Sebastian Beltran, Christopher Cenci Wednesday, October 10, 2018

Written Documentation

Our concept entailed a simple game where the user must combine timing, skill and luck. The game involves the user having three rotating balls and targets appearing randomly on the screen. The user must then aim the correct colored ball at the matching target. When the user hits the correct target, they score a certain number of points, the player has 15 red balls, 10 yellow balls and 5 blue balls. The fewer the balls available, the more points they are worth if they hit the right target. Blue balls are worth 500 points, yellow balls are worth 200 and red balls are worth a 100.

The purpose in the interaction was simply to create an enjoyable yet challenging game for the player. The player can decide on what ball to play at what time depending on his strategy. Although there could have been more dynamics to the game, however, the time that took us to familiarize ourselves with JavaScript concepts and the PixiJS library that we used did not leave us enough room to make it more complex.

We decided on this game from our initial idea of a game where you shoot fireworks and watch them explode. From there, we decided to create a more interactive experience with the game and player. Our interactivity in this project involved the player actually having to focus to hit the targets, if the timing and focus is not right, then the chances of hitting targets are significantly lower.

In our project, we used the JavaScript library PIXI to facilitate our creation of the game. PIXI is a rendering engine which facilitates rendering our sprites into the game. The targets and balls are sprites and the library made it easy for us to put in the sprites and use them with collision detection. The library also has built in functions which helped us manipulate the sprites to ease our design choices within the game. One function for example, was pivot and anchor, pivot and anchor made it so that the sprite rotates on a certain point, that is how we manipulated it so that the triangle rotates around the balls. The rotation in particular was made simpler by taking out the hassle of rotating the canvas and saving states. Other functions in the PIXI library aided us with the little details, when hovering over the balls, the mouse changes,

the text display, PIXI gives you the ability to separate the code into three parts, the setup, play and functions loops, which helped us keep organized. We used the beginner's tutorial on PIXI's website (https:// github.com/kittykatattack/learningPixi) as a reference to get us started. Because the visual aspects of the game were rather simple, we created our own images in Photoshop.

We divided the project so that the work load was balanced. The division of the tasks is as follows:

Sebastian Beltran:

- Brainstorming
- Research of library
- Handling the Targets' set up and updates
- Collision detection
- Points
- Game Over
- Clean up

Chris Cenci:

- Brainstorming
- Research of library
- Handling Balls' creation and display
- Handling user click of the balls
- Written Documentation
- Storyboard