Project Report/Documentation

What is this:

This project is an implementation of the concept of the game *2048* that is available on mobile phone platforms. The game is to combine (add) numbers on the grid to get bigger numbers before no possible moves are available (loss).

Files:

In the zip file:

```
├── game.c      -- game implementation
├── grid.c      -- grid library implementation
├── grid.h      -- grid library header file
├── report.pdf  -- this report
└── Makefile -- the Makefile!
```

How to build:

Go to the directory in command line and simply do the make command. You will then get the executable file “2048c”.

How to run:

The game can be run without extra argument, in which case it will start a default 4-by-4 grid. Extra arguments are:

-h	-- display a help on how to use the arguments and exit
-c [x] [y]	-- start a customized game of x-by-y

Example:

```
./2048c -c 10 20
```

Game Features

Spawn:

After generate a new grid and making moves, certain percentage of the tiles will be filled by 2. This percentage is called spawn rate in this game and can be changed by entering “r” in the game. The spawn rate should be a floating point number between 0 and 1 (0 and 1 included). The default value of the spawn rate is set to 0.3.

Loss:

If all tiles on the grid is surrounded by a number that is different to itself, then this round of game is considered loss. Because there is no way to combine any more numbers any more in this case.

Future Improvements

Save:

A save function can be added to the game that allows player to resume the game after saving and quitting the game.

Scoreboard:

A scoreboard function can be added to the game to display the least movements made to get the biggest number.

GUI:

GUI looks more appealing to player than command line. A GUI can be added to enable keyboardless interaction with the game, and make the game more user friendly.

External Software:

I did not use an external software, but I looked up a “shuffle algorithm” on stackoverflow.com and used it in my game to generate non-repeat random numbers.

Link: http://stackoverflow.com/questions/5064379/generating-unique-random-numbers-in-c