## Criteria B: Design

By: Yash Oza

Date: June 4, 2016

### Detailed Design Specifications

|  |  |
| --- | --- |
| Criteria A | Criteria B Details |
| 1. There must be a grid game and/or puzzle | 1. 2D array (int x [][]=[4][4];): leading to a grid by the means of a JLabel or even JButton |
| 1. There should be pieces for the players to move | 1. JButtons leading to various commands in the ActionPeformed, with it’s different purposes 2. (specifically to my game, there are the left, right, up, and down buttons which move the pieces in my 2048 game, and allow the game to function) 3. (also I am attempting to add keyboard input to perform the function, without the use of JButton (so the pieces can move) |
| 1. There should be comments for the code to be easily adaptable | 1. Comments will be used to identify various methods and their purpose in the overall game 2. Comments will be focused on if/ while statements and major declarations, so that one can easily determine their function/purpose |
| 1. No animation | 1. No animation 2. Only applets 🡪 Graphical User Interface |
| 1. Should be a winning condition and/or score | 1. If statements 2. Checks to see if the player wins (if there are no places to put a tile in my 2048 game and if there are no possible moves left) |
| 1. There should be thinking and/or problem solving to complete the puzzle | 1. Topic is chosen with due thought 2. Game (2048) requires problem solving, with patterns of strategy |
| 1. There must be a theme to make the problem more engaging (eg. League of Legends, Spongebob). | 1. There are a plethora of themes that can be unlocked through the process of the game (identified through if statements) 2. Specificallt there will be a Marvel, Nintendo, and special GIF theme |
| 1. Should be instructions | 1. JLabels specifically created for instructions |
| 1. Should follow Good UI design principles | 1. Efficiency within code:  * Else/ else if statements in opposed to just if * Efficient code itself and meaningful placement of method calls * Good use of images and JLabels to have aesthetically pleasing screens |
| 1. There should be color and pictures | 1. JLabel’s with createImageIcon for pictures/ gifs 2. Set Colors: x.setForeground (Color.red);/ x.setBackground (Color.red); |
| 1. Pictures should be sourced at the top of code for copywriter purposes | 1. Use comments containing sourced information 2. Use //, /\*-\*/ |
| 1. Multiple screens | 1. Card layouts and JButtons through the ActionPeformed to have multiple screens |
| 1. Widget should use set dimension so that there are a correct size if needed | 1. Pictures correctly sized beforehand 2. JButtons correctly sized with setdimension |
| 1. A new widget should be used (menu, checkbox, radio button, and progress bar) | 1. Keyboard input 2. Ability to save game |
| 1. 500-2000 lines of code | 1. Code will be done efficiently |

### Feasible Design Ideas

## Themes

Option 1: Cool theme that fits with the color scheme

* High Concept statement: The user will have the option of using special gif tiles that fit with the color scheme for a unique take of the game.

Option 2: Marvel

* High Concept statement: The user will have the option of using special tiles with Marvel characters on them, each having an exponential output of 2 up to 2048.

Option 3: Nintendo

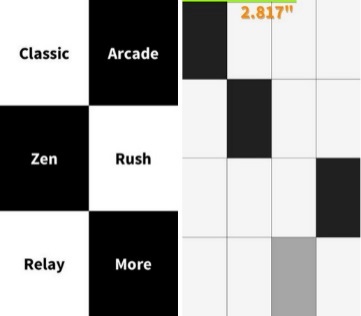
* High Concept statement: The user will have the option of using special tiles with Nintendo characters on them, each having an exponential output of 2 up to 2048.

I choose all 3 options as themes for my game as combined they will work amazingly and proficiently; especially with a 4 by 4 game (as such: 2048) with its advantages (referring to Criteria A).The game can start off with neutral plain tiles (specification 2: pieces to move in my game), but as the player succeeds past certain checkpoints (like tile 512) they will unlock special themes (specification 10: additional aesthetic appeal). This will be effective as it will add an alternative checkpoint/ wow factor to the game in opposed to winning (specification 5: winning condition), so that those who are challenged at the game can still rejoice with the triumph of unlocking a special theme. Furthermore all 3 themes are very influential/ popular amongst the age group. Thus they will be more than happy to see their favourite characters (which Nintendo and Marvel have almost a monopoly on) in a different format (referring to Criteria A again). This wide option of tiles will give a new sense of aesthetic appeal (specification 10) sourced at the top of my code (specification 11), to have good user Interface (specification 9), without the use of animation (specification 4).

