Group Report: Higher or Lower

Group Name: Not Enough Input Arguments

Group members:
Aksel Akbermez (6511993)
Jasmine Saini (6513175)
Kam Yu Lam (6457073)
Milan Parmar (6416957)

I)Introduction

Our aim for this group project was to learn the fundamentals of coding through MATLAB while replicating a game which could be enjoyed by all ages. In our first group meeting we came up with several potential ideas, including recreations of Space Invaders, Pac-Man and a rhythm game. After looking at the basic logistics of these ideas and playing various versions of the games via the MATLAB File Exchange, we realised that these ideas were very common and had been produced many times. As a group we agreed that we wanted to do something unique; this is where the idea of a card game was proposed. We considered games similar to those of Poker and Blackjack but ,again, found these to be too common and so settled on a Higher or Lower card game.

II)How to run the code

To run the game, download the folder 'FinalGame' from the master branch on Github. All of the sound files and images are there for the game to work. This folder can be opened and then run from MATLAB. To start the game, run the file 'StartMenu2.m' which will open the first GUI. The start menu explains the rules of the game and allows the player to proceed to play.

III)How to play the game

The game itself is very simple. The player has to guess whether the next card that appears will be of a higher or lower value than the current card displayed. This is done by pressing either the 'Higher' or 'Lower' buttons on either side of the displayed Card. If guessed correctly, the box beneath the card will display the message 'YOU WIN (+$100)' and the players score in the top-right-corner will increase by $100. If guessed incorrectly, the box beneath the presented card will display ' YOU LOSE (-$100)' and the players score in the top-right-corner will decrease by $100.
The aim of the game is to obtain the highest score possible.

IV)Challenges Faced

As a group, we encountered more challenges during the development of our game than we had initially anticipated.

GUI Control

We decided to create our Higher or Lower Card game using MATLAB's Graphical user interface as it seemed the most suitable for our game. Having creating a rough outline of what we expected our game to look like, we experienced lag and flickering when attempting to run our GUI for the first time and eventually realised that this was down to us removing key parts from the skeleton of the GUI. As a result, the game had to be recreated several times from scratch so we could familiarise ourselves with MATLAB's GUI.

Generating random cards

Our initial layout of the game was five separate axes acting as cards and an image of a back of card layered on top of each axes. However, we found the inclusion of several different axis more challenging due to our lack of experience with MATLAB; we were unable to randomly generate different cards on each axis. Therefore, it was necessary to simplify the layout of the game and reduce our five axes to just one.

Using different variables such as 'randperm' and 'cla' we were able to clear the single axes repeatedly and generate a random new card in the place of the previously displayed card.

Assigning values to images and ranking them

This was the most important part of our game. We attempted to assign values to images but struggled to compare the card values as every time our GUI was run their values would not appear.

We tried several different ways to compare the value of the first card to the second. We thought we could identify previous cards and newly produced cards by counting the number of times the buttons had been pressed. We used a function to assign 'first card' as the card when 'handles.counter==0') and 'second card' as the card when 'handles.counter==1'). Moreover, we used 'ismember' to include the ranking and values of the cards so that we could simply compare the 'first card' and 'second card' variables. However, this did not achieve what we had hoped.

For our second attempt, we created three variables, 'current card', 'higher card' and 'lower card'. We created functions and used 'if', 'elseif' and 'else' to describe all the possible situations with the cards that we had. However, this caused problems; in order for the game to coordinate well, we needed to make sure that the local variables were consistent outside of our initial function. We then discovered that we could declare the variables as global.

We chose to give the image files numerical names such as '11.jpg', rather than 'Jack.jpg'. Using 'setGlobalCurrentCard(filenumber)' allowed us to store the file name into our global variable which gave it the value of its file name.

Scoreboard

The difficulties we encountered when working on creating a scoreboard for our game were not immediate but appeared after we made sufficient progress. Creating an edit text box that added and deducted one point after pressing 'Higher' or 'Lower' respectively was relatively simple. However, figuring out how to accumulate these points became an issue; each time we pressed 'Higher' or 'Lower' the scoreboard would switch between the values 1 and -1. Determining how to accumulate points was time consuming, nevertheless, we eventually learnt that to solve this we needed to create the global variable 'POINTS'. Using our variable 'POINTS' to then reference under the codes that triggered a win or loss allowed us to achieve a functioning scoreboard.

Background

Adding a background to our game was not a priority but we decided that we wanted to make the game more aesthetically pleasing. At first we tried to insert an axes across the whole GUI but found that the rest of our axis would shift around into an unusual layout. After looking into GUI backgrounds more, we realised that this problem could be easily resolved by directly manipulating the properties held by the background. Here we used ‘bg=imread(‘’)’ to import the background image and to prevent plotting over the background we turned the handlevisibility off. Finally, to make sure that background was behind all the other uicontrols we used ‘uistack(ah, ‘bottom’)’. Using this code allowed our background to be shown without interfering with any of our other axis.

V)Communication

Communication within the group

We created a group chat on Facebook to act as our main communication platform. We found it more convenient to use Messenger as opposed to Github when it came to sending simple text messages. Therefore, we appeared to not be very active on Github until we had to upload our final game.

Communication with the Lecturer

Over the course of the project we reached out to Dr. Mele at various stages. Our first meeting was arranged through Doodle where we had a Skype call to discuss our very first ideas and gain an insight on how each of these ideas would be tackled. We made full use of the Wednesday drop-in sessions whenever we felt like we needed some extra guidance. Github was the assigned platform used to communicate with Dr. Mele. It allowed us to open issues whenever we had a problem and Dr. Mele was always quick to respond and help us to the best of his ability.

Sharing Files

We attempted to mostly use Github for sharing all the different files of our project, such as images, audio and MATLAB files. However, there were times when we were not able to upload files from our accounts as we did not have the authorisation to edit the repository and this did delayed the speed of our progress. Therefore we resulted in emailing the files to one another. Eventually, we were able to use Github again after Dr. Mele had amended the authorisation of the repository.

To conclude, we have to admit that Github was not very user friendly for beginners.

VI) Results

As a group we are very proud of the final outcome for this project. Seeing our game develop from a very basic template to a working game is really gratifying.

Despite brief communication and coordination issues within our group we believe that the skills we have learnt from MATLAB will be transferable and of use in the future.

VII) Contribution Marks

Name	**Mark**
Aksel Akbermez	
Jasmine Saini	
Kam Yu Lam	
Milan Parmar	