**ECO2048: Game 2048 Report**

**Motivation**

When first brainstorming what to do for our project, we quickly decided that a game would be the most entertaining to work on so we decided to recreate the game ‘2048’ as it seemed highly relevant to our lectures on matrices. And so we engaged to transfer the knowledge we learnt from our lectures into reality, which is game ‘2048’, which based on 4 x 4 matrix. The game's objective is to slide numbered tiles on a grid to combine them to create a tile with the number 2048. The game seems straightforward, but since there are only 16 available spaces to work with and a new 2 or 4 tile being generated with every move, if you do not have a good idea of what you are doing then you will quickly find yourself running out of space and as a result losing the game. Therefore, we set out to create this simple but challenging game of 2048.

**Methodology and Obstacles**

Our first obstacle was learning to use GUI, and we quickly found out that the app ‘App Designer’ was the best way to integrate the two primary tasks of app building, the visual components of the app interface (GUI) and actually the coding app behaviour. Although we also familairized ourselves with GUIDE, App Designer had a easy to use interactive interface that was effective and superior.

Originally we had planned to use a simpler and more plain base code from <https://www.reddit.com/r/matlab/comments/3x9yqa/i_made_the_popular_2048_game_in_matlab/>. However, we had many difficulties with the code, the most disturbing one being our inability to merge the code into the GUI we created from App Designer. As a result, our team agreed to start from scratch and work off a new base code. Therefore our project is based off the codes provided from <https://uk.mathworks.com/matlabcentral/fileexchange/46124-2048-matlab-edition>.

Our first step was to code the actual game, however since the game already existed, we felt that it wouldn’t be prudent to code another one, but rather to use a basic code available on the web and use our skills to develop it further.

1. Editting the base code to allow for different fonts, colours and shapes.
2. use audioread and player and its functions to play music and create a mute button
3. using imread to find the RGB arrays of images and then saving them into the icon data to create custom icons into the IUtoolbar
4. mapped callbacks to the icons to make a new game, mute and unmute and information buttons

Although we had learned how to use both the App Designer and guide functions in MATLAB in order to create our project with the original base code, we encountered difficulty when we used the new base code since they had already added a GUI themselves. Subsequently, we were unable to edit their GUI through either App Designer or guide and as a result had to make our edits to by adjusting the code rather than using the user friendly systems MATLAB has in place which proved challenging and more time consuming.

**Results**

As a result, we were not able to code the game as we originally had intended but instead we did our best to add features as we saw fit. This has allowed us to produced a unique version of the game on MATLAB with sound effects and a different layout. The sound effect given when two tiles merge was a bonus and was above what we had expected to achieve at the beginning of the project. Unfortunately we were unable to code an undo button or a volume slider which we also thought would be a nice addition to the game due to our lack of ability in matlab coding. Overall whilst we have produced a version of 2048, we did not code as much of the project as we had originally set out to but we tried to add as much as we could to showcase our abilities.

Contribution

|  |  |
| --- | --- |
| Samuel Cockroft (CockroftSam) | 20% |
| Jun Hao Deng (DaedenD) | 20% |
| Ethan Kelly (ek00524) | 20% |
| Gyu Kim (gitoffmedude) | 20% |
| Woon Ki Yuen (phyuen) | 20% |