Petrit Krasniqi T00192719

JaVA CA  
Pac-Man Game

Table of Contents

[Requirement Spec 2](#_Toc499750729)

[Pak – Man Game: 2](#_Toc499750730)

[Graphics: 2](#_Toc499750731)

[Moving the Images: 3](#_Toc499750732)

[Game Rules 3](#_Toc499750733)

[Class Diagram 5](#_Toc499750734)

[VOPC Diagram 6](#_Toc499750735)

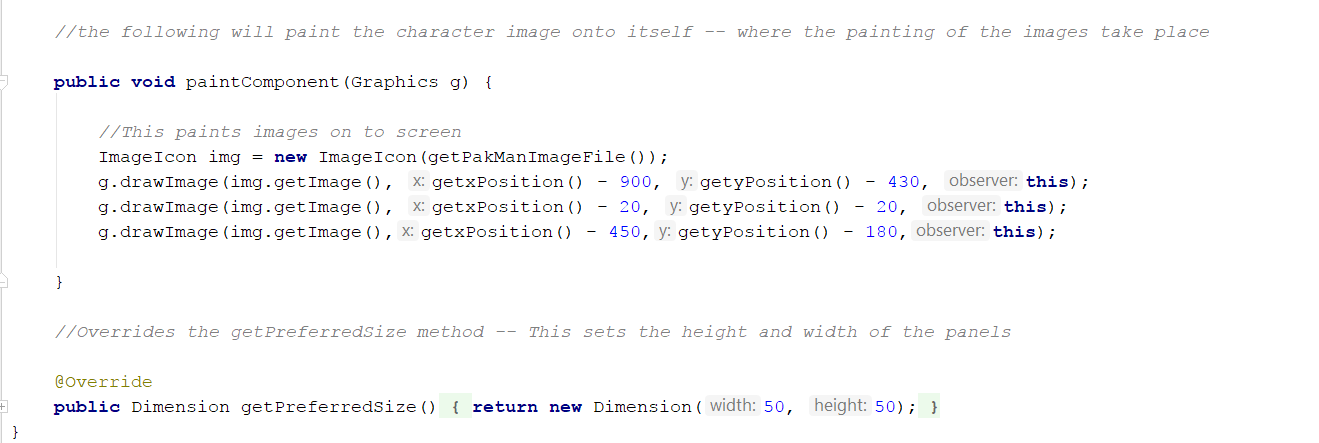
# Requirement Spec

## Pak – Man Game:

My java project is a simple two player game which uses the Pac-Man characters where one player controls the ghost and the other controls Pac-Man. The goal of the game is for Pac-Man to move around the screen and collect as many coins as possible before the player controlling the ghost catches him.

