**Scientific Computing Assignment 2**

**F60C03**

Chen Wen Kang 024552

**Assignment 2 Question 2**

**<Try Your Luck> Game**

**Problem Description and Motivation**

The aim of this project is to create a pseudo-blackjack card game that is suitable and easy to play for children or teenagers between the age of 10 to 17.

The idea first came fruition after an example in the lecture involving a function that can accept input in the graph plotting interface (ginput). The example provided an idea that it may be possible to create a user interface that is interactive using only graph plotting functions.

The original intention was to recreate Blackjack in MATLAB. However, due to the complexity of the rules of Blackjack, it has a high learning curve for beginners as some of the concept concerning the types of card (ex.: spade, hearts.) and some special combinations (ex.: An ‘A’ and a card with a value of 10 means an instant win.) which are relatively hard to convey without a few matches of game played. It was decided that a simpler game that resembles Blackjack in playstyle but is easier to play and doesn’t lose the strategic aspect of the original Blackjack game should be made.

The game was titled simply as <Try Your Luck>, and its game interface is made entirely using only built-in MATLAB graph plotting function. A few built-in MATLAB graphic user interface function was used to ensure a better navigability for the players to choose the difficulty, view their high score and to view the instructions if they do not understand how the game is played.

The player start with 3 lives. At the start of the game, the player was given a fixed number of cards depending on the difficulty level. Each round, the player can choose either to reveal a card of forfeit the round if desired. If the player chooses to reveal a card, the computer’s card will also be revealed and the player can decide their next move depending on the card that is revealed. If all cards are revealed, the total sum of all the player’s card and the computer’s card will be compared. If the player’s total sum is greater then the player’s score is added. If the computer’s score is higher however, the player’s life will be deducted by 1. If the player chooses to forfeit the match, all the cards will be shown, the player will not be penalized but no score is added either.

**Previous Work**

There are multiply MATLAB programmed Blackjack games out there on the internet with different approaches in doing a Blackjack game, using different algorithms for the shuffling of cards to ensure fairness is maintained and for the dealer’s AI (Dealer must know how to deal) when playing the cards. It should also be noted that most Blackjack games that are done using MATLAB require the use of third party packages (toolboxes) and GUI for optimal user experiences.

Despite the number of MATLAB programmed Blackjack games on the internet, there were no simplified version or similar card game to this current one. There were also no Blackjack games that are created using purely graph plotting functions only.

Although there are a lot of Blackjack examples out on the internet, the belief that a simplified version will do better when exposed to children or teenagers between the age of 10 to 17 was held. This is because, as mentioned in the above section, Blackjack have a higher learning curve for the player to master and might pose confusing for first timers and beginners. It is also originally designed for teenagers and adults instead of a younger audience as it contains a gambling element. Thus, a game that kept some element of Blackjack but does not contain the complex bits of Blackjack was designed and created.

Blackjack is played with poker cards. It determines a win or lose condition based on the combination of the cards and sum of the cards on one’s hand. The player will try to get a sum of number as close to 21 with specific combinations. The player shall try to obtain a higher total sum than the host. If any side got a total sum of more than 21 they will ‘bust’, which will make them lose. All the face cards (ex. King, Queen) has a value of 10 and the ‘A’ card have a value of 1 or 11. (Moler, 2012)

The comparing of numbers in Blackjack are taken as the main idea of this game. In <Try You Luck>, the cards range from 1 to 10 and each side are given 4 cards. The player opens the card one by one and try to make sure they get a total sum of numbers that is more than the computer’s total sum. However, if at any point in the game before all cards are revealed that the player feel as if they are at a disadvantage and risk losing, they can choose to forfeit the match, which will reveal all cards and end the match without losing or winning the match. There is no currency figure involved in the game but only lives and scores. If the player loses a life will be deducted. If the player wins 100 points will be added. The players are only given 3 chance (lives) before the games ends.

**Requirements**

This game was created using only built in MATLAB function without the use of any external libraries.

First the plotting function were researched to understand how each of the function works. After searching for the MATLAB documentation, it was decided that the ginput function, rectangle function, clf function and the text function are the most essential for the creation of the game interface.

