Christian Saltarelli
Prof. Arias
CMPT 220-204
4 May 2018

The Remake of 2048

Abstract -

As to date, I have been working on creating a remake of the wildly popular mobile game; 2048. Although, this game may come off to the user as being very simplistic. I found that the theory behind such a game is extremely intriguing. Along with this I also viewed the recreation of this game to be an incredible learning experience as I have begun to learn various aspects of Java that I felt I may have never come across otherwise.

Introduction -

The motivation for wanting to create a remake of this popular mobile game had stemmed from my own personal fascination in mobile development. Ever since I had started programming, I have always loved the idea that I could have the ability to create something that millions of people around the world could use to better their everyday lives. With the popularity of this mobile application I wanted to see how such a game could be made in a different language (Java). Throughout the rest of this paper I will be discussing how my version of 2048 was developed and how the system should be used.

Detailed System Description -

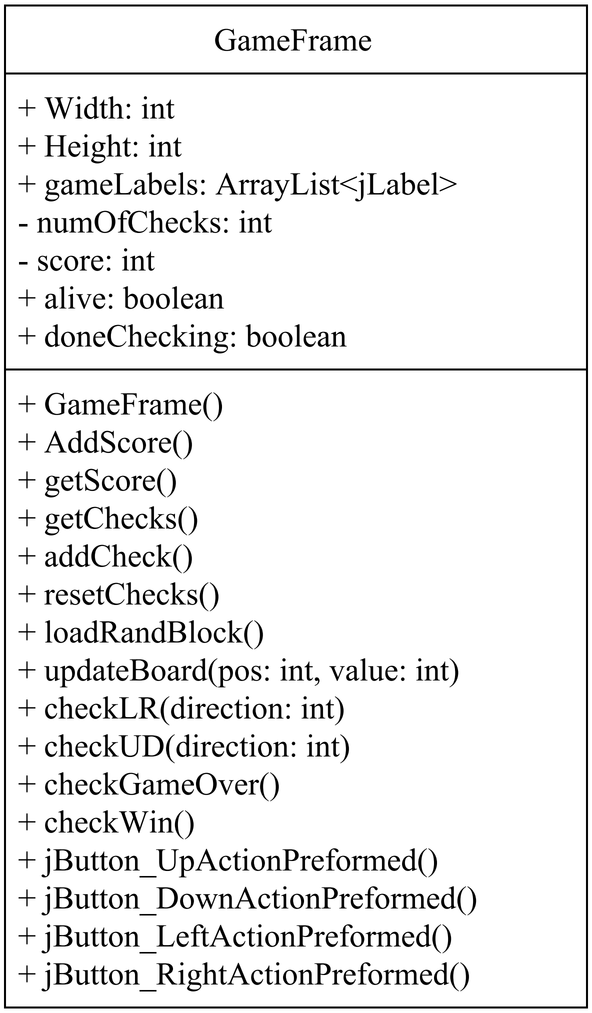
The system or program of my version of 2048 is intended to work very similar to that of the original game; 2048. When the user runs my application they will be presented their Game

Frame for that current session of the game. The user can then click on the buttons “Up”, “Down”, “Left”, and “Right” in order to make moves within the game in hopes of beating the game and acquiring the 2048 block. A feature that I had added, was the ability of everytime the user makes a move the board is then checked four times in that given direction. The reason why I had added this feature was to avoid the redundancy of the user making the same move multiple times. I felt that this had made the game flow easier and had made the user’s experience better overall.

This system was created in NetBeans 8.2. NetBeans is known as an IDE meaning an Integrated Development Environment. An IDE in simple terms is basically used to describe a software application that gives programmers access to extremely comprehensive tools that aid in the development process of various types of software. For my application, I had written it in Java which is a general computer programming language used to produce programs for any computer system.

I was able to program this game by creating one class which I had called, Game Frame. In this class (as seen below in my UML Diagram) I have a preset Width and Height which I use to create a fixed size for the window of the application. I then have two other global variables which are the Score and the gameLabels. The score variable is used to keep track of the users Score while they are using the application. This variable is global due to the multiple references that I make within my code. My gameLabels variable is an ArrayList of jLabels which are used to represent each of the panels that can be found within my jFrame. The rest of my class is comprised of multiple methods, the first being a constructor method which allows for a new instance of this application to be created upon every run. I then have an accessor method for my

