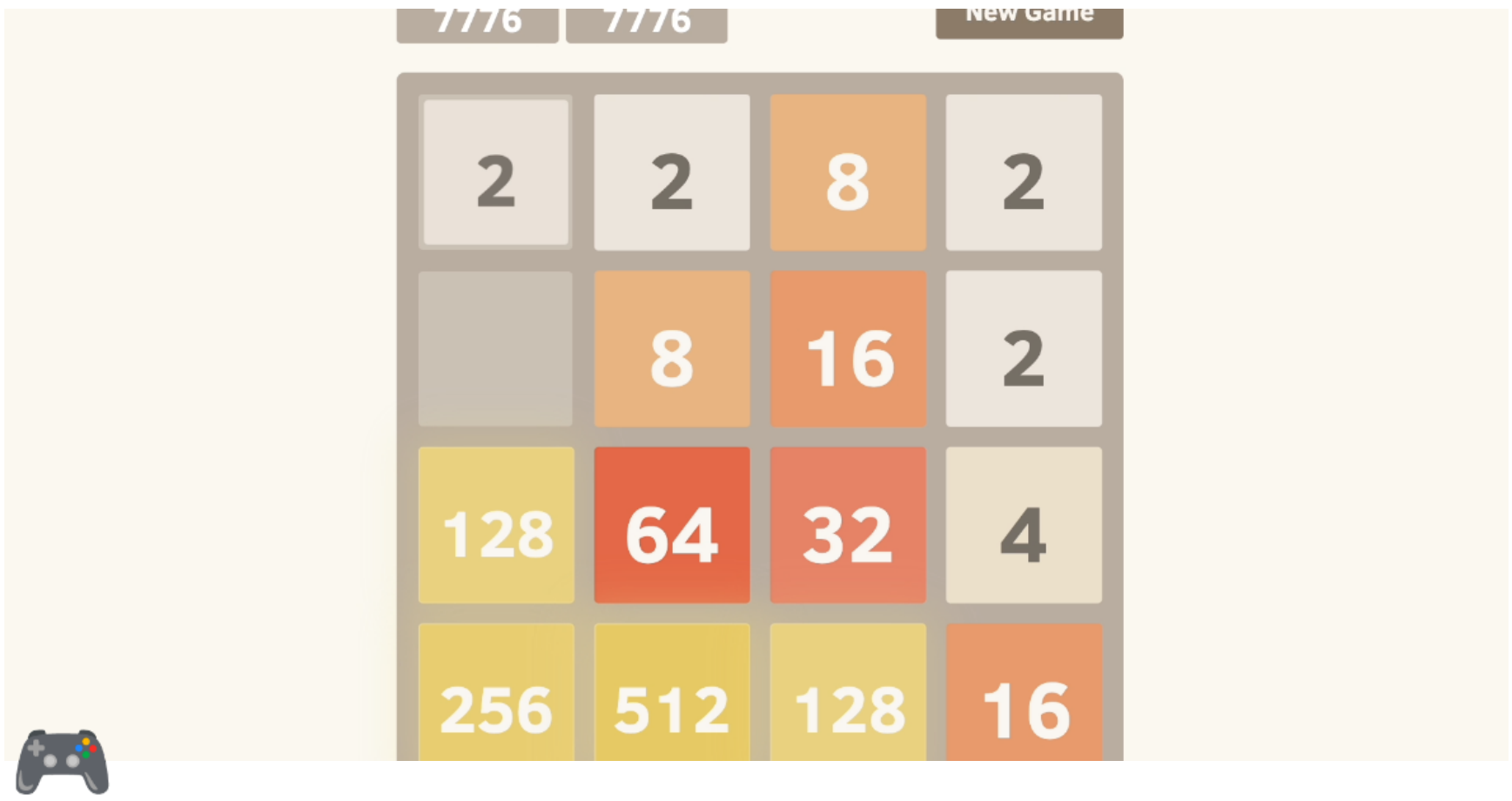

Assignment 01

COMP 295: Introduction to C/C++

❗ **Deadline: 26th March 2023**

➔ Plagiarism is absolutely forbidden.
➔ Do not use any programming constructs that are not covered in class.
➔ Use only what has been covered in class (conditional statements, loops, arrays and functions)
➔ Submit the repository link of your assignment (pushed on Github) along with the actual assignment on moodle.
➔ **Your program must not crash**. If the user enters wrong input, your program should handle it and asks for the input again.