## Color Scheme Selection

[[1]](#footnote-1)Option 1: Classic

* Background: white
* Foreground: Titles and Subtitles are White (with a contrasting background in the back)
* Accents: Tiles ranging from gray, to yellow, to orange, to red, to green

[[2]](#footnote-2)Option 2: Piano Tiles

* Background: white/black
* Foreground: Titles and Subtitles are Black/White (with its opposite contrasting background)
* Accents: Tiles ranging from many tones of gray, to white, and black

[[3]](#footnote-3)Option 3: “Special Purple Gif” theme

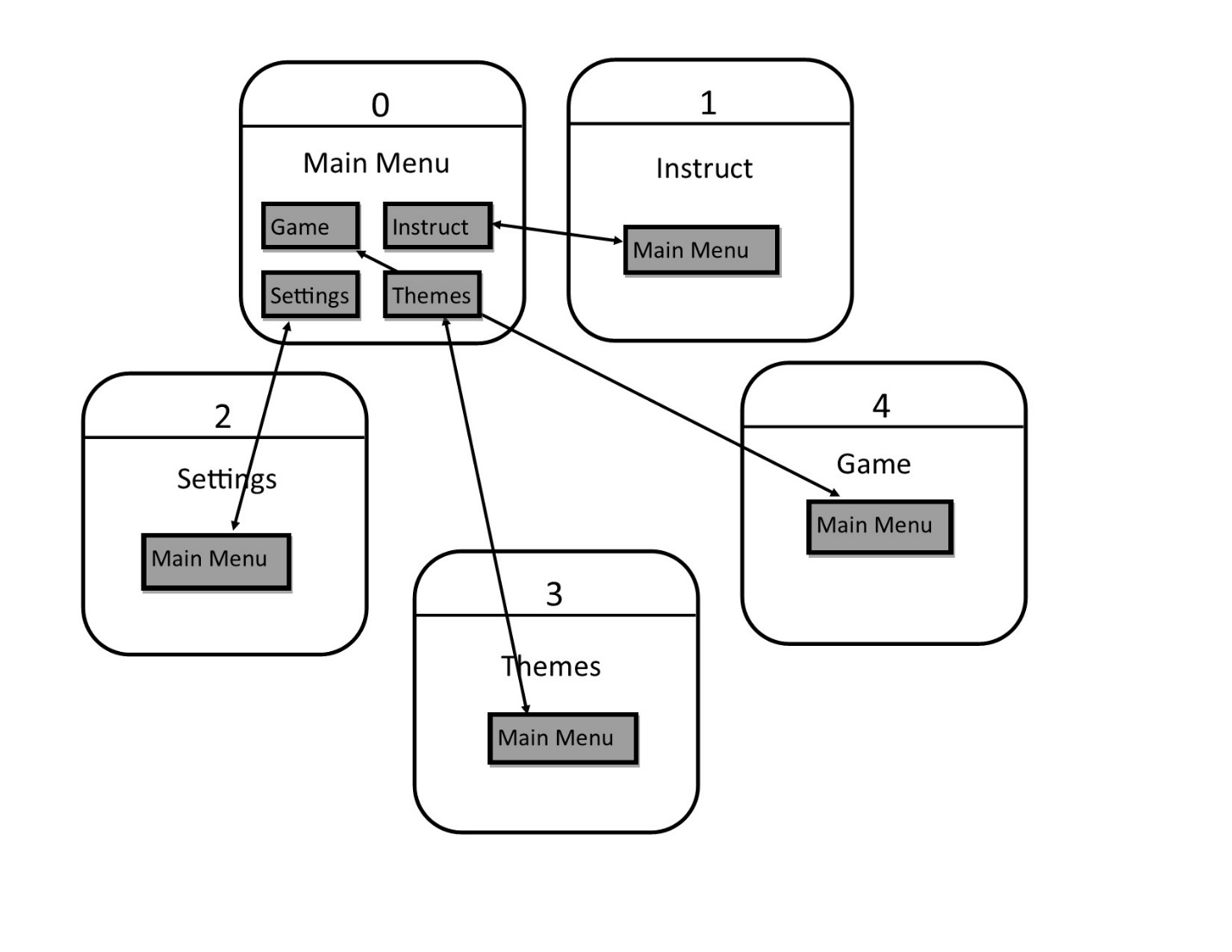
* Background: Tone of purple
* Foreground: Titles and Subtitles are White
* Accents: Tiles ranging from a darker tone of purple, to red, and blue

