CS 12 Machine Problem 1 - 2048

Submitted by: Gueco, Maria Rosario

Generated by Doxygen 1.8.9.1

Sun Apr 19 2015 23:40:53

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 G2048 Class Reference

Public Member Functions

- G2048 (char *filename)
- void startGame ()

Private Member Functions

- void addTile ()
- char getInput ()
- bool move (char direction)
- void drawBoard ()
- bool hasReached2048 ()
- bool hasMove ()
- void loadGame ()
- void saveGame ()

Private Attributes

- int **board** [4][4]
- int **score**
- char * **filename**

2.1.1 Constructor & Destructor Documentation

2.1.1.1 G2048::G2048 (char * *file*)

This initializes a game of 2048 - the board, score, and filename. If there is no filename given, a fresh board is given, else, the function loadGame() is called.

2.1.2 Member Function Documentation

2.1.2.1 void G2048::addTile ( ) [private]

This function randomly adds a tile of value 2 or 4 to the board after each valid turn. The random value generated by rand() determines which tile will have a value added and whether that value is 2 or 4.

2.1.2.2 void G2048::drawBoard ( ) [private]

This function clears the terminal and prints the current score, state of the board and the directional keys. The functions used for clearing the screen has only been tested on a Windows machine . This function will also notify the player if they have reached 2048.

2.1.2.3 char G2048::getInput ( ) [private]

This function prompts for an input character from the player, then returns the value. If the player inputs a series of characters, it will only take the first one.

2.1.2.4 bool G2048::hasMove ( ) [private]

This function uses nested loops to checks if there is still a possible move. There is a move left if at least one tile is blank (board[x][y] == 0) or if there are still tiles of the same value next to each other.

2.1.2.5 bool G2048::hasReached2048 ( ) [private]

This function checks if the game is already won. Tiles are checked for 2048 using nested loops.

2.1.2.6 void G2048::loadGame ( ) [private]

This function tries to load a 2048 game from the specified file. If it fails to open the file, it loads a fresh board. Failure to open a file can be because the file extension used was wrong. The program is also not able to check if a file is a valid file for the game.

2.1.2.7 bool G2048::move (char *direction*) [private]

This function moves the tiles in the specified direction using nested loops. If two same tiles are the same are moved next to each other, they are combined into one and the new value is added to the score. But like in the game, the tiles are only added if they have not yet been combined for this move. (ex. Tiles are moved to the right = [2][2][2][2] -> [ ][ ][4][4]). It returns true if the state of the board has changed (move is valid). If the player inputs a character not in the choices, the program will return false.

2.1.2.8 void G2048::saveGame ( ) [private]

This function asks the player if the want to save the game or not. If yes, it saves the instance of this class to the specified file. Note: If the argument does not have a filename, the program saves it without adding a filename. If the argument is NULL, the player is prompted for a filename.

2.1.2.9 void G2048::startGame ( )

This function starts/resumes a game of 2048. For as long there are valid moves, the games continues until the player decides to quit or there are no moves left. Every loop the program asks for an input. If the player gives a valid move, addTile() is called. When the loop is broken, saveGame() is called and the program ends.

The documentation for this class was generated from the following files:

- CS 12 MP1/2048.h
- CS 12 MP1/2048.cpp