# AC22005 – Coursework 1 – C# Array Game

# Group 20

## Andrew Holligan - 170014201

## Jodie Laurenson - 170006409

## Aim

The aim of this project was to create a game in C# using Visual Studios which features a code-generated grid of buttons. This game had to be neat, attractive and functional and it should have programmed intelligence.

## Approach

For this assignment we decided to create a game which involved the player having to click matching adjacent boxes of the same colour in order to clear the board and gain points. We then developed our game to feature two different game modes including a time trial where players are against a clock and multiplayer where two players can battle against each other to get the higher score.

We managed to attempt most of the further extensions as they weren’t as much of a problem as we thought they’d be. The high score table proved to be harder than we thought but the different game modes and sound effects were no problem for us

## Problems & solutions

We didn’t experience many problems while implementing our game but one of the problems we faced was the timer. We struggled to find a solution of a timer that would only reduce the seconds and stop after the time had elapsed. We then came up with the solution of using the timer class and timerTick event handler which allowed us to decrease a counter to keep track of the time. This then led to another problem as the timer was in another thread that made it difficult to call other methods without causing an exception. We then overcame this problem by moving some methods away from this thread and using the invoke function.

We also struggled with writing our recursive algorithm to find all the squares together with the same colour. We kept getting stack overflow exceptions because we were searching adjacent squares for squares of the same colour. This meant that the program would get stuck in an infinite recursive loop between 2 squares as the colour wouldn’t change. We fixed this issue by temporarily changing the colour of the squares to white and then changing them back after the algorithm was completed. This happens so quickly it is unnoticeable to the user.

Another challenge we encountered was the score board. We saved the score board to file and read it in again when we needed to show it but we needed to sort it by score. We could not find a way to sort the array of objects so we decided to write our own bubble sort as it is a small number of objects so a more complicated and efficient algorithm was not required. We also had an issue with the list view as when we tried to generate it from code the columns were far too narrow and we could not find a way to change it. So we added the columns to the form by hand and then used the code to add each row instead.

# Reflection

We were happy with how we worked on this project and there isn’t much that we would do differently. We had considered taking the game further and adding power ups so this is something we would definitely add if we were to do this project again.

# Conclusion

Altogether we are very pleased with how this project has gone. We worked well together and have developed a sophisticated solution with multiple extensions. We put a lot of work into this assignment and dealt with the issues we faced well to create a good all round C# game.

Word count: 584