. Falling into the canyon and letting the enemy catch you will have personalized sounds.
6. **Assets**: List of assets used in this project.

Disclaimer: I did not create anyof these assets. All copyrights belong to their respective artists and creators.

a. **Ambient (sound)**: "Fantasy Ambience" by David Fesliyan". (This piece of music is not subject to our regular policy. This loop is 100% copyright free and does not require any attribution). https://www.fesliyanstudios.com/royalty-free-music/download/fantasy-ambience/1702

b. **DieSound** (sound): 2-oof - meme sound. Internet free use. Ref.: https://www.myinstants.com/pt/instant/roblox-oof/

c. **Footsteps** (sound):

d. **Game Over** (sound):

e. **Collectibles** (sound): 3-hehe boy - meme sound. Internet free use. :https://www.myinstants.com/pt/instant/op-hehehe-boy/

f. **Jump** (sound): p5.js sound library - https://p5js.org/libraries/

g. Enemy (image): artist: rrockstarrr, posted on https://www.deviantart.com/rrockstarrr/art/Michael-Myers-Halloween-1978-pixelated-771370205

h. Enemy killing sound: Free resources from myInstant.com. Ref: https://www.myinstants.com/en/instant/stationary-kill-among-us-11888/

i. 4-oh my god wow - meme sound. Internet free use. https://www.myinstants.com/en/instant/anime-wow/

Skills learned/practiced in this project

Best Practices: Coding using best practices was hard at the beginning, but as I practiced more, it became natural, and actually helped a lot when editing the next implementations. Especially when we had to add platforms using constructor functions along with loops.

Object Oriented: It was a great opportunity to learn how to create objects, and access and call functions to build them in **draw()**. I think this knowledge goes far beyond what we learned in the class, and I can even see myself using it in a future project.

Difficult parts

- **Project Structure:** Understanding the basic functions that run the game was hard to understand. Sometimes I was not sure if I need to add code to draw(), setup(), or in a particular function, outside the ring. But, as the course progressed, I could understand the variations we could make and the role of each function can be intrinsic, private, or global, as well as the variables.
- **Factory Pattern**: It was hard to understand the abstraction of this technique. I got very lost when I saw we can use a draw() method within a factory function. It turns out that it was very useful to replicate elements with quite a flexibility, but I had to watch the classes a couple of times to get the basic concept.
- **Constructor functions:** Understanding how to use `**this**` was very hard, and I could see the point of making a new element with **new NewElement**, but it took me a while to understand. It was used less in the game, but I still need to review and make it more fluid.