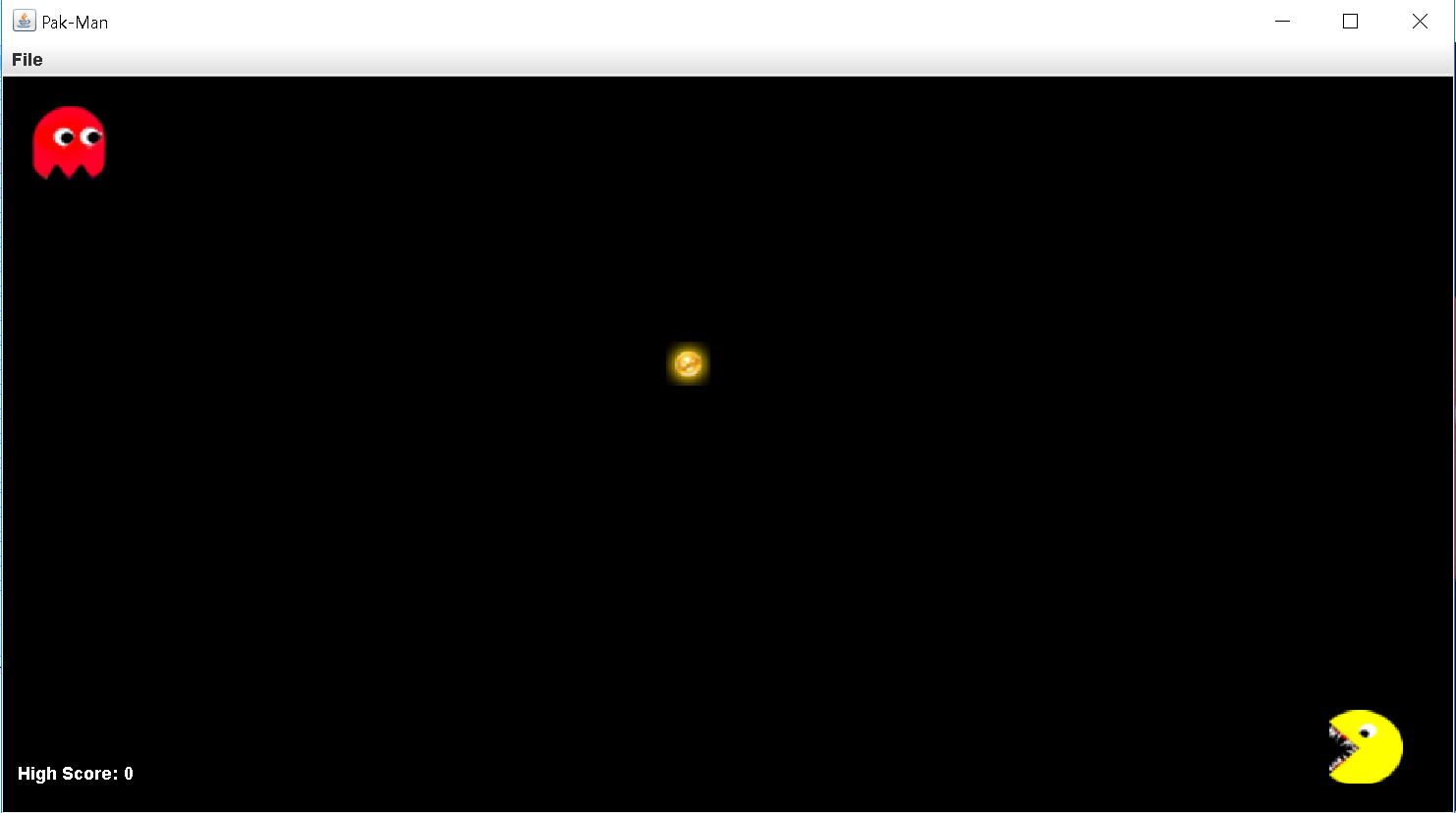
## Graphics:

I animated the bodies of the characters with four images each depending on the direction. To create my images I gave them the following attributes: int xPosition, int yPosition, String ImagePathFile. The two int values store the values for where the image will be positioned on the screen and the string stores the image file path. The reason I gave the image a string storing it’s image path because to create the images I used ImageIcon which is an implementation of the Icon interface that paints icons from images by their file path name and reads it as a string. I override the paintComponent (Graphics g) method to draw and position my image onto the screen by using the g.drawImage . G.drawImage takes in an image and sets an X and Y position for it on the window. Graphics G is a reference variable for an object of graphics. I the override the getPrefferedSize method to set my desired size of the JPanels which was 50 x 50.



As you can see I used the getImage() method to get the image path stored in the ImagePath String variable and used the getters method for the xPosition and yPosition variables to set the location of the image onto the screen. I then created another JPanel called ContentPane in my GUI class to hold both my images. I then added a JFrame to hold the contentPane JPanel to display my Images on screen.

Screenshot of my images on JFrame



## Moving the Images:

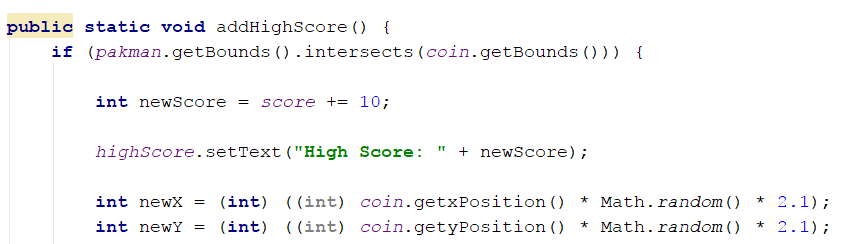
To move the images I created a class called KeyBoardAnimation which implements the KeyListener interface to receive keyboard events(keystrokes). I used a hashmap data structure to keep track of all the key presses and releases that have occurred.

In this class we capture the key stroke the user selected (E.g. LEFT key). When the user selects a particular key the object’s x and y position is changed.

