Yvette Williamson and Lara Disuanco
Rong Zhang
CAP3032
April 23, 2019

"Smoothie Warrior" Game: Final Report

Work Distribution

.pde	Yvette Williamson	Lara Disuanco
smoothie	Implemented powerups, bomb and fruit displays, and score keeper	Implemented images, buttons, time keeper, and sound effects
Bomb		Implemented Bomb class
Fruit	Implemented Fruit class	
Knife		Implemented Knife class
Power_ups	Implemented Power_ups	
Sound_effects		Implemented Sound class

Instruction Manual

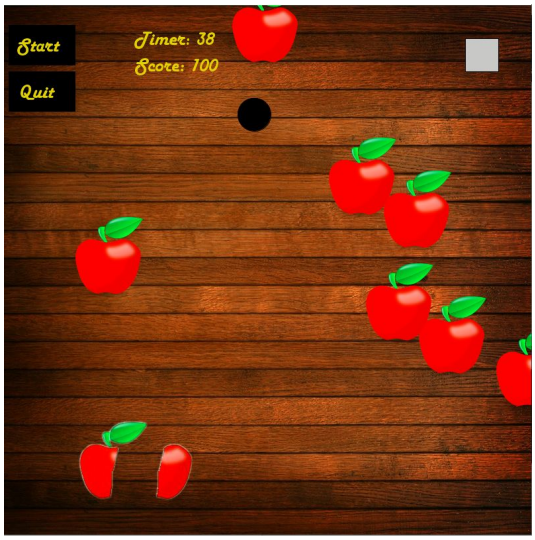
Summary of How to Play

The goal of this game is to slice as many fruit as you can as they are thrown through the air in the time allotted. However, you must do this without slicing any bombs. If a bomb is touched, the game is over. When a player gets a certain score, a power up will be unlocked and the user will be able to utilize it in the game.

The Beginning of the Game

The image below shows the first screen that will be displayed to the player when the program runs. The buttons on the left will be displayed on the entire time of the gameplay so the player will have the option to start the game or quit the game, and also which level they will be playing, either level 1 or level 2.

Levels



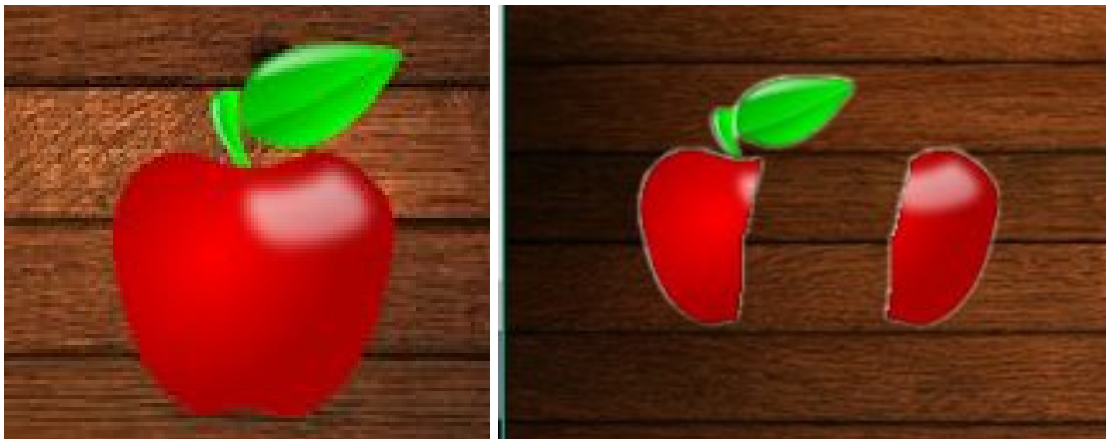
Each level will have this same look with the same gameplay as shown in the image above. Each level will be 1 minute long; the player must survive that minute to move on to the next level. The fruit, shown as an apple, will appear on screen as if it was being thrown up in the air in front of the wooden background. There will also be bombs that will be “thrown” in as well. The player should avoid “slicing” the bombs. The

score will be displayed at the top of the screen, where each slice will give different amounts in points. The start and quit buttons will be displayed on the left that the player can click on. The square on the right is the indicator for the player to display which power ups can be used. The different colors to signify when certain power ups can be used are: green, red, and blue. Check KeyBoard section for explanations of colors.

