The Million Pound Drop Group Report

## Group participants

|  |  |
| --- | --- |
| Name | Contribution |
| Kristen Man | 19.8% |
| Tom Evans | 19.8% |
| George Ault | 1% |
| Cameron Pociecha | 19.8% |
| Acacia Amin | 19.8% |
| Umar Aziz | 19.8% |

## Motivation

The motivation to choose a quiz type game came after our initial group meeting. As a group, we had a mutual idea of creating some form of quiz show game. We began brainstorming different games and decided on creating ‘The Million Pound Drop’. This was due to the fact that we felt that we could implement the skills learned throughout the course of ‘Economic Analysis with Matrices’ in making the game from an idea into a reality. We also felt that this gameshow would allow us to put our own variation on the interface, such as the number of answers we wanted to have for each round and how we could make the game of increasing difficulty the further the player progresses. We noticed that a group had previously created this game and we felt confident in our abilities that we could create an improved variation using code. Due to some members in the group having never seen the show on television, we watched clips and episodes of the show in order for everyone to become familiar with the concept that we were going for, and also for everyone to be able to contribute on how we could adjust the game to make it unique to our group, whilst still also keeping the vital parts and layout of the show in our game.

## Methodology

Initially, we decided that the game would need a large database of questions, in order to allow the game to be played repeatedly without repetition of the questions and answers. Therefore we collected many questions with various different options. For each round, we decided that for each question, there should be 4 possible answers so that it fitted the image we wanted to use, as the image had 4 possible trapdoors on it. We also decided on having 4 possible answers as this is the number of options they have on the game show. For each round (1-4), we came up with 25 questions, thus allowing the game to be able to be played without repetition. The game works by downloading the entire zip file from the master branch and then opening it on Matlab. After this type 'start' into Matlab in the command window, where you will see a popup start screen we created for our game. We did this by creating a graphical user interface and then adding static text as well as push buttons on it. One of the buttons is '?', which when clicked on would open another pop up window showing the rules and how to play our game. Once you understand the game the back button returns you to the start screen. Upon starting the game you are able to see the round number as well as background audio, which is the theme tune of the game show on television so players would have more of a sense that they are playing 'The Million Pound Drop'. The number at the bottom of the screen tells you the total amount of money you have available to assign to the 4 possible answers. The game however, does not lead to an error if the player adds more than the money they assigned, so we are relying on the integrity of the player, playing as the rule state. After assigning your money, you press the 'drop' button, which causes the number at the bottom of the screen to change to the number, which matches the money you assigned to the correct answer. If you attached no money to the correct answer then you go directly to the endgame screen. This is because when the value in the static box is equal to 0, we coded for the endgame screen to appear. However, if you have money remaining you click the 'next' button and move on to the following round. This occurs 4 times as there are 4 rounds in our game. After clicking the 'next' button for the final time (provided you made it to the final round), you will be taken to the endgame screen where you will see your winnings in a static text box. We used the code 'global winnings' for this, so it shows the total amount of money you had remaining after playing the game. On the endgame screen there is also a button to return to the start screen so you can play again.

## Challenges

Throughout the process of making our game and writing the code, more issues arose than we had originally expected. The first challenge that we faced was understanding how the GUI (Graphical User Interface) works and how to create one specific to our project. Although this was very time-consuming, after researching some examples and watching some tutorials on it, we were able to grasp how it works. During the process of working on our game, we had an issue with the background of the game, when working with the GUI. Making any edits on the GUI proved challenging because the background would disappear, leaving only the push buttons and textboxes visible.

Another challenge for our project was when making the rounds for the game. We had originally planned to increase the number of rounds every time the player clicked the ‘drop’ button but we then assumed a more practical way of increasing the rounds, by creating 4 pages, one for each round.

Furthermore, there was the challenge of the ‘Win/Lose’ screen for the end of the game. We had planned to create two separate pages for the player winning and another for the player losing the game. However, we decided to make one screen instead, showing the amount of money the player had won at the end of the game. This makes it easier for the player to see exactly how much they have won at the end of the game. If the player loses the game, the total winnings would be displayed as zero. Doing this reflects how our game is not a standard ‘Win’ or ‘Lose’ game. The initial ‘Win/Lose’ screen would’ve meant the game would have led to the same outcome, whether the player had won £1 or £1,000,000.

Another challenge we faced was randomizing the questions so that only one showed up in our static text box rather than our entire list of questions showing up as it had done. We fixed this by adding the questions (labelled Q) to our string as well as the answer options (labelled A, B, C, D) whereas initially we had only had the answers.

There was also an issue with the practicality of working on our game, in terms of the group being able to work on individual aspects of the game. This was due to the fact that there was only one updated version of the game at a particular time. It was difficult for us to work on different parts of the game and put it all together, so instead we had to work on the project in stages as we had to wait for a particular task to be finished before moving onto the next.

## Conclusion

This project has given us, as a group, insight on what we can achieve. The project started off as just an idea on a piece of paper, which we then developed into a proper functioning game. We exceeded our expectations of our capabilities when using Matlab, and we are astonished with all we have accomplished after just 10 weeks. If we’d had more time we would’ve liked to implement more ideas into our game, such as a multiplayer option. We also had the idea of creating a ‘host character’ to ask the trivia questions to the players, to make the game more interactive. Although we were not able to implement these last few ideas due to time constraints, we are proud of how this project has turned out.

## References

We drew inspiration for our project from a previous group who also made ‘The Million Pound Drop’ game. This was done because at moments whilst completing this project, we were unsure of what the next step was, in creating our game and from looking at a previous groups project, it allowed us to proceed with our project. We also coped their random question selection code: “

Round1questions={Q1R1,Q2R1,Q3R1,Q4R1,Q5R1,Q6R1,Q7R1,Q8R1,Q9R1,Q10R1,Q11R1,Q12R1,Q13R1,Q14R1,Q15R1,Q16R1,Q17R1,Q18R1,Q19R1,Q20R1,Q21R1,Q22R1,Q23R1,Q24R1,Q25R1};

chosenquestionR1=Round1questions{randi(length(Round1questions));”.