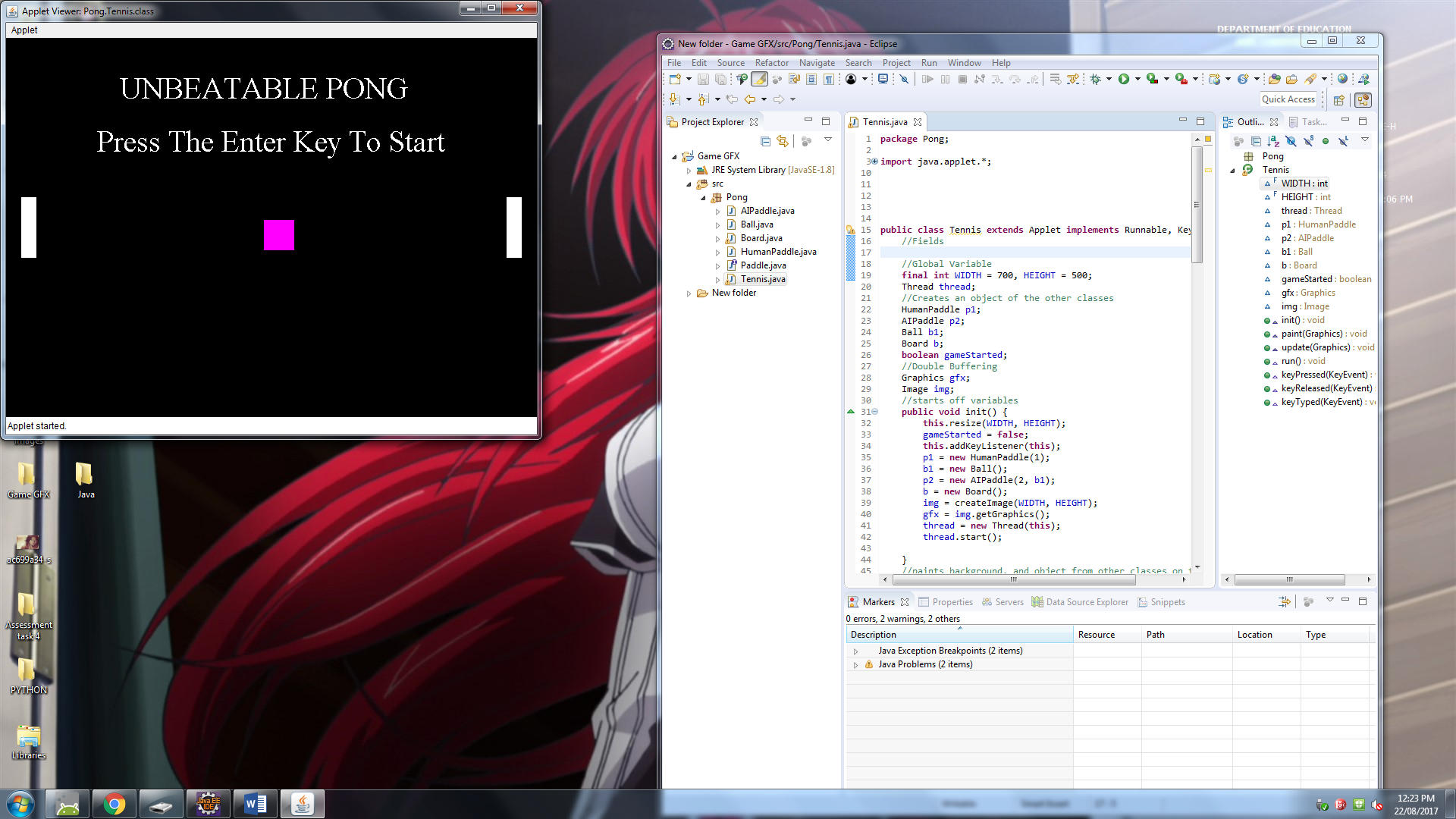
