|  |  |  |
| --- | --- | --- |
| Group Member | Student URN | Percentage of Mark |
| Patrick Dobinson | 6420583 | 25 |
| Ryan Croft | 6421218 | 25 |
| Yram Duadze | 6428487 | 25 |
| Alex Drew | 6417705 | 25 |

**MATLAB Mavericks Report**

Hangman

How to play

1. Create a folder named *‘Hangman’*.
2. Download all the files into the folder *‘Hangman’.*
3. Open MATLAB and then add the folder to the path on MATLAB.
4. Open ‘*twit.m’*, ‘*twit1.m’*, *‘wordgame.m’* and *‘hangman.m’* files and then save them with the names of each of the functions.
5. Type *‘wordgame’* in the command window to initialise the game.
6. Follow the instructions displayed and enjoy!

Motivations

When our group first got together, one of the first things we agreed on was that we wanted to create a game. We quickly decided that the game would be more along the lines of a retro game such as battleships or mine sweeper. After doing a bit of research we set our goals on Hangman. We chose this game as we saw it challenging to code, but simple enough respective to our MATLAB abilities. In addition, as technology continues to revolutionise traditional games, we wanted to bring Hangman into the 21st century, with no need for pen and paper! This 21st century vision is furthered by our social media aspect. With twitter, we can create a hangman community, facilitating interaction between players.

Improvements for the Future

1. When we added the ‘*twit*’ function (M,2003), Twitter didn’t allow repetitive tweeting from the same account. I.e it came up with an error.
2. Furthermore, we would like to tag the player’s twitter username in the tweet.
3. Add more levels for different categories to increase the length and difficulty.

Challenges

1. We initially tried to create the entire game in one script/function but ran into problems. We kept getting error messages so decided to create another function titled ‘*wordgame*’ which called the hangman function, hence making the code more simplistic.
2. When we ran the game, and had someone play it for us they inputted a capital letter which led to an error. To overcome this, we used the lower case command to change all characters to lower case.
3. Originally our tweet just stated that someone had played the game. Instead we made the tweet more specific if you win or lose.
4. We struggled with the PNG imaging, however found a command that made it easy. This command was called ‘*imshow*’
5. Initially the same word came up every time, so we expanded word bank and randomised it.

Results

It works! We have a game that is simple and straightforward, no knowledge of MATLAB needed to use the game, making it more accessible to everyone. It clearly shows you how many letters the word is, if your letter was wrong (PNG image) or right (letter appears instead of the asterisk) and how many lives you have left. Then if you win, you get a congratulations and if not you can simply enter the command ‘wordgame’ and play again, with new lives etc. In addition, a tweet is successfully sent on the twitter account when a player plays the game. Everyone who has used it has had no problems.

Communications

1. WhatsApp

We organised meetings to discuss the code and any ideas we had for the project, as this platform was quick and convenient to use.

1. GitHub

This platform wasn’t used as much for communicating as we decided to do all the code when we were together, organised through WhatsApp.

1. In Person

When we came together, we allocated tasks to match each person’s set of skills so no time is wasted and no work is overlapped.

Bibliography

M, M. (2003). *Hangman*. Mathworks.

Ruthramoorthy, N. (2016). *Twit*. Mathworks.