Work Journal

May 3rd, 2019: (Time Spent: 4 hours)

The first day I started to program my game - Ninja Mission. The first thing I tried to accomplish is to create a simulation of a bouncing ball. At first, I have managed to display an image onto the screen and made the image to move down. However, when the image was moving down, a series of images were being displayed and there was no separation between them. As a result, I felt annoyed. So, I looked up the “Intro to Animation” program and figured out the problem. Apparently, I was missing the “windowSurface.fill (BLACK) statement. Afterward, I felt relieved. Then, I tried to make the image to bounce on the screen. Initially, I set up a value for gravity, the horizontal speed and the vertical speed. It turns out, the horizontal movement of the image is perfectly fine. However, the image was bouncing at the same height. Consequently, I was a little frustrated. After another of hour of thinking and researching, I figured out a solution; Instead of multiplying the vertical speed by -1 when the image hits the bottom screen, multiply it by -0.97.  So that when the image bounce upward, it will not get back to its original height. Therefore, the height of the image will gradually decrease as it bounces. In the end, I felt satisfy and excited for the next step - create a Ninja character, makes it move along the bottom of the screens and allows it to shot out weapon.

May 10th, 2019: (Time Spent: 4 hours)

Alright! My goal for today was to create the character animation. At first, I did some research online to see what is the best way to display a series of different pictures on the screen while a key is being pressed. After an hour of researching, I came across this brilliant idea where I just have to store the images into the two lists. One list is for the right movements and the other list is for the left movements. So, I went back online to look up some ninja sprite sheets. Shortly after, I found the one that I desire. Then, I used photoshop to edit to sprite sheet. Finally, I started to code and managed to create the character animation. During this whole process, there wasn’t any frustrating bugs or problems. So, I felt really “smooth” and pleasant.

May 13th, 2019: (Time Spent: 4 hours)

Today was a bit stressful. I was planning to get the Ninja to shoot out its weapon whenever the spacebar is pressed. However, the weapon (ninja star) couldn’t be properly displayed on the screen. When the spacebar is pressed, the image of the ninja star would appear and travel upward. But while it is moving upward, the image seems to “glitch”. So, I asked Mrs. Barlow for help, she suggested that in order to solve my problem, I have to write my game in the form class and use sprites groups. After school, I went home and try fix my game. However, I still could not get the bullet to work. As a result, I felt quite upset. Hoping to solve it by tomorrow.

May 14th, 2019: (Time Spent: 2 hours)

Today in class, I got the shooting animation to work by looking up the Game Class program. The adding food algorithm gave me a big hint. Once I discovered the problem and get my bullets to work, I felt rejuvenated from yesterday’s disappointment. In addition, I have changed the background for my game and it looks awesome!

May 16th, 2019: (Time Spent: 2 hours)

Today in class, I got the collision between the ninja and the bouncy balls to work. Since my game is already written in the form of class and I have used the sprite function, checking for collision between two sprite groups was fairly easy. So, I was really glad that I got the collision to work.

May 17th, 2019: (Time Spent: 2 hours)

Today in class, I realized there should be some sort of reaction when the ninja gets hit by a bouncy ball. So, I saved the dead image of the ninja from online. After an hour of coding, I could not it to work properly and felt extremely annoyed. Whenever the ninja gets hit by a bouncy ball, its dead image will be displayed, but it will not change back to its regular movement images when left/right arrow key is pressed. So, I ask Mrs. Barlow for help, Mrs. Barlow helped me to come up with an algorithm by writing the details and logics on a piece of paper. In the end, Mrs. Barlow and I debug the problem. I felt very thankful

May 21nd, 2019: (Time Spent: 1 hours)

Today in class, I have got the collision between the ninja stars and the bouncy balls to work properly. I have also made the original bouncy balls to split into smaller balls once they got hit by the ninja stars. All I have to do was to put the collisions into a list. Then, I use a for loop to go through the list to check if the ball that got hit is one of the original balls. If it is, I generate two smaller balls from that exact location and make them to move at different directions. I felt very content about this unique feature of my game.

May 23rd, 2019: (Time Spent: 0.5 hours)

Today, I have decided to split up the original bouncy balls into smaller balls when they hit the ninja. So, I did almost the same thing as yesterday, except I had to check for the collision between the bouncy balls and the ninja character. No surprise today, so my feelings were quite calm.

May 27th, 2019: (Time Spent: 1 hour)

Today, I have started to create the second level of my game. In this level, my plan was to have three walls (or barriers) that divide the screen into four little sessions. It will also leave some space at the bottom for the ninja to move around. Each session will contain a single ball that bounces around. So, I went online to search for some purple walls that will fit my background and I displayed them onto the screen. Today was an easy day for me, but I am still quite happy that I found the wall that I desire.