For the creation of the card entrance animation, clf function and rectangle function are the most used function. However, to reduce the refresh speed to a point where the animation could be seen, the pause function was used to slow down the iteration speed of the for loop.

The functions involving graph manipulation and plotting are the most used functions in the game interface especially in the creations of the card animations.

The method of passing arguments and returning arguments form functions to functions are also researched to understand how to enhance the communication between functions without using a global variable that might reduce the security of the program. This information was obtained by looking at different discussions on MATLAB forums.

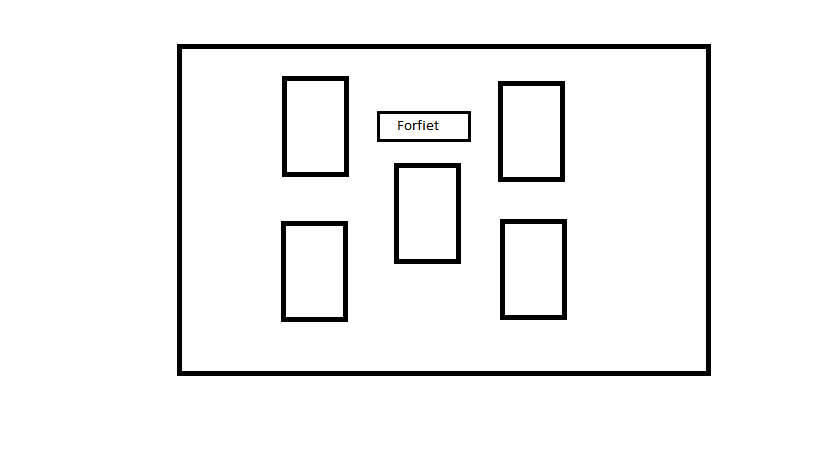
The MATLAB functions for random number generation, randi is used to generate random numbers that lie between 1 to 10. This is used to generate both the computer’s card number and the player’s card number to ensure fairness in the game. This information was obtained from the official MATLAB documentation which contains various function for random number generation. The decision was set on randi function because it generates an integer within a scope which was exactly what was needed.

The function for file manipulation are also researched via reading the MATLAB documentations to create the save high score and read high score functions.

**Design**

A skeleton of the game is first typed before everything starts to provide a blueprint on where the game is going.

The designing process of the game revolves a lot around the main game interface instead of miscellaneous details such as the main menu and the high score feature. Thus, the focus of this section will be revolving around the design of the game interface.



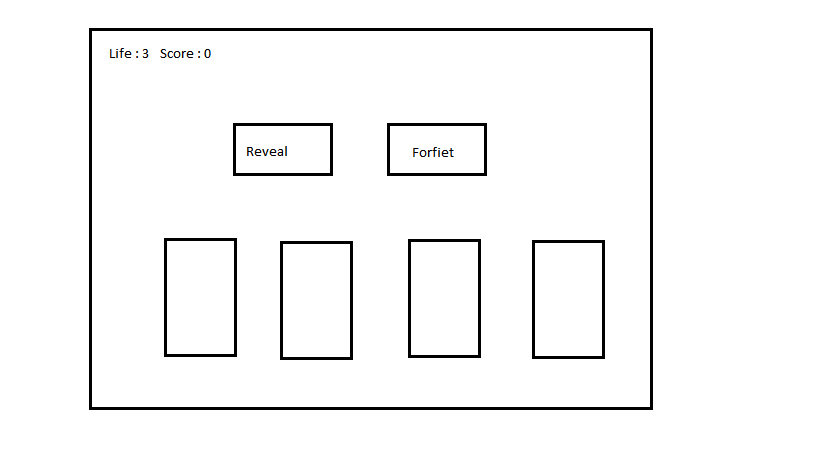
Stage 1: Starting Design

In the 1st stage, most of the functions are place aside in favor of the main game function. The structure of the program was getting drafted.

In this stage, the focus is on how the game interface should look and feel to the players. This stage involves solidifying of game concept, game rules and game flow. This stage also includes the development of the base game interface. Various designs were created and tested mainly for its user-friendliness.

