Nathan Quinn

Prof. Jerkofsky

CS 201

16 April 2019

Project

My project is based around the game: League of Legends. In this game a player can get kills, deaths, and assists over the game. With kills and assists a player earns gold which they can spend on items. Items improve a player’s character, so a player is highly incentivized to buy as many items as they can afford. The items are limited to six for each player players have to carefully chose which items are best for their character while being able to afford them. I wanted to create a project that would help players look back on past games and see meaningful statistics to help them improve their gameplay.

The main menu form is the first form a user will see of my project. There is a list box that displays previously recorded games that can be selected to see their statistics for that game on another form. There is also a menu strip that allows the user to close the project, add a new game, and display a message box with a brief description of the project. At first there was also going to be a way to delete previously recorded games. I decided not to add one since predefined arrays are used to store a games data a user is limited to five games. This with the fact once the main menu form is closed the recorded game data is lost, I did not see much use for a delete option of previously recorded games.

The new game form is where the game data is inputted by the user. This includes a nickname for the game, kills deaths, assists and the items from a game. The nickname will be used as an item for the list box on the main menu to identify the recorded game. The user will type out the nickname, kills, deaths, and assists. Those values will be evaluated to make sure they are the specified data type. The user will pick the six items they could have bought over the course of the game. There is also the option “None” if they did not get six items. All of the data provided by the user will be passed on as string to the “RecordAndCalculateGame” module.

In the module it will insert the data into a predefined string array. It can store up to five games like previously mentioned and it does this by using a variable to keep track of how many games have been recorded. From there the rest of the subroutines in the module are used by the game stats form. The game stats form displays the data provided previously provided about the game, the kills per death, assists per death, gold earned, and gold spent on items. It gets the previously recorded game data by using the selected index from the list box on the main menu form which has been stored in a global variable to get the appropriate array from the module. The game stats form gets kills per death, assists per death, and gold earned by sending the specific cells of the string array containing the relevant data to functions in the module. The module converts the provided string data into the data type to be returned and performs the calculation. It returns the result which is converted and formatted back into string to be displayed on the game stats form. For getting the gold spent on items for a specific game the game array is sent to the module where a subset containing the items is used. It loops through subset using the stored strings to determine how much gold was spent. The amount of gold spent is then returned to the game stats form to be displayed. It is important to note that a user could select items that in total come out to more gold than they earned. This is because in the game itself a player cannot buy an item they cannot afford. It also allows a user to see how much gold they would need to afford a particular set of items.

Overall, I feel this demonstrates a fair amount of what I have learned over this course. Various controls are used throughout the different forms which show an understanding of when it is appropriate to use certain controls and how to use them. Various datatypes are used and converted like string, boolean, integer, and double. I also use arrays to store related data. I use conditional statements effectively in all of the forms and the module. The use of loops to loop through arrays. I use multiple procedures and functions to perform various tasks and pass data between forms. Multiple forms and a module are used together to make a functioning program. With everything mentioned above I feel it shows a good knowledge of what we have learned in the course and how to use it to make real world applications.