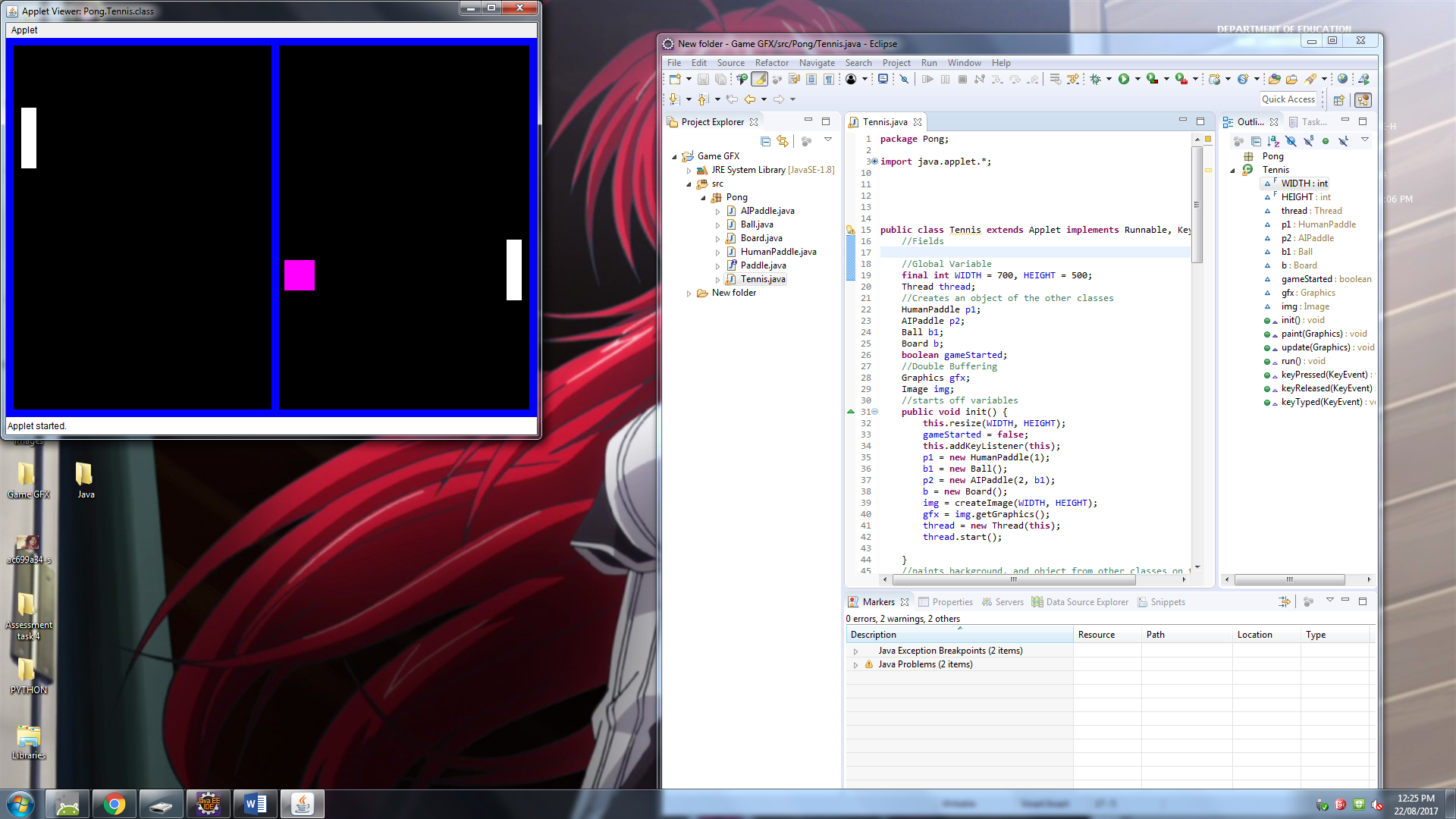
Game Screen Shots

