

Evelyn Zheng
15-112
Kirby: Nightmare in Dreamland

Project Description [2.5 pts]: The name of the term project and a short description of what it will be.

Term Project Name: Kirby: Nightmare in Dreamland

Description:

This rendition of the real game Kirby: Nightmare in Dreamland will entail an extended obstacle course that gets harder as it progresses with an increasing number of challenges. It will end in an epic boss fight between Kirby and an AI driven enemy.

Similar projects [2.5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.

One similar project I have seen is from the gallery and it is called "Super Mario Side Scroller." Super Mario is quite similar to Kirby as it involves a scrolling background and a main player that is controlled by the user. It also involves features such as sprites that interact with the mario by following it around and a point system. My project will be similar to this one as it will involve enemies with varying capabilities such as flying, shocking, etc. Both projects will have a boss level fight. The difference is that Kirby is able to consume other sprites, shoot out puffs of attacks, and the graphics are different. I hope to make a long level progressively harder as the sprite approaches the boss.

Structural Plan [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or classes.

The finalized project will be very animation heavy, so it will likely involve event functions corresponding to many keys and graphics functions that we have used thus far for homeworks such as that for Tetris. It will also include classes for types of enemies and the subclasses will correspond to qualities unique to categories of enemies. For example, the super class will be "Enemy," but a subclass could be "Shooter."

Algorithmic Plan [2.5 pts]: A plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most difficult, and include some details of how you expect to implement these features.

The trickiest part of my project will likely be the floating feature of Kirby as well as the boss fight at the end of the game. I believe I can approach the implementation of the

floating by using some basic physics equations involving projectile motion to model the pathway of each jump. I will approach the boss fight by coding the enemy to make moves based on proximity to the user's Kirby. This can be done with if statements and providing the enemy with abilities such as charging, which is a sudden speed up at the Kirby. This will make it so that the enemy is self-controlled.

Timeline Plan [2.5 pts]: A timeline for when you intend to complete the major features of the project.

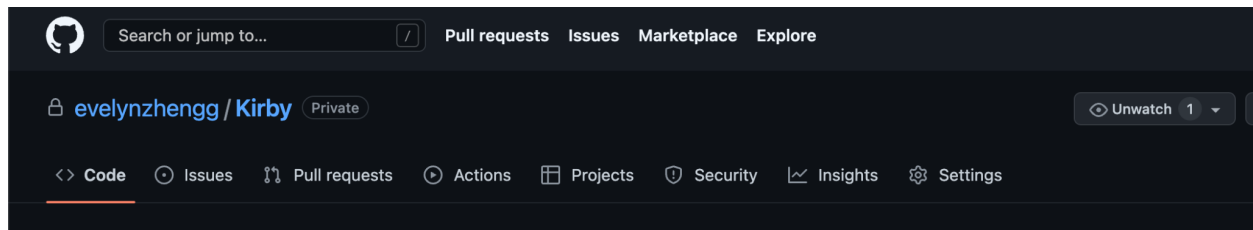
11/20: Layout of background and all sprites drawn

11/28: All abilities of sprites coded and functional obstacle course

12/5: Boss fight and project basically completed

Version Control Plan [1.5 pts]:

I will backup this code on Github.



Module List [1 pts]:

N/A

Storyboard [5 pts]

In separate file

TP1 Updates:

For TP1, I was able to implement some side scrolling, Kirby eating, Kirby attacking, and enemy patrolling. I also incorporated a game loss state as well as a visual representation of Kirby's lives. I am still working on including projectile motion, jumping, and the inheritance of abilities. My Kirby project will likely have the same speed throughout the obstacle course with up to 5 sprites other than the kirby. 3 of those sprites will be patrolling enemies that walk around, 1 of those sprites (meta knight) will have an ability that kirby can acquire, and the last sprite will be the boss. The sprites will walk around randomly while the boss will follow Kirby around when close in proximity.