Matching Game - Memory Challenge

Development report - Luping Yu - ly15516

1 Introduction

This game is my deliverable for the Graphics assignment. The main reason for developing a game is that I want to practice with the Java's graphics facilities. Besides, I'd like to use the OO design idea which I learned from the course to build it. Now before reviewing the story of my development of this matching game, some features should be pointed out:

1. The development tools are Atom and Java SE 8
2. Java Layout Manager is MigLayout

The reason I don't want to use IDE is that I want to make sure I know each line code of my program, but IDE will automatic generate some "filling" or regenerated your code which I don't like. Another important reason is that Ian also said that IDE facilities are usually poor, it will produce non-window-size-independent code or poor code. In this case, Atom (the original code editor) is my choice for this development.

For the Layout Manager, Ian described how to use the standard library classes and MigLayout library clearly in the slide 8. Personally, I prefer MigLayout since its much easier to adjust the layout which may changed by me sometimes.

How to run it javac Mgame.java, java Mgame

How to play it Click -About- button after load into game to find the way to play.

2 Stage Summary

Stage 1 In this stage, I extend the example from the lectures (Layout2.java) as the main class and designed the original layout of this game. The left half of GUI is game zone and the right half is menu zone. I used the logo of university as icon to fill the game zone as an example. Besides, I choose the debug mode of MigLayout so I can identify my layout immediately.

Add Display.java, Mgame.java, Grid.java (Empty)

Todo Complete Grid.java to make the logic module behind game works properly. And try to display different kinds of icons in game zone.

Stage 2 In this stage, I collect a set of pictures about the club logo of The Premier League. Then I used GIMP to normalize them to 80x80 size icons. To display them in the game zone randomly, I used Collections.shuffle to randomly generate a number list and then transfer it to 2d integer array as the logic grid of game.

Add Grid.java, icon(foulder)

Todo Try to hide icons at first, then click them to turn outside in.

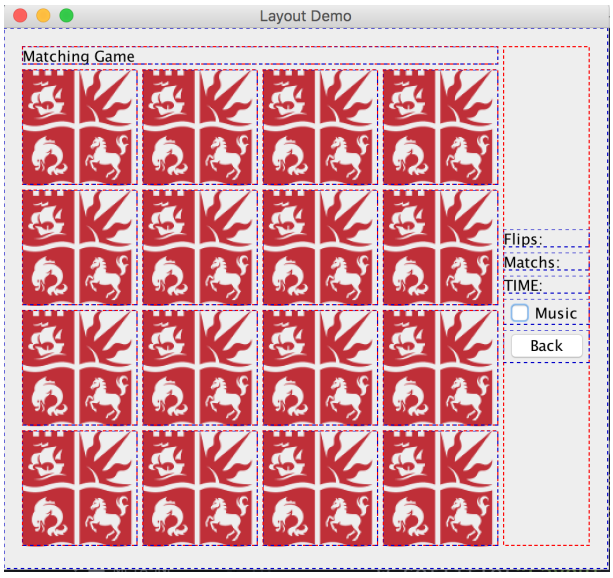

Figure 1: Stage 1

Figure 2: Stage 2

Stage 3 To click icons then turn it outside in, definitely I need to listen the mouse click. In this case I "lent" the example from the lectures (MouseClicker.java) again. In the example Ian listen the coordinate of click then find which parts of the window has been clicked. However, its not so good because it cant deal with zoom.

I decided to listen to the click event of each JLabel to figure out which one has been clicked. However, our game has 12 pairs of icons which mean I have to initialize 24 JLabels with different name and then using e.getSource().getText() to find which JLabel has been clicked. To avoid it, I decided extend the JLabel class then add the sequence number in it to help me figure out which one has been clicked. This kind of extension is MyLabel.java.

Soon I found that after I change the image of JLabel, it wouldnt update automaticly and updateUI(), repaint(), revalidate() and validate() dont work at all. I did some experiment about it and I found that I can put the MigLayout into the public void paintComponent(Graphics g0)

and use repaint() to refresh it. However, it is incredible slow. Finally, I found that I have to removeAll() and then reset the MigLayout then updateUI(), it works!

Add MouseClicker.java, MyLabel.java

Todo Hold the icons if the two successive clicks turn the same kind of icons outside in.

