Team Mooncakes
15 October 2018
CMPE 135
Assignment #5

Assignment #5 Report

Unfortunately we were not able to get wxwidgets working on any of our systems. The difficulty of installing and getting the libraries to work on xcodes took too much time and we were not able to successfully install resulted in not get our code running. The following paragraphs would explain how we would have implemented our code from assignment 4 and answered all the question of assignment 5 involving wxwidgets.

- What events does your game application generate?

The events that our game generates are the various button events including new game, exit, and set name. It also includes the events to select each choice such as rock, paper, or scissors. These are all triggered by click events when a button is pressed. We also generate titles and different headings to toggle a prompt event. This will show the player who won and lost the most matches. These are triggered after the code executes once after the button is pressed. There is also one more event for the exit. We close out the application when pressing the exit button which is an exit event. We also include the about event where it tells the player the information about the game and how it works. There is also a function to allow the player to change the maximum number of games instead of defaulting the games to 20 games. In doing so there is a save event where a player inputs a number and we save that number inside of our code to change the number of games.

- How did you use CALLBACK functions to handle the events?

The way that we handled the callback functions were after a button was press. When a button is press, it executes our lines of code that would compare, save, and confirm the player's responses as well as the computers. This way we would be able to handle the callback functions with the same logic that we used during assignment 4. We also used some callback functions to save values that were input by the player to our memory. When done, we saved both the players selection as well as the number of games that the player input prior to the game starting.

- How were you able to reuse code from assignment #4 now that you have inversion of control?

When executing each scenario, we reused the logic used in our player class, and computer classes to work with the events on a click. So when a button is clicked, the line of code runs to execute much like when we used a command line to select 1, 2, or 3. This allows the user to enter data into the interface with a much better aesthetic feel. We also changed all of the cout statements into statements that would run with the labels on the application to print the prompts for the player. When generating the results, after the final button has been press, the app will report the results to the player via the labels. This will provide a more user friendly designated area showing all the results of the game.