**Prancing Pete Documentation**

By Zach Brown & Nathan Stoeckle

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Changes**

Multiple changes were made during our design process, which ultimately led to the result we have currently as a fully fleshed out game. We both changed a lot within our game’s main code, including creating a levelFactory file which makes it simplistic to make multiple levels within the game. We also separated each piece of the file with more efficiency and reprogrammed the game as a whole, as the old version we created was very inefficient, especially main.js which we didn’t use app.main which caused bugs. We also added js files for organization and extra features, including a timer file for the game timer which counts from 30 seconds to 0 seconds, a images file, which stores sprites, pressing P pauses the game, and changed up our main.js file.

**Requirements**

My partner and I fulfilled the requirements of the assignment and tried to make our code as efficient as possible. Earlier in the project, our levelFactory was a bunch of for loops that were about 300+ lines, however, we decided to use case statements to shorten the code to 83 lines, which made our levels also easier to program if we want to addon to the project in the future. We have the media requirements. We do not have mouse controls, but we have keyboard controls, as well as this controls being precise, which I believe we fulfilled these as well. We also used various files in our game and we had 3 levels to show various difficulties (easy (1), medium (2), hard (3)). Has a title screen, main game screen, game over screen, and a game over screen for if you win the game. This is a separate statement, so this can be considered a 4th screen. Our game experience has depth, however, I wish we could add more levels. Your choices do matter because if you leave the character for 30 seconds, you get a Game Over and lose, so it’s essential to beat the level for progression. Our code did not use any external libraries except the Boomshine code that we used for our Homework assignment (not really a library). We did our best to follow D.R.Y. but there were some things I wish I could change including making main.

**What went wrong**

In our project at first, a lot went wrong. We couldn’t figure out how to implement our sprites into the project, couldn’t get the sound working correctly at one point and at one point the game just wasn’t playable. Despite these obstacles, we figured out how to implement the core features that we wanted in the game. If we had more time with the project, I’d add a few more levels, however, we decided not to due to time constraints and a ton of bugs once we tried to implement them. We also wanted to focus on functionality and quality, rather than the quantity of the levels to try and earn as many points as possible and make a polished game. The timer was also difficult to implement into our code, but we made a timer.js file to get it done.

**Noncourse resources**

We didn’t use too many non mycourses resources. The only thing significant we followed was a simple how to make a simple platformer tutorial online and after that we free reigned the whole project. We didn’t even use that tutorial in the end either, as we reprogrammed the entire project to fit our needs. It was a tremendous task, but luckily with the partner I had, we both managed to do a significant amount of work and get it done.

**Contributions**

My partner Nathan Stoeckle and I worked very well together. This included Nathan making levels in levelFactory.js, and reprogramming main.js, while I worked on implementing sprites into the game, the basic code structure, and audio/other files that were necessary for the project as well as the two of us debugging.

**Grade Eachother**

My Grade: 90%

Nathan’s Grade: 90%

We both worked the same amount on the project, I believe we both deserve the same grade due to that. We also had different roles throughout the project including Nathan making the levels, and filling in the structure of main while I worked on separating and creating levelFactory, implementing sprites, and fixing bugs/audio.