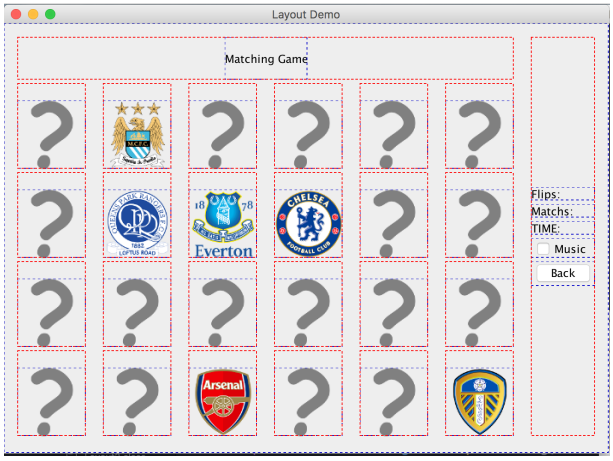

Figure 3: Stage 3

Stage 4 To hold the icons, I develop the Grid.java, add a new 2d integer array to copy the items from original array if the successive clicks turn the same kind of icons outside in. And the Display.java only get the new array from Grid.java. The original array is private in Grid.java.

Besides, the flips and matchs counter had been added to record game data.

Add Flips(Label), Matchs(Label)

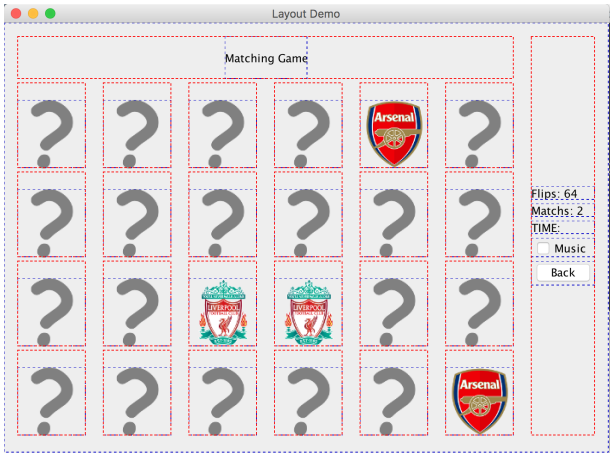

Figure 4: Stage 4

Stage 5

Add Reload (Button), Popup message when you win (New Feature)

Todo Timer function

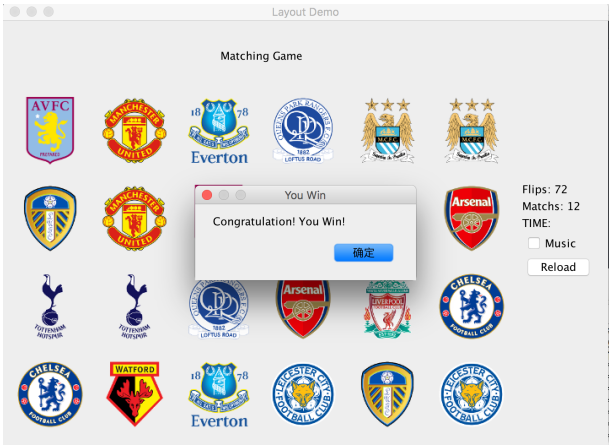

Figure 5: Stage 5

Stage 6 In this stage, I try to add a Timer to count the time that player spend for the game. However, I dont find a way to update a specific widget in MigLayout so far. To avoid refresh the whole window after each time period, I have to put the time Label to the bottom of the window, then used this.remove(), this.add(), this.updateUI() to update this widget only.

Add Timer function

Todo Audio function

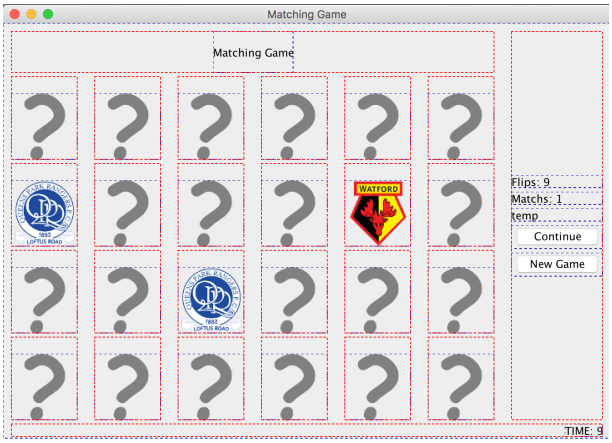

Figure 6: Stage 6

Stage 7 To add audio function, I did some experiment about how to read audio file and play it. In the end I found that java.applet.AudioClip is the best way to do what I want. For testing if audio function works properly, I downloaded "AngryBirds.mid" which is the background music of Angry Bird as the temporal background music of this game.