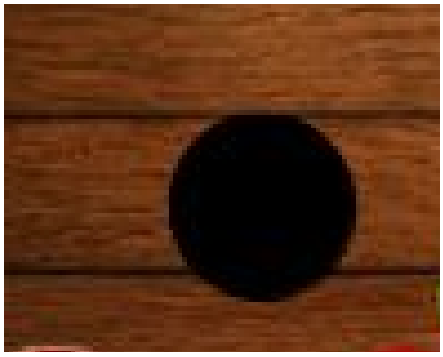
Mouse Control

The player will use the mouse as the “knife” to slice the fruit. You must press down on the mouse, drag it over the fruit, and release the mouse press for a “slice” to be registered.

The image below is how the cursor of the mouse will be displayed during the game.



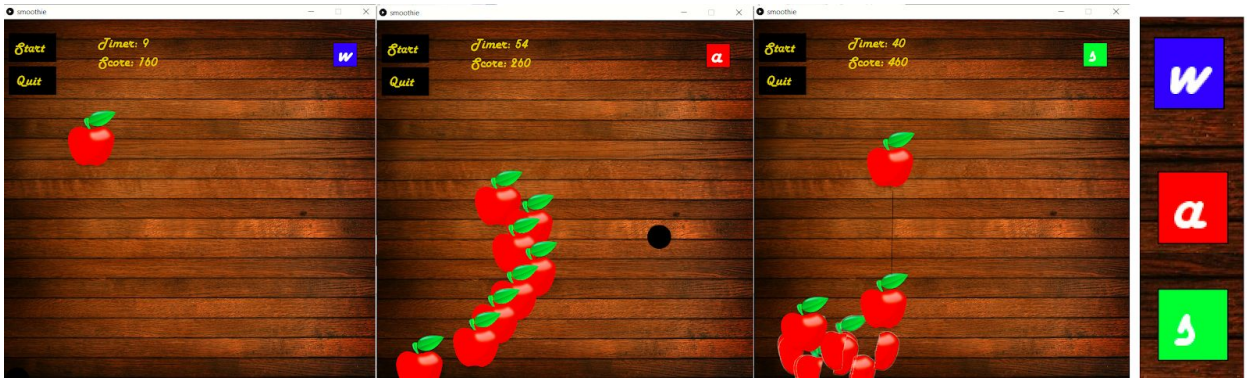
The images above display the before and after of the “slicing” of the fruit the player controls using the cursor.



The image above displays the bomb. The player should avoid clicking and dragging on the bombs as the game will end as soon as a bomb is “sliced.”

KeyBoard

The w,a,s keys on the keyboard will be used for power ups to trigger different play modes. As the score goes above certain numbers, the indicator square will let the player know when power ups are ready to use. The “w” key will be for the green power up, increasing the amount of fruit. The “a” key will be for the red power up, removing bombs. The “s” key will be for the blue power up, slowing down time to slice more precisely. Once the key for the corresponding power up is pressed, then the player has activated their power up.



Difficulties Encountered

Lara Disuanco

I faced a few difficulties with the bombs, the buttons, and the sound effects. I had a hard time understanding how to implement multiple bombs going up in the air at different positions and how to get the position of every single bomb in order to calculate its' distance from the mouse. Creating an array of bombs and using the get() function was extremely helpful and something I learned from my partner! The buttons weren't initially a difficult thing because I was hoping to implement ControlP5 buttons that I had previously used in another class. However, I was having trouble setting the buttons' visibilities at certain points, and it was just simpler creating rectangles and setting parameters for them like we learned in class. Originally, I used the Sound Library, but it kept giving a warning about latency, but later I discovered in class that the Sound Library does not like .mp3 files, so our professor suggested using the Minim library instead and it worked! Overall, I didn't have too many problems because my partner was really understanding and patient with me on this project!

Yvette Williamson

For this project, the difficulties I encountered were mainly in the implementation of the objects. For setting the bomb and fruit arraylists, I had trouble figuring how I could reset or clear the list once I reached a set value. I did not want to keep adding objects to the arraylist as that would result in too many objects still in the arraylists that would not be displayed on screen. Eventually I figured out how to clear the arraylist after a certain number using the clear function and a variable to keep track of the number of objects. The other difficulty I had was setting random trajectories for the bombs and fruits. I had to make sure that I continually got a random number, so I had to keep moving the random variable initialization around until I got it correct. Another difficulty I encountered was having the objects show up on screen at different times. I used the frame count and a variable with random values being set to it to implement this. The next difficulty I had was setting up and implementing the powerups. I had to make sure the power ups would be displayed after a certain amount of points, then implement the change each power up brought. It was complicated, but I figured it out using booleans and integers. Overall, the difficulties I found were fixed easily and my partner helped me with figuring out what those fixes could be and how I could implement them.