Group name: Git Em

Tracy Chea

Zee Hussain

Nicholas Nguyen

**Basic description:** We created a program to host quizzes for users. The program was coded in python. When first logging in, the game asks what question file the user wants to use. The game then asks for the player to enter their name, and the name of any additional players. Users will then be able to choose from a variety of different types of questions such as: multiple choice, True/False, user input, etc. The winner is determined by which player has the most points at the end of the game.

**Major Changes:** The user is able to input multiple expenses. Each user has their own account and can add expenses to their expense sheet. The expense sheet is written onto an excel sheet, which will then be read to be displayed in a GUI. The buttons were padded for spacing, and the functionalities of writing expenses and logging in were embellished.

**User Stories:**

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Description | Estimated Time | Estimated Difficulty |
| Web Application | Set up our application to be able to be accessed by multiple users | 3 days | 7/10 |
| Users | Each user is able to be created and their answers recorded | 1 day | 5/10 |
| Questions | Users are able to type their questions or upload an excel file with own questions | 3 days | 7/10 |
| Display | Questions are able to be displayed as well as users’ answers and score | 6 days | 6/10 |
| Points | Point deduction/accumulation functionality to keep track of scores | 3 days | 5/10 |
| End Game | Game ends when all questions asked/ goal point met | 4 days | 4/10 |
| Display Winner | Game correctly shows winner | 3 days | 2/10 |
| Timer | Game will only accept answers from players after timer goes off | 3 days | 6/10 |
| Format | Game format will change based on user preference | 3 days | 5/10 |
| Computer Player | Create a way to allow computer players to select answers | 3 days | 7/10 |

**Accomplishments:** We established all core functionalities that we were hoping to. Players are able to add multiple players or play against a computer player. Players can use their own questions, and display them in game. Points are rewarded and kept track of per player. Once questions have run out, the game ends, and the winner is displayed.

**Alterations/changes from release #1 and #2:**

In release #1, the user was not able to use their own questions and were limited to our own excel file. The option to add multiple players also did not exist, and the player was limited to playing against the computer. We have now implemented multiple players and all of the attributes that go along with that change, like declaring the player with the most points the winner. The game also prompts the user to enter the name of the excel file that they wish to use for their questions.

**Things Added/Dropped for Final Release:**

We were unable to create the web application portion that we intended. None of our members have ever used python for web development, and it proved to be very strenuous and a large time sink to figure out how to implement things. We spent a large amount of time, and even tried switching to javascript temporarily, but ultimately decided it was best to cut our losses and stick with python.

**Pair Programming Chart:**

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Driver | Navigator | Completion |
| Web Application | Tracy | Zee | No |
| Users | Nick | Tracy | Yes |
| Questions | Tracy | Nick | Yesi |
| Display | Zee | Nick | No |
| Points | Tracy | Zee | Yes |
| End Game | Nick | Tracy | Yes |
| Display Winner | Zee | Nick | Yes |
| Computer player | Tracy | Nick | Yes |

**Reflection:**

We underestimated the scope of this project and overestimated our ability to figure out an entirely new web development framework . The amount of time we spent attempting to complete the web application portion of our game detracted from the rest of it. This caused major delays and as a result, we were not able to implement it, opting to have a less clean, but finished product. Moving forward, our group will make sure to stick to languages and frameworks that we have full confidence in, so we can make sure to meet our goals that we set in our proposal. Functionality wise, the project was a success and is completely playable. We have fulfilled most of the goals that we set for ourselves, and are excited to apply the knowledge we gained in future projects.