The gameplay was temporarily designed to be a pseudo-number guessing game at this current stage. The solid idea is not yet established at this point. Players are allowed to open the cards in any order and must try to obtain a higher number than the randomly generated computer’s number. Player may choose to forfeit if they feel as if they cannot obtain a number higher than the computer’s generated number.

At first the idea was to create a starting animation that have the cards spread out from the middle deck. The idea of being able to click on any card to reveal the card was suggested. However, this proved to be counter intuitive and ineffective in conveying what the game is trying to achieve when the coding of the game proceeded. Most player, when presented with such interface express confusion and lost as to what they should do and what the game is about. The lack of instructions and pointers on the game interface itself caused the game to be very unattractive to the players. It should also be noted that at this point of time the concept of the game was not completely solidified yet. The idea for a forfeit button was considered near the end of this stage.

 Stage 2: Mid Stage

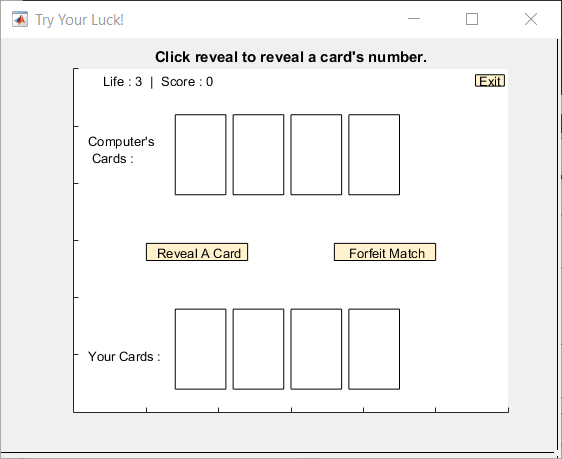
In the 2nd stage, the solid design of the game interface is already in place and the idea and flow of the game was decided and fixed. The animation was completed and the focus moves towards the logic and flow of the game. Other parts of the game such as the menu and instructions window are slightly touched on, but are not completed first.

The game interface now has a starting animation that have the cards slowly move downwards until a certain point where they stop and the game starts. The game interface is still relatively vague; however, it is more clear compared to its predecessor in the 1st stage. Now, instead of giving the player the freedom to click on any card at any given time to reveal the cards which was proven to be a flawed idea, that freedom is now denied and the cards are now revealed for them in a sequential order by pressing the ‘Reveal’ button. This might limit the player’s freedom, but the objective of the game becomes relatively clear and the interface becomes more user friendly. Compared to the first interface where the player might become unsure of what they should do after the start of the game, the interface now indicate that one of the 2 buttons should be pressed even without the aid of the instructions. It can be concluded that a simple trade of freedom of player actually yield a positive result in this case as a card game is a rule based game and freedom is not a necessary element that will aid the gameplay experience. The 2 buttons are positioned at the center to highlight its importance as it is one of the major elements in the gameplay.

Another important element that were added in the 2nd design was the addition of the life and score system. The life and score system is designed to keep the gameplay time short and non-continuously as this is a game designed as a short mini game that will be played during short free time. Players are given 3 lives at the start of the game. Each time the player wins a match, the score increments by 100. The life will be deducted each time the player loses a match. If the player loses 3 matches, the game ends and the score is saved if it exceeds the previous high score. The addition of the life and score system also resulted in the addition of the life and score display to remind the player about their score and their life.

The gameplay at this stage was solid: a simpler game that is blackjack-like but without its complex mechanisms and combinations. The player is shown a fixed amount of cards depending on the difficulty chosen. The player will try to guess the computer’s number and try to obtain a higher total sum than the computer’s generated number. Every round player is given a chance to reveal a card or forfeit the match. If the player feels that they are at a disadvantage, they can choose to forfeit which will end the match without any loss of life or increment of scores. The player loses if there is no life left.

At the end of the 2nd stage, all the function and/or modules are completely coded. Most of the function are coded using basic built-in MATLAB GUI function such as msgbox and menu. File processing functions were also used to create the save and read high score function. Player can navigate relatively easily around the menu using GUI. The amount of input checking was also reduced due to the use of GUI.