I am going to use the “Special Purple Gif” theme

|  |  |  |
| --- | --- | --- |
| Colour | Use | RGB Code |
|  | Background | 87, 64, 124 |
|  | Titles/ Subtitles | 255, 255, 255 |
| Varies | Button/ Tiles | All vary (especially from all the themes ranging from Nintendo to Marvel to the tile theme above) |
|  | JButton Background | 61, 41, 99 |

### Accurate and Detailed Designs

## Screen flow Diagrams:

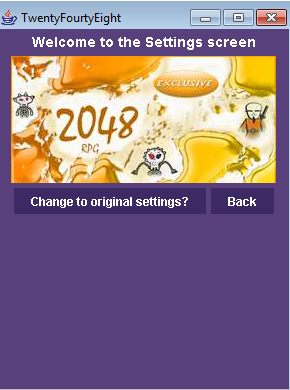


## Screen Layout:

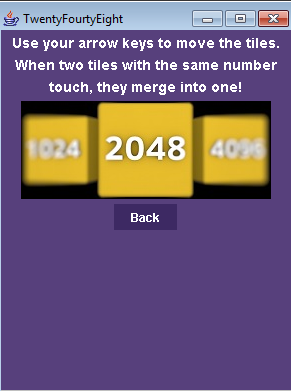
Here is the opening screen of my game with 4 JButtons leading to 4 different other screens (specification 12: multiple screens) with a gif picture and title that looks amazing (specification 10 / 9: aesthetically pleasing).



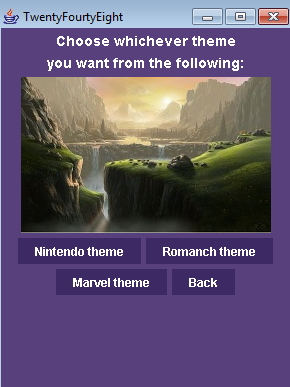
Here is the Setting screen of my game with 2 JButtons (one changing the games back to original settings, and the other back to the main screen) with a cool picture and title that looks amazing (specification 10 / 9: aesthetically pleasing).



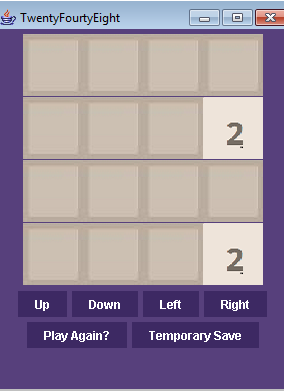
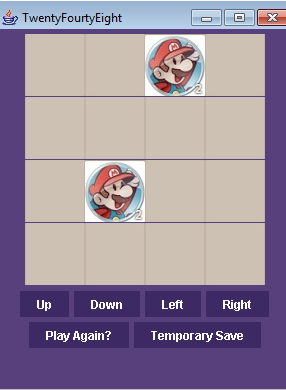
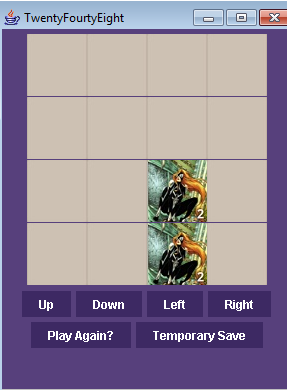
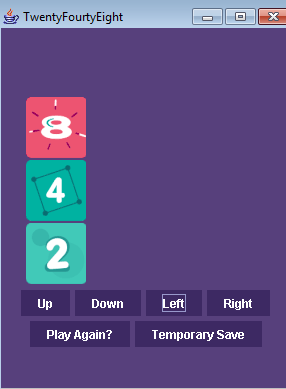
Here is the Instruction screen of my game with 1 JButton (moving back to the main screen) with a cool picture and title (specification 10 / 9: aesthetically pleasing) with the additional purpose of informing the player about the game (specification 8).



