

# Software Engineering Assignment 3

## Structure

I decided to use for loops for the bulk of my program that would shift the board, end the game, or do nothing depending on which conditions are satisfied. I decided that placing all the game operations into functions made the main easier to follow and more compact. This also helped with the debugging process.

The structure is further explained through comments within the source file.

## Functions

In my program I defined a function called ***shiftGrid***. This function is able to shift the board in all four directions due to my function ***rotateGrid***. Depending on whether the user input is w, a, s, or d the board is rotated either 0, 1, 2 or 3 times respectively. I decided to use these two functions as I realised that the shifting process is exactly the same for all four directions, the only thing that changes is the orientation of the board. Therefore I realised that it was not worth defining four functions that did exactly the same thing but in different directions.

The shift function itself calls on one more function which is ***shiftZeroes***. If the input is either w,a,s,d then after the board has been rotated then all the numbers are shifted upwards to fill in any zero tiles. I decided to call this function before attempting to merge any tiles as it simplifies the merging process significantly. Instead of the program being required to check a tile two spaces away or a tile three spaces away to see if they can merge, the program only needs to check the adjacent tile as there are no more zero tiles between the non-zero tiles. After two numbers that are the same have been merged, the ***shiftZeroes*** function is called again to shift the board again.

The ***checkforspaces*** function and ***gameover*** function work in conjunction with each other. At the beginning of each round the function ***checkforspaces*** is called. If it returns false (no spaces) then the function ***gameover*** is called and if ***gameover*** is true then "game over" is printed and the program ends. If ***gameover*** does not return true then this means that although there are no unfilled spaces in the board there are still possible moves and the rounds continue until ***checkforspaces*** returns false and ***gameover*** returns true.

At the end of each round if the board has been shifted (***shiftGrid*** returns true) and ***checkforspaces*** returns true then the function ***randomTwo*** is called that will keep generating a random number until the square that the random number corresponds to is zero and then a two will be placed in that square.

## **Additional features**

### ***In 2048variation.cpp***

I have added a save feature that, when the user input is “save”, a file is created called savefile.txt and the contents of the vector game is written into the file. If the file already exists then it is overwritten with the new version of the vector game.

I have also added a scoring feature that will add the combined value of two tiles to the score every time two tiles merge. The score is printed at the end of every round. When the game is saved, the score is also stored in the 16<sup>th</sup> index of the game vector. When the save file is inputted at the start of a game, if the size of the vector is 17, then the accumulated score so far is shown, otherwise the score is 0.

Finally, I have decided to show the zeroes in the grid as dots as I thought that this increased clarity and therefore made the game more user friendly as I found during testing that the 8s were often mistaken for 0s.