**Journal Entrys**

**Space Adventure Version 1:**

Entry 1, Date: March 19, 2013:

Today I started messing around with frames and panels. I went on our class website and started testing out multiple snippets of code posted. I managed to display a button, change a label with the help of the button using this bit of code :

public static void main(String[] args) {

JFrame frame = new JFrame();

JButton btnClick = new JButton("Click me!");

frame.add(btnClick);

frame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

class ClickOutput implements ActionListener {

public void actionPerformed(ActionEvent event) {

System.out.println("I was clicked. Yeah");

}

}

ActionListener clickOutput = new ClickOutput();

btnClick.addActionListener(clickOutput);

}

private static final int FRAME\_WIDTH = 100;

private static final int FRAME\_HEIGHT = 60;

}

Entry 2, Date: March 20, 2013:

I created a new project called Space Adventure V.1 where I intended to create the very first version of my graphics project. I successfully loaded up a frame. Within this frame I created two new panels. First one is called panMenu which will be my main panel. Within this panel I made and added a new panel called panSouth which contains all three of my buttons. I set panSouth to grid layout for a nice smooth flow of three buttons. I did it this way :

panSouth.setLayout(new GridLayout(3, 1));

I found how to set up gridlayout on our class website.

Entry 3, Date: March 21, 2013:

I created a new jPanel class called PanMaiin. Unfortunately, I was unable to name it “PanMain” without the double Is. This was probably because I had initially created a new class called PanMain but decided to delete it since I felt I had not use for it at the moment. My guess is that it did not completely delete the class from the project. I found how to set it up on our class website. This panel will be placed in the center region of panMenu which is the ‘mother’ panel. I set the class up this way:

import javax.swing.JPanel;

public class PanMaiin extends JPanel { // extend JPanel meaning the class is a panel

public PanMaiin() {

}

}

**Space Adventure Version 2:**

Entry 4, Date: March 24, 2013:

I plan on making this version a busy one. Today I managed to load up two images (one on top of the other). This was a very difficult task for me. It took a lot of research at home to get this working. Initially I was able to get only one image loaded but eventually I came across this website that taught me how to load up multiple images at once <http://zetcode.com/tutorials/javagamestutorial/basics/>

I found this website on our class website when Patrick’s code was demonstrated.

Entry 5, Date: March 25, 2013:

The final challenge was to load up an image up my spacecraft. This was a little challenging since I wanted a class of spacecraft that could return its position, update its position and so on. I once again checked zetcode and in the basics link, I found a neat way to initialize this class. Here is how I did it :

public Spacecraft() {

ImageIcon ii = new ImageIcon(this.getClass().getResource(scraft));

image = ii.getImage();

nx = 40;

ny = 60;

}

public int getX() {

return nx;

}

public int getY() {

return ny;

}

public Image getImage() {

return image;

}

In this code I have a constructer that has the image of my spacecraft and its initial position. Then I have two critical methods that return the x and y position of the spaceship. This was a must needed example that I came across and I hope to develop new classes for each of my image components in the next release

**Space Adventure Version 3:**

Entry 6, Date: March 24, 2013:

Since the due date is just in couple of days, I had to make this version a busy version. Today I managed to make my spaceship moves on the screen when pressed the arrow keys. This was a fairly quick process since I had an excellent source to copy this bit of code from; ZetCode. This website had a tutorial on how to get a sprite to move around the screen. I learned that this requires the use of timer. Every 50 ms, these functions are called

private class TAdapter extends KeyAdapter {

@Override

public void keyReleased(KeyEvent e) {

craft.keyReleased(e);

}

@Override

public void keyPressed(KeyEvent e) {

craft.keyPressed(e);

}

}

If a key is pressed or released, this function will update a variable called ndX which is the displacement of the spacecraft. Later this method is called which will update the displacement of the spacecraft:

public void move() {

nx += ndx;

ny += ndy;

}

Entry 7, Date: March 25. 2013:

Today during my 2nd period spare I managed to get an asteroid moving across the screen. I realized that I had to make another timer that will update its position every 50 ms. I got this timer idea but I lacked the concept of setting up a time. This is when I ran across this website:

<http://www3.ntu.edu.sg/home/ehchua/programming/java/J4b_CustomGraphics.html#animation>. I copied this bit of code:

int delay = 500; // milliseconds

// Create an instance of an anonymous subclass of ActionListener

ActionListener updateTask = new ActionListener() {

@Override

public void actionPerformed(ActionEvent evt) {

// ......

}

};

// Start and run the task at regular delay

new Timer(delay, updateTask).start();

This code throughtly explained how to set up a timer and the use of the timer. I was able to modify this code in this fashion:

ActionListener updateTask = new ActionListener() {

@Override

public void actionPerformed(ActionEvent evt) {

update(); // update the (x, y) position

repaint(); // Refresh the JFrame, callback paintComponent()

}

};

new Timer(nUPDATE\_PERIOD, updateTask).start();

This code will call a function called update that will update the asteroids position and will repaint the asteroid on the screen.

