# C# Game Report – Flo

*Sophie Darby, 170025581 Callum Darling, 170014082*

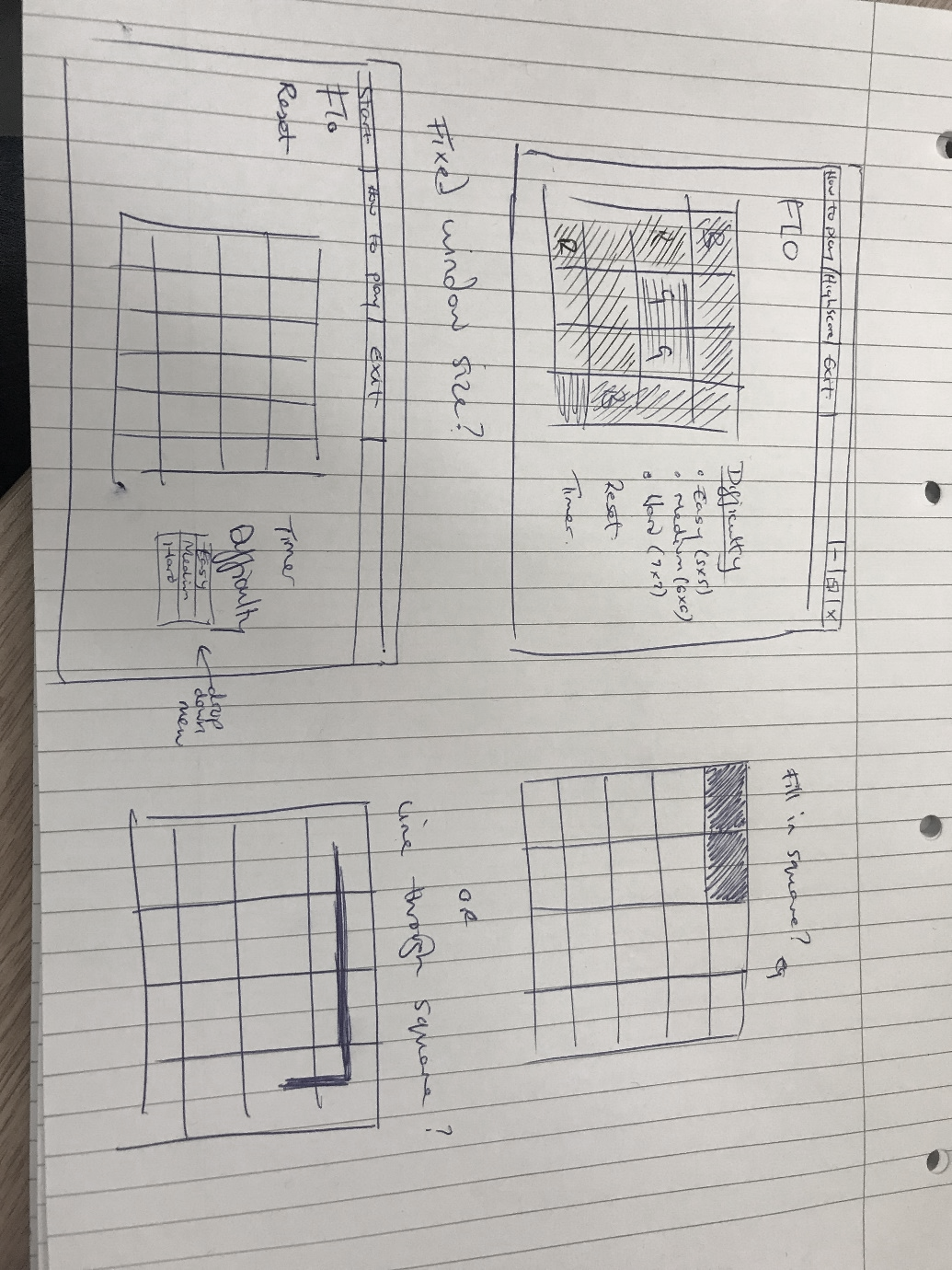
For our C# game, we decided to make our own version of flow free. The objective of the game is to connect the coloured boxes to each other without overlapping or cutting through the other ‘flows’ in the quickest time.

**Our Approach**

The first decision to be made was which game we were going to make. We had 3 choices, a winning chess move simulator, Flow game and Sudoku. The chess move simulator and Sudoku seemed overly difficult to code with the knowledge and time we had so we decided to make Flow.

After we picked the game, we had to decide the size of the 2D array we were going to create. This is when we chose to have easy, medium and hard options. Each option would need a different sized array; easy 5x5, medium 6x6 and hard 7x7.

Our next port of call was to sketch the window for our game (*Fig1 below*)



**Difficulties**

We had several difficulties when creating the game. Our first issue was when making the 2D array. The array was displayed and changed size from 5x5 to 6x6 to 7x7 when the appropriate radio button was clicked. However, when we tried to click easy again after the hard or medium was selected, the array wouldn’t go back to 5x5. We ended up solving this problem by totally destroying the array a creating a new one every time the board size was changed. Although this may not have been the most efficient method we decided that it would be fine for this project since there is a small number of objects and the board should not be getting reloaded that often. Another problem we had was using the git source control system. While we had experience with using git an GitHub before instead of using it normally as we had before we attempted to use a plugin for visual studio that we had not encountered before. It ended up being somewhat unintuitive and we spent a decent amount of time attempting to decipher the documentation. We overcame this problem by simply using the GitHub desktop app instead and deciding to leave the plugin for now. Our final major problem was finding a way to effectively store the different levels of the game and then also the solutions to those levels so that the user had a win condition. We decided the most efficient way to generating levels would be to create a method that would only require the coordinates of the ‘nodes’ in our grid and the colour that they needed to be, this meant that we could create a number of levels with the minimum amount of inputs. For the solutions we decided that we could convert the entire board into a string at any time. Having done this we then completed the levels ourselves, converted the board state to string and then stored for comparison with the user.

It should be noted that we did take our levels from the ‘Flow Free’ mobile game

https://www.bigduckgames.com/flowfree/