2048 Game

Objective:

The objective of the game is to slide numbered tiles on a grid to combine them to create a tile with the number 2048.

Instructions:

1. Display a menu with (Start, Instructions and Quit).
2. When the user choose Start, the game starts with two tiles, each with a value of either 2 or 4, on a 4x4 grid.
3. The player can move the tiles up, down, left, or right using the arrow keys. All tiles slide as far as possible in the chosen direction, until they are stopped by either another tile or the edge of the grid.
4. If two tiles of the same number collide while moving, they merge into a single tile with the total value of the two tiles.
5. Every time the player makes a move, a new tile randomly appears on the grid with a value of either 2 or 4.
6. The game ends when the player reaches the 2048 tile or there are no more possible moves.
7. Keep a Score counter of the score and keep updating it with every move. Your counter should be displayed on the screen all the time.

8. Keep a highScore counter which keeps the highest score and update it if your Score counter > highScore counter. It should also be displayed on the screen with the Score counter.
9. After game ends, you must prompt the user if he/she wants to play the game again or go back to the main menu.

Implementation:

To implement the game, follow these steps:

1. Create a 4x4 grid to represent the game board. You can use a 2D array to store the values of each tile on the grid.
2. Initialize the game board with two random tiles with a value of either 2 or 4.
3. Create a function to display the game board on the console.
4. Create a function to handle the movement of tiles on the game board. This function should take a direction (up, down, left, or right) as input and slide the tiles as far as possible in the chosen direction, combining any tiles with the same value that collide.
5. Create a function to generate a new tile on the game board with a value of either 2 or 4.
6. Create a function to check if the game is over. This function should return true if the player has reached the 2048 tile or if there are no more possible moves.
7. Create a main loop that continuously takes user input (arrow keys) and updates the game board accordingly. The loop should end when the game is over.

Tips:

- You can use the **rand()** function to generate random numbers for the value and position of new tiles.
- Use nested loops to iterate through the 2D array that represents the game board.
- Break down the movement function into smaller sub-functions to make it easier to understand and implement.
- Use the **system("cls")** command to clear the console before displaying the updated game board. (Window users)
- Use **system(”clear”)** for macOS users. Otherwise you can use:

```
#include <unistd.h>
#include <term.h>

void ClearScreen()
  {
  if (!cur_term)
    {
    int result;
    setupterm( NULL, STDOUT_FILENO, &result );
    if (result <= 0) return;
    }

  putp( tigetstr( "clear" ) );
  }
```

Game Link:

Here’s a link of the game (chrome extension). Give it a play and understand the working!

https://tinyurl.com/2048GameLink

Demo:

Here’s a link of the demo:

https://youtu.be/LmODr8la2YQ

Your game should resemble to the above demo. However, adding colors to your output is a choice and doesn’t carry any additional marks. If you still wish to, following is a link where you can get help:

https://www.codespeedy.com/color-text-output-in-console-in-cpp/#:~:text=Colorizing text on the output,text on the output console.

Github Instructions:

1. Create an ‘assignments-comp295-fccu’ repo on Github.
2. Create a folder ‘ass-01’ in it.
3. Put all of your code in this folder.
4. Make a README.md file for the documentation of your assignment.
5. Your program must have at least **10 reasonable commits**.
6. Copy the link of your repo ‘assignments-comp295-fccu’ and submit it on moodle in a **.txt** file.

End.

> "Debugging is like being a detective in a crime movie where you're also the murderer." -Unknown (but probably a C++ programmer)