May 28th 2019: (Time Spent: 3 hours)

Today, I have worked on the functionality of my second level. So, I check for the collision between the balls and the walls. Whenever, a ball hits a wall, it will bounce or changes its direction. However, I have encountered a problem. The problem was that when the balls hit the bottoms of the walls, it will not bounce back down, instead, it will continue to go up. As result, I felt really frustrated. After an hour of thinking and trying, I have solved the problem by keeping track of the vertical movement (or position) of the bouncy balls. So, whenever the balls get to the bottom of the walls, I multiply its vertical speed by -1. Therefore, balls’ direction will change and enable them to bounce down when they hit the bottom of the walls. In the end, I felt relived.

May 29th, 2019: (Time Spent: 3 hours)

Today, I have started to work on the third level of my game. In this level, my plan was to have a boss at the top of my screen, moving horizontally and shooting out two bouncy balls at a time. For the design (or look) of my boss, I asked my friend - Ethan, for help since he is amazing at photoshop. Just within 10 minutes, Ethan and I have created a desirable image for the boss. I felt confident about this level. After one hour of coding, I got the third level to work. However, I have realized that the boss was shooting out way too many balls and they fill up the screen way too quickly. So, I was a little annoyed and tried to come up with a solution. In the end, I decided to add some if statements to limit the amount of balls that appear on the screen. Overall, I was very satisfied with the look and functionality of my third level.

May 30th, 2019: (Time Spent: 3 hours)

Today, I played my semi-finished game and acknowledged its difficulty. I was worry about the fact that people will not beat my game. So, I came up with an idea and that is to allow the player to gain health (or lives) by shooting down the bouncy balls. So, whenever a ball is destroyed or split up into two smaller balls, there will be 20% chance of dropping down a free life. In order to pick up that life, the player just have to walk (collide) into that life image and its lives will automatically go up by one. After 1.5 hours of coding, I implemented the “HealthPack” class and created a new sprite group. I played my game couple more times and was very glad about that fact that it is a lot more beatable. In addition to this new feature, I have also added a health bar to my boss. Adding the health bar to the boss in level 3 was a bit challenging, I was not able to figured it out by myself and felt a bit upset. So, I decided to search up online. In the end, I realized the health bar algorithm is not as challenging as I thought and I managed to get it working.

June 3rd, 2019: (Time Spent: 2 hours)

Today, I have set up my instruction menu and tried to add a blinking text to my menu. First, I struggled to come up with an accurate algorithm for the text to blink at regular time interval. I felt quite worried. So, I asked Mrs. Barlow for help. Unfortunately, she was not able to come up with an algorithm immediately. But, it’s alright. I moved on to create my instruction menu. I found an image for the background and displayed it onto the screen. So that was fairly easily, except I had to add some if and elif statement to check for user input. Today, I felt a bit disappointed since I did not get the blinking text to work, but I was glad that I found a desirable background image for my instruction page.

June 4th, 2019: (Time Spent: 2 hours)

Today, Mrs. Barlow gave me an amazing algorithm for blinking text. So, I added to my program and now it works perfectly. I felt extremely thankful for Mrs. Barlow’s help. Beside this good news, I have also added some music and sound effects to my game. However, the music and sound effects could not be muted or played properly. Once again, I felt frustrated. So, I took a look at The Game Class program and realized the error; I should’ve changed the value of the Boolean variable inside the while loop instead of putting it outside. Also, I should’ve used the “not” keyword instead of just changing the value of the Boolean variable to “False”. After I got my music and sound effects to work properly, the frustration was gone and I felt great!

June 5th, 2019: (Time Spent: 1.5 hour)

Today, I have found a bug in my game. The bug was that when the player already has a maximum of 5 lives, the free lives will still drop out from the bouncy balls when the player destroyed them. Consequently, I was quite nervous. Then, I took a look at my program and added in some print statements. I realized it was just a silly mistake, I should’ve done < 5 instead of <= 5 in one of the if statements. After I fixed this bug, my tension was relived. Then, I searched up more music and sound effects for my game. I added them to my game, which really enhanced the overall experience my game and I was genuinely please about these extra features.

June 7th, 2019: (Time spent: 1.5 hour)

Today, I have designed some background images for my menu, instruction, mission failed and mission complete pages by using photoshop. Also, I searched up some keyboard clipart for my instruction page. Moreover, I added more comments to my program and created some functions to get ridded of the duplicated codes. Finally, I finished my game, and was extremely content about the game as it is fun to play and it matches the initial game that I wanted to create.