Add Audio.java

Todo Background image

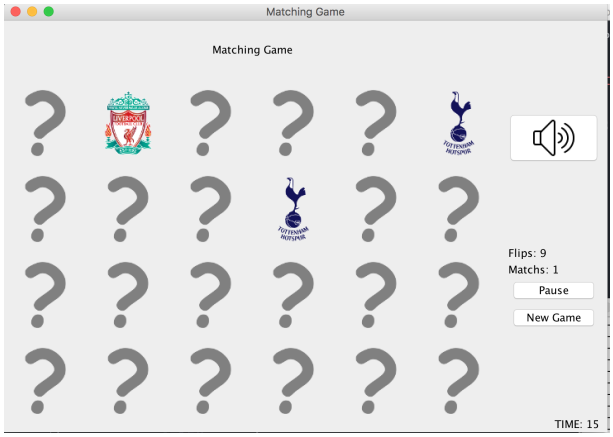

Figure 7: Stage 7

Stage 8 In this stage, I added background image by overwrited the protected void paintComponent(Graphics g), its not so difficult.

Add Background image

Todo Backgournd selection buttons

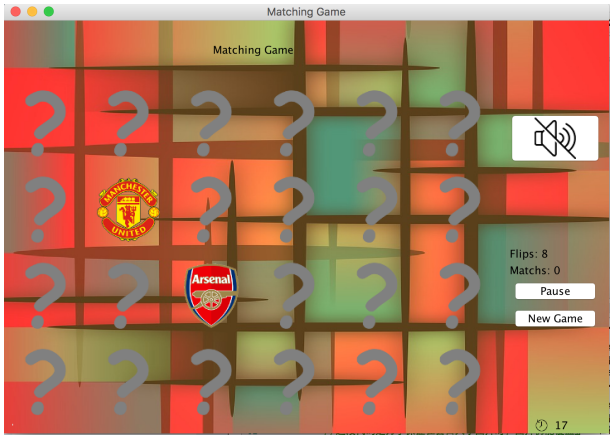

Figure 8: Stage 8

Stage 9 I rebuild the structure of my program, separate Image.java to read image and Clock.java to record time from Display.java. Then, I added three buttons to change background image. In particular, the Gradient background is not insert an image, it draws a mixture of two colors as background.

Add Background selection buttons

Todo Start menu and more levels

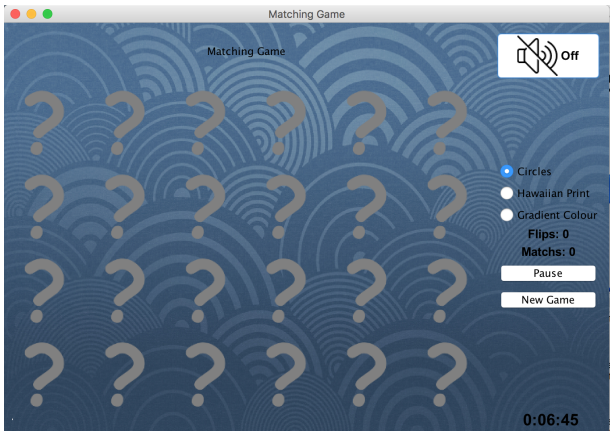

Figure 9: Stage 9

Stage 10 To add start menu, I add a new MigLayout as the start menu user interface. When player click the level selection button, this program will replace the start menu layout to game layout. In this case, add more levels is quite easy, just replace the icons, problem solved.

Add Start menu, Three levels

Final stage Reconstruct code and add comments. Actually I always reconstruct my code in each stage to make sure that each class is responsible for what I want it to do. In the end, I am pretty proud to see that my program works as my plan, I think it is time to stop at this stage.

Sources From Background Music: @AngryBirds, Icons: @Google Image