CS 12: Computer Programming II
Machine Problem I: 2048

1 Introduction

For this machine problem, you will be implementing a terminal-based game using all the C++ concepts discussed so far.

2 The Game

2048 is a single-player puzzle game created in March 2014 by 19-year-old Italian web developer Gabriele Cirulli, in which the objective is to slide numbered tiles on a grid to combine them and create a tile with the number 2048.

A player moves the tiles using the four arrow keys. Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4. Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again in the same move.

3 Implementation

You may implement the game in C++ in any way, provided that you meet the following specifications:

3.1 Code Template

You are required to use the code template given. You cannot erase parts of the code template. Insert your implementation on the appropriate sections of the template.

3.2 Classes

Implement a class named G2048 that represents a 2048 game. An object of this class contains the 4x4 board configuration of the 2048 game and the score. This must also include the following member methods: startGame, addTile, move, drawBoard, hasReached2048, hasMove, loadGame, saveGame. You are free to add additional methods and functions that you think are necessary for your implementation.

3.3 Controls

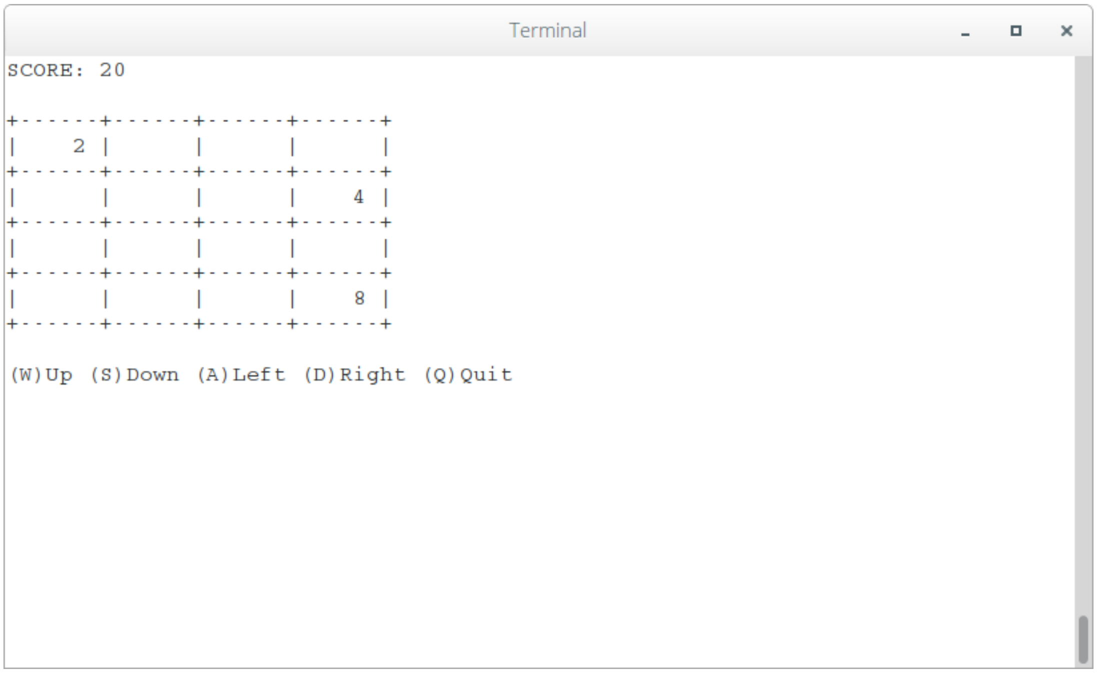
Use the following characters as controls: *W* - Up, *S* - Down, *A* - Left, *D* - Right, and *Q* - Quit.

3.4 Display

After each turn, the 4x4 board must be displayed in the terminal. Previous board states must be cleared such that only the current board state and the running score will be displayed. To clear the screen at each turn, you can use the following code snippet.

```
// system("cls") works on Windows OS, system("clear") works on Unix-like OS
// system("cls") returns 0 if it works
if (system("cls")) system("clear");
```

The following image shows an example of a board stated displayed in the terminal.



```
SCORE: 20

+------+------+------+------+
|    2 |      |      |      |
+------+------+------+------+
|      |      |      |    4 |
+------+------+------+------+
|      |      |      |      |
+------+------+------+------+
|      |      |      |    8 |
+------+------+------+------+

(W)Up (S)Down (A)Left (D)Right (Q)Quit
```

3.5 Persistence

At any time during the game, the player can quit his game (See controls subsection). You need to provide persistence such that the player can resume his game. Upon quitting, save the board configuration and the current score to a file whose name is passed as a commandline argument. In the case when no filename is passed (in the start of a new game), you must prompt for the filename upon quitting. You must these load persistent files accordingly when the corresponding filename is passed at game start/resume.

4 Deliverables

For this machine problem, you must submit the following:

- All source and header files
- Documentation - this must discuss in detail your implementation.

Email the deliverables to your lab instructor with the subject: **[CS 12 MP1]** <**Surname1, First_Name1 (Section) and Surname2, First_Name2 (Section)**>) (*Ex.* [CS 12 MP1] dela Cruz, Juan (MDE/WFY) and Santos, Maria (MLN/THV)). You are required to demonstrate your implementation on a date to be scheduled by your instructor.

5 Deadline

The deadline of submission of complete deliverables is on **17 April 2015, 11:59pm**.

6 Notes

- You may only collaborate with your partner.
- Code plagiarism will be dealt with mercilessly.
- Follow all specifications. Non-compliant submissions will not be entertained.
- Any modifications to the specifications will be announced in the FB group. Check it regularly.
- As always, if you have any questions, you may consult with your instructors.

References

[1] 2048 Official Website. Available: *http://gabrielecirulli.github.io/2048/*.