Group ‘OllieJoeTomWill’ Project Report - Deal or no Deal

Members: • Ollie Wise • William Rennicks • Joachim Matheou • Thomas Talbert

Topic/Motivation:

We decided on the game show ‘Deal or no Deal’ for our group project topic. Drawing inspiration from a similar project (Who wants to be a Millionaire) completed last year. It was complex enough to apply various new techniques and skills that we’d learnt ourselves, yet still accommodate what we gained from the assignments. Above all, we chose ‘Deal or No Deal’ because once all the work was finished, there would be a fully-functioning engaging game for us to play and enjoy.

Methodology:

Since we were making a game we were keen to use Graphical User Interface (GUI). The first parts of building the game involved creating a function to randomize the boxes and a function to generate an offer from the banker. We had to build these into the GUI. Once the game was working a lot of time was spent customizing and experimenting to try and make the game aesthetically pleasing to play.

The Code:

The user runs HomeScreen.m, where a pushbuttons run a new GUI either instructions or the game. The player is asked to choose his box and then proceed to eliminate a number of boxes before the banker makes his offer. The randbox function randomises the boxes making sure that values are randomly assigned each time the game is played. The bankers offer takes the values of the boxes remaining and gives an offer as a random percentage between an upper and lower boundary. Clicking ‘Deal’ will allow the user to take home the banker’s offer, whereas ‘No Deal’ will continue the game until the next banker’s offer. This creates a realistic scenario, and a figure that isn’t just as simple as the mean of the boxes left. The deal pushbutton shows what you have won, what you could have won, and then a message depending on whether you beat the banker or not (the value of your box < your offer or vice versa). It shows what has been won, and what could have been won (had the user clicked ‘Deal’), with accompanying sound effects and specialised messages.

Challenges:

Throughout writing the code we encountered many challenges, a lot more than anticipated. One of the main and most time consuming challenges was learning how to use the GUI. Our understanding of the GUI developed as we made the game, to the extent that we might have done things differently now. Getting the banker to provide a “random” offer was challenging but very simple once we came up with a solution, this is key to making the game more interesting. A massive problem with the code we had to overcome was that if the player chose boxes too quickly the bankers offer wouldn’t come up. In order to to fix this we had to disable all the boxes within the bankofferGUI function. This was a tradeoff for practicality over appearance as the game does not look as good with the boxes being disabled and doesn’t run as nicely. Overcoming the compatibility between matlab on mac and windows was an unexpected problem we ran into when trying to check the code would work for you, library computers using matlab 2014a also wasn’t helpful.

Results/improvements:

Everyone in the group is very proud of the project, seeing it turn from a few ideas and drawings to a working game is extremely satisfying. If we had more time, there were a few ideas we were unable to implement into our game. Most of these involved using images to generate animations, we also tried including a gif. At the start of our project we wanted to be able to give the option to play the game again without closing the figure and re running but we couldn’t manage.

To run the game on a PC with windows, you simply need to download the whole zip file from the master branch. All of the sound files and images are there for the game to work seamlessly. It can then be copied into a folder of choice. This zip file can then be unzipped, and with all files open in the workspace, then be run from matlab itself . To start the game, proceed to running the homescreen.m file.

Marks:

Everyone contributed, but some had greater involvement in more difficult areas of the project.

Ollie Wise – 28%

William Rennicks – 26%

Joachim Matheou – 23%

Thomas Talbert – 23%