**Space Adventure Version 5:**

Entry 8, Date: March 26-. 2013:

Today I stayed up late and managed to get collision detection going. It was a very quick process. I looked at the class website and the way collision detection was done there was a little messy to read and process so I decided to Google search collision detection. I came across this (<http://stackoverflow.com/questions/335600/collision-detection-between-two-images-in-java>) question asked on Stackoverflow. I saw this bit of code that was really helpful:

foreach (Enemy e in EnemyCollection)

{

Rectangle r = new Rectangle(e.X,e.Y,e.Width,e.Height);

Rectangle p = new Rectangle(player.X,player.Y,player.Width,player.Height);

// Assuming there is an intersect method, otherwise just handcompare the values

if (r.Intersects(p))

{

// A Collision!

// we know which enemy (e), so we can call e.DoCollision();

e.DoCollision();

}

}

This code snippet was written in C# but the syntax was the exact same for java (aside from the loop of course). I was able to modify this code in this manner to get my collision detection working with 1 asteriods

public void collision() {

Rectangle rect1 = new Rectangle(craft.getX(), craft.getY(),

50, 30);

Rectangle rect2 = new Rectangle(aste.getX(), aste.getY(), 50, 30);

if (rect1.intersects(rect2)) {

aste.setX(1001);

aste.setY(craft.getY());

craft.setX(0);

craft.setY(0);

}

}

Initially it was difficult to judge how big the rectangles should be. After multiple trial and errors, I decided they should both be 50 widths and 30 heights. I also made it so that the craft is set to its initial position after collision and the asteroid is also put back to its initial position. Furthermore, during my spare I was able to get the space background moving. It took me a little time to think about how I was going to get it working but I finally figured that I could use the same timer to update my background. I would have to run two parallel space backgrounds at once. When one is sliding across the screen, the other will immediately follow.

**Space Adventure Version 6 & 7:**

Entry 8, Date: March 28-. 2013:

Due to time constraints, this version will have to be the final version of my graphics project. Today, before bed time, I decided to look for a way to load multiple asteroids at once. I went back to this (<http://zetcode.com/tutorials/javagamestutorial/movingsprites/>) link. Here I scrolled down and saw that they had a code demo for how to shoot missiles. Even though the idea of shooting was no use for me, I decided to modify their idea of shooting missiles to make multiple asteroids move across the screen. I copied this bit of code

ArrayList ms = craft.getMissiles();

for (int i = 0; i < ms.size(); i++ ) {

Missile m = (Missile) ms.get(i);

g2d.drawImage(m.getImage(), m.getX(), m.getY(), this);

}

ArrayList ms = craft.getMissiles();

for (int i = 0; i < ms.size(); i++) {

Missile m = (Missile) ms.get(i);

if (m.isVisible())

m.move();

else ms.remove(i);

}

And modified it this way :

ArrayList arlAste = spCraft.getAsteroids();

for (int i = 0; i < arlAste.size(); i++) {

Asteriods asteroids = (Asteriods) arlAste.get(i);

g2d.drawImage(asteroids.getImage(), asteroids.getX(), asteroids.getY(), this);

}

ArrayList arlAste = spCraft.getAsteroids();

for (int i = 0; i < arlAste.size(); i++) {

Asteriods asteroid = (Asteriods) arlAste.get(i);

}

if (asteroid.isVisible()) {

asteroid.update();

} else {

arlAste.remove(i);

}

}

This got me to move multiple asteroids across the screen. Now the tough part remains: collision detection.

Entry 9, Date: March 29, 2013:

Since the due date is the day after tomorrow, I had to give final touches to my graphics project. Today I managed to get collision detection going with multiple asteroids. I added the following code to the project:

Rectangle rctCraft = new Rectangle(spCraft.getX(), spCraft.getY(),

50, 30);

Rectangle rctAste = new Rectangle(asteroid.getX(), asteroid.getY(), 50, 30);

if (rctCraft.intersects(rctAste)) {

spCraft.setX(0);

spCraft.setY(0);

arlAste.clear();//clear all asteroids

if ((timerAste.getDelay() - 100) > 0) {

timerAste.setDelay(timerAste.getDelay() - 50);

}

}

This bit was placed within the loop which removes/updates the position of the asteroids. This was a big success for me since I didn’t imagine myself being able to finish collision detection within the time limit.

Entry 10, Date: March 29, 2013:

Since the collision detection took a while, I had some leftover components that were yet to be added. I added a health bar, asteroids bar, new method of bounds check as well as a timer that times how long the user was able to survive in the game. I also managed to put the splash screen and menu buttons up and running.