Stage 3: Final product

The 3rd Stage started after the completion of the game in the 2nd stage. The start of 3rd stage involves mainly code testing and bug catching. After fixing the bugs, feedbacks were taken from various testers to improve the game.

According to the testers’ feedback, a few major points of improvement was concluded. The game needed more interactions as only the player’s card was shown (the design in 2nd stage), and there were no indication or clue on the computer’s number in the entire process of playing until the last card is reveal or when the player chose to forfeit. This cause the game to be boring and unattractive. This also made the game a little bit too luck oriented, as player does not have any ways of predicting the computer’s number up until the end.

Thus, various improvement was made to the game. Some spaces were made for the display of the computer’s card. Now, in each round when the player chose to reveal a card, the computer will also reveal a card, which will provide a clue to the player to determine if the computer’s total sum of card number is greater or lesser than their potential total sum of card numbers. This change not only increases the interaction between the player and the computer, which makes the game less boring, it also made the game more skill oriented and more fair, which will add to the user experience.

The graph title feature was used to give player short instructions on what they can and should do. This further improve the user experience as most of the information that needed attention will be shown by the text that will appear at the top of the graph. At the end of the match, the title will also show the total sum of user’s card and the total sum of the computer’s card, then later tell the player who won the game before proceeding to the game over window which will show the player their total score.

The exit button was also added in this stage to allow the players to exit the game correctly as there were no other means to exit the game. To prevent accidental exits during the gaming session due to accidental clicks, a user-friendly feature, which requires the player to click once more to confirm the exit of the game was implemented into the gaming interface.

In the final stage of the game, various finishing touch were made to the game to increase clarity of the game. Labels were added on each side to indicate who the cards belong to. The buttons that can be clicked were also colored to highlight their importance in the gameplay. These few changes act as a pointer to the player on what they can click on, thus reducing confusion and improving user experience.

**Testing and Results**

|  |  |  |
| --- | --- | --- |
| Input | Output | Validity |
| Launch of the game | **C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.png** | Valid |
| Pressing ‘Instructions’ Button | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\instructions.png | Valid |
| Pressing ‘Highscore’ Button | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\highscore.png | Valid |
| Pressing ‘Start Game’ Button | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\start game diff.png | Valid |
| Choosing ‘Easy’ Difficulty | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\easy.png | Valid |
| Choosing ‘Intermediate’ Difficulty | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\normal.png | Valid |
| Choosing ‘Hard’ Difficulty | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\hard.png | Valid |
| Pressing ‘Reveal’ on the game interface | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\reveal.png | Valid |
| Pressing ‘Forfeit’ on the game interface | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\forfiet.png | Valid |
| Pressing ‘Exit’ on the game interface | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\exit.png | Valid |
| On Game Over | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gameover prompt.png | Valid |
| When new high score is reached | C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\highscore prompt.png  C:\Users\AlphonsusChen\AppData\Local\Microsoft\Windows\INetCache\Content.Word\name prompt.png | Valid |

**Conclusion**

From the process of creating this game, the impact of the utility and clarity that a Graphical User Interface (GUI) provides can be seen. A good GUI can give pointers to the user on what to do and what to click on without the user needing to read the instruction manual or the instruction. With a good GUI, the user can also navigate and find what they need easily, which will increase the user experience. A poorly design GUI on the other hand, can ruin the user’s experience severely.

For a better user experience, there must also be a degree of interaction between the interface and the user. The user must receive some sort of feedback when an input is fed into the interface. This will create an engaging interaction between the user and the interface. For example, when a button is clicked, a click sound is played and the button will show a clicked down animation to indicate that it has been clicked down by the user.

It could also be seen that MATLAB, as a programming language that specialized in matrix computation, can also be used as a tool to create other applications with relative simplicity without the need of third party dependencies. Compared to the other programming languages, MATLAB might appear dim as it is not a general-purpose programming language and might not possess some of the redeeming qualities of the other programming languages; However, when it comes to problem solving involving mathematical calculations and matrix manipulation, MATLAB is definitely a solid and ideal programming language.

**Reference**

Moler, C. (2012). *Simulating Blackjack with MATLAB*. Retrieved from https://www.mathworks.com/company/newsletters/articles/simulating-blackjack-with-matlab.html