Score variable and I also have a AddScore method which allows for me to add points to the user's score as they make it farther within the game. I also have a series of getter and setter functions that are used to check if the program is done ‘checking’ the users move. I had implemented this to avoid the user making multiple moves before the system can finish; which allows for the system to run more efficiently. I then use two methods called loadRandBlock() and updateBoard(). These methods allow for me to randomly generate a block on the given grid, which is called after the user makes a successful mood (such as a block sliding or combining with another block). I then use my updateBoard() method to update the given block with the calculated value. After these methods I then have four action methods which are called upon from the user clicking on any of the four desired buttons. The actionPerformed methods then call on two Check methods which check if the user’s desired move is possible. These two methods are also used to update the user’s Game Board. The reason why I have two Check methods is in order to check the two different directions in which the bored may move. My first Check method is called CheckLR(), this method is only called upon if the user wishes to move Left or Right (hence the LR at the end of the method name). Depending on the direction the user wishes to move, the actionPerformed methods also send an argument of either ‘-1’ or ‘1.’ In this instance (of the user wishing to move Left or Right), if the user wishes to move Right then the method CheckLR will be called with a direction of ‘1.’ The ‘1’ in this instance signifies that the board must move positively or to the Right. This same train of thought is also used within my second check method, which is called CheckUD. In this method ‘-1’ signifies that the user wishes to move ‘Up,’ meanwhile, ‘1’ indicates that the user wishes to move Down. These methods and variables make up my GameFrame, which allow for the application as a whole to run.



```
GameFrame
---------------------------------------
+ Width: int
+ Height: int
+ gameLabels: ArrayList<jLabel>
- numOfChecks: int
- score: int
+ alive: boolean
+ doneChecking: boolean
---------------------------------------
+ GameFrame()
+ AddScore()
+ getScore()
+ getChecks()
+ addCheck()
+ resetChecks()
+ loadRandBlock()
+ updateBoard(pos: int, value: int)
+ checkLR(direction: int)
+ checkUD(direction: int)
+ checkGameOver()
+ checkWin()
+ jButton_UpActionPreformed()
+ jButton_DownActionPreformed()
+ jButton_LeftActionPreformed()
+ jButton_RightActionPreformed()
```

Requirements -

Due to the fact that this application is a game, the only problem in which this project is addressing is lack of entertainment in which many individuals may feel throughout a given day. The recreation of this mobile game also gives any user on a desktop the ability to play it without the need of a mobile device. Overall, the recreation of this popular mobile game is meant to bring endless entertainment to it's users while providing multiple platforms in which this application can be used.

Literature Survey -

This popular mobile game has been recreated thousands of times by countless developers of all ages. Just from my own personal research I have seen multiple versions of the game, however, I have also seen a lot of unique and creative additions in which developers have added to the game. Some of these additions have included things such as a bigger board to play with or I had also seen an individual who had added the ability for the computer to play for you. I found this option incredibly interesting due to the different ways the computer would play. For instance, in this remake of the game you could have the computer randomly select choices as you watched the board change according to those random moves. Overall, the recreation of this game has been done by an enormous amount of people and although many may think that the recreation of this game now would be rather stale, I must say that working on such a game has been enjoyable and an incredible learning experience.

User Manual -

This application is intended to be used on any desktop. Once the user begins to run the application they will be presented a window which will be the Game Frame for that current session. What the user will see will be a four-by-four board which will be known as the user's current Board. The user will also see four buttons across the bottom of their Game Frame that will be labeled Up, Down, Left, and Right accordingly. As the user will also be able to notice, they will see their Score along the top center of their Game Frame with the title of the game displaying '2048'. The user will also notice that one of the blocks within their current Board has been populated by the number two or four.

When the user wishes to begin to play the game, they can simply press one of the four buttons found on the button of their current gameScene. Once this happens, the Board will then be updated with the only populated block adjusting accordingly to the button pressed and another random block within their current Board will then become populated with a value of two or four. The user then must try to combine blocks of equal value within their current Board in attempt to get the 2048 block.

In addition, the user will only be able to combine blocks if the desired blocks fulfill two requirements. The first one being that each of the two blocks must have the same value, such as two, four, eight, etc. Secondly, the two blocks must be in either the same row or column. The user does not have the ability to combine blocks diagonally, only vertically or horizontally.

The user must also remember that it is possible to lose while playing this game, if the user's board becomes filled with blocks that are unable to combine with one another then the user has lost. At this moment, the games 'Title' text will change to state that the user had lost.

Conclusion -

The goal of this application, as with the goal of any gaming application is to bring endless entertainment and fun to that of the desired user. The recreation of this incredibly popular mobile game on desktop ultimately allows for more users to have access to this clever and addicting game. With that being said, this application is intended to bring entertainment to that of a wider audience of users.

References/Bibliography -

Any reference that may have been used throughout the development of this project can be found below.

https://netbeans.org/downloads/
https://www.youtube.com/watch?v=TrmC1cMtduc&t=2443s
https://www.java-forums.org/new-java/29836-possible-create-2d-arraylist.html
https://stackoverflow.com/questions/5522336/repainting-refreshing-jlabels-on-a-jpanel