Here is the Theme screen of my game with 4 JButtons (3 of which changes the theme of the tiles and one that moves back to the main screen) with a cool picture and title (specification 10 / 9: aesthetically pleasing). (specification 7: theme)



Here is the Game screen of my game with the 4 possible tile themes. It has 6 different JButtons with various functions (like the function of the game to move the pieces (specification 2), and lead to a winning condition (specification 5), with another one leading back to the main screen (specification 9: Good UI), and 1 to temporarily save the game (specification 14: Something new).

\*All the screen sizes are the same\* 260 by 360 (specification 13)

## Pictures

**Tile Pieces:**

Here are the normal tile pieces (which are the original ones)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Pic File | Name | Pic File |
| N1.jpg |  | N64.jpg |  |
| N2.jpg |  | N128.jpg |  |
| N4.jpg |  | N256.jpg |  |
| N8.jpg |  | N512.jpg |  |
| N16.jpg |  | N1024.jpg |  |
| N32.jpg |  | N2048.jpg | [[4]](#footnote-4) |

Here are the Nintendo tile peices

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Pic File | Name | Pic File |
| d1.jpg |  | d32.jpg |  |
| d2.jpg |  | d64.jpg |  |
| d4.jpg |  | D128.jpg |  |
| d8.jpg |  | d256.jpg |  |
| d16.jpg |  | d512.jpg | [[5]](#footnote-5) |

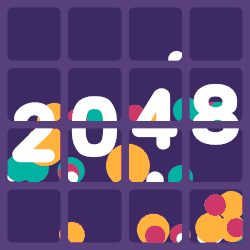
Here are the Marvel tile pieces

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Pic File | Name | Pic File |
| m1.jpg |  | m32.jpg |  |
| m2.jpg |  | m64.jpg |  |
| m4.jpg |  | m128.jpg |  |
| m8.jpg |  | m256.jpg |  |
| m16.jpg |  | m512.jpg | 5 |

Here are the cool gif tile pieces

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Pic File | Name | Pic File |
| r1.jpg |  | r64.jpg |  |
| r2.jpg |  | r128.jpg |  |
| r4.jpg |  | r256.jpg |  |
| r8.jpg |  | r512.jpg |  |
| r16.jpg |  | r1024.jpg |  |
| r32.jpg |  | r2048.jpg | [[6]](#footnote-6) |

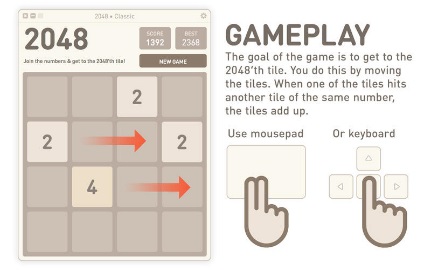
Main menu Picture:

6

Settings Picture:

[[7]](#footnote-7)

Themes picture: Instructions picture:

[[8]](#footnote-8) 10

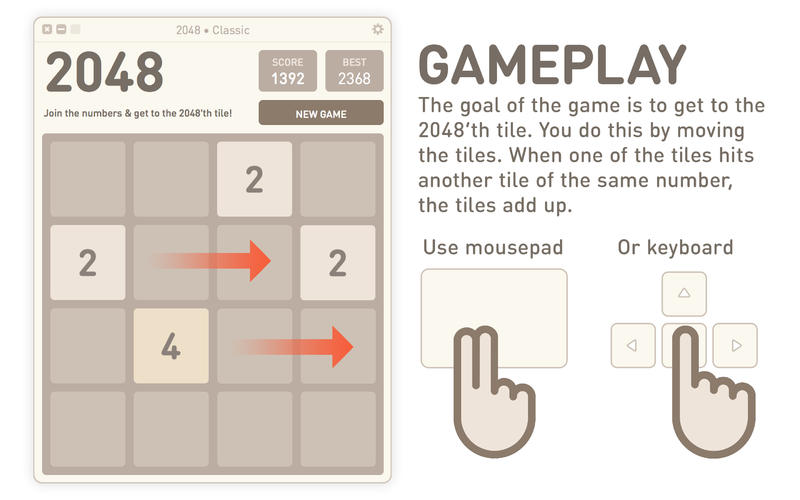
## Game Instructions:

All titles and subtitles have been pasted in word before they have been coded into the game to avoid any errors, and at the same time other efforts have been made to have a good UI (specification 9). Here are the clear instructions of my game (specification 8).