## Game Rules

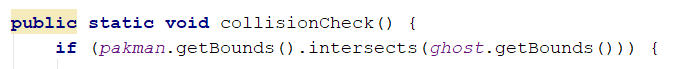
The rules of the game are that Pac – Man must collect as many coins without hitting the wall or without getting caught by the ghost. The ghost must avoid the wall also.

When the Pac-Man hits the coin the players score increases. To check for this is I created a method called addHighScore(). This checks to see if the pakMan image intersects the coin image from the get bounds method(). This method also moves the coin to a random new location. To do this I set the coins location on the x and y axis by multiplying its current x and y location by math.random by 2.1.

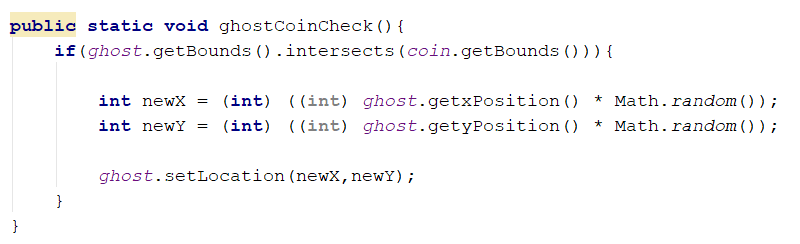


If Pac-Man or the ghost hit the wall they die and the game is over. To do this I checked to this if their position on the x-axis is less than 0 or greater than the frame width and that their position on the y -axis is less than 0 or greater than the frame height.

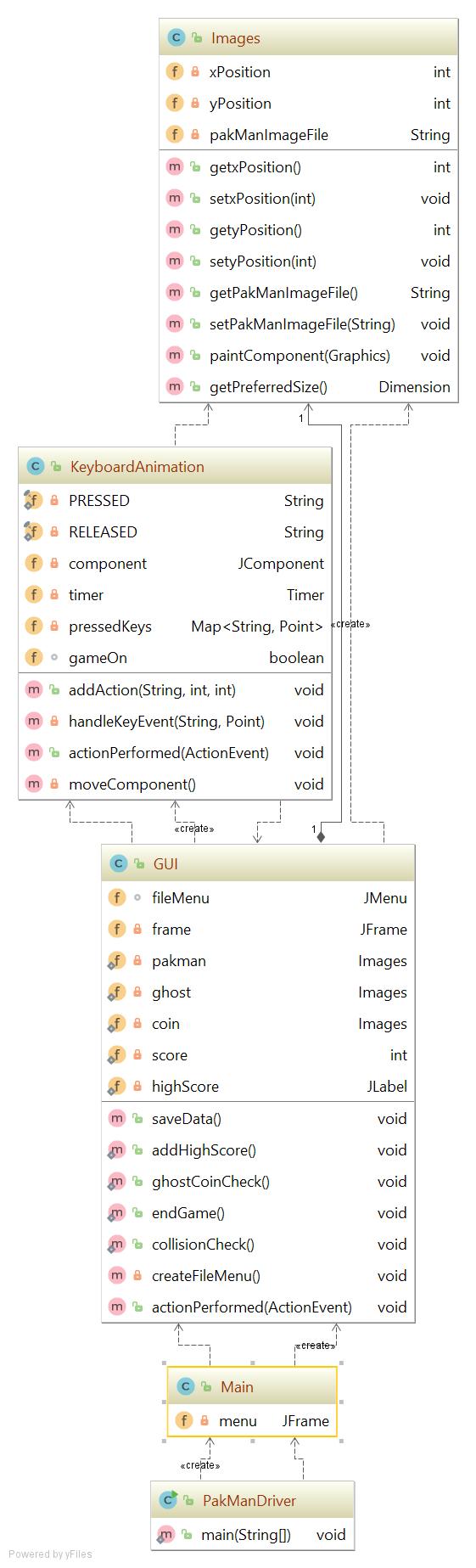
If Pac-Man and the ghost collide the game is over. To check for this I created a method called collision check which checks if the ghost and pac – man intersect using the getBounds method.



To avoid the ghost staying near the coin to wait and capture pac – man easily I created a method called ghostCoinCheck which checks to see if the ghost is near the coin from the getbounds method and if he is the ghost is moved to a new random location.



# Class Diagram



# VOPC Diagram