These instructions are directly quoted from the official 2048 website.

“The objective of the game is to get the number 2048 using additions of the number two and its multiples. You will have a grid of 16 tiles. Two numbers will be given: usually two number twos, maybe number four. Move up or down, left or right trying to join two equal numbers. When two equal numbers are in touch, they will add up. If we are run out of equal numbers on our grid, or we can not put them in touch, the game will provide us another two, a four or even an eight so we can keep on playing. If there are no free tiles on our grid, the game ends. By adding numbers, we get higher numbers and we can approach to 2048, which is the goal of the game.”[[9]](#footnote-9)

Here is a diagram created by , to help show how the game works.

[[10]](#footnote-10)

As seen the game requires a certain amount of problem solving, spatial visualisation, and patterning, which are all key attributes that are required in mathematical fields (specification 6).

## Variable and Array structures

Actual Game Board

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] |
| [0] |  |  |  |  |
| [1] |  |  |  |  |
| [2] |  |  |  |  |
| [3] |  |  |  |  |

It will hold this data for game 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] |
| [0] | 1 | 1 | 1 | 1 |
| [1] | 1 | 1 | 1 | 1 |
| [2] | 1 | 1 | 1 | 1 |
| [3] | 1 | 1 | 1 | 1 |

A ‘3’ in the tracker array matches the 1.(gif/jpg(according to the variable assigned)) picture in the picture section.

**The grid array will be declared like this:**

int pic[][]={{1,1,1,1}, {1,1,1,1}, {1,1,1,1}, {1,1,1,1}}

int row=4; //for the dimensions – both row and col are the same.

**The grid will be displayed with a 1D button array.**

JButton a[]=new JButton[16];

**This will allow me to show the screen with a simple nested loop:**

int m=0;

for (int i=0; i<row; i++)

{ for (int j=0; j<row; j++)

{

a[m].setIcon(createImageIcon(pic[i][j]+”.jpg”);

}

}

**I will also make variables to track which tile will be used.**

char a = ‘n’; //for normal tiles

a = ‘d’; //for nintendo for tiles

a = ‘m’; //for marvel tiles

a = ‘r’; //for gif tiles

**Then the tiles position will be updated with the redraw method:**

redraw ();

1. Cirulli, Gabriele. Classic Tiles. Digital image. 2048. World Wide Web, Mobile Phone, Nintendo 3DS, n.d. Web. 5 June 2016. [↑](#footnote-ref-1)
2. Hu Wen Zeng. Piano Tiles. Digital Image. Umoni Studio. n.d. Web. 5 June 2016. [↑](#footnote-ref-2)
3. Cirulli, Gabriele/ Romain Cousin. Adapted Tiles. Digital image. 2048. World Wide Web, Mobile Phone, Nintendo 3DS, n.d. Web. 5 June 2016. [↑](#footnote-ref-3)
4. Cirulli, Gabriele. Classic Tiles. Digital image. 2048. World Wide Web, Mobile Phone, Nintendo 3DS, n.d. Web. 5 June 2016. [↑](#footnote-ref-4)
5. 2048. 2048. N.p., n.d. Web. 06 June 2016, from http://allthe2048.com/ [↑](#footnote-ref-5)
6. Romain Cousin. Animated Tiles. Digital image. 2048. World Wide Web, Mobile Phone, Nintendo 3DS, n.d. Web. 5 June 2016. [↑](#footnote-ref-6)
7. Microsoft Incorporated. RPG 2048. Digital Image. n.d. Web. 5 June 2016, from https://www.microsoft.com/en-us/store/apps/2048-rpg-exclusive/9nblggh2wdds [↑](#footnote-ref-7)
8. National Geographic. Digital image. National Geography. National Geographic Partners, LLC., n.d. Web. 6 June 2016. [↑](#footnote-ref-8)
9. 2048 Game - How to Play Tutorial Instructions 2048. 2048 Game. N.p., n.d. Web. 06 June 2016. [↑](#footnote-ref-9)
10. 2048 • Free on the Mac App Store. Mac App Store. N.p., n.d. Web. 06 June 2016. [↑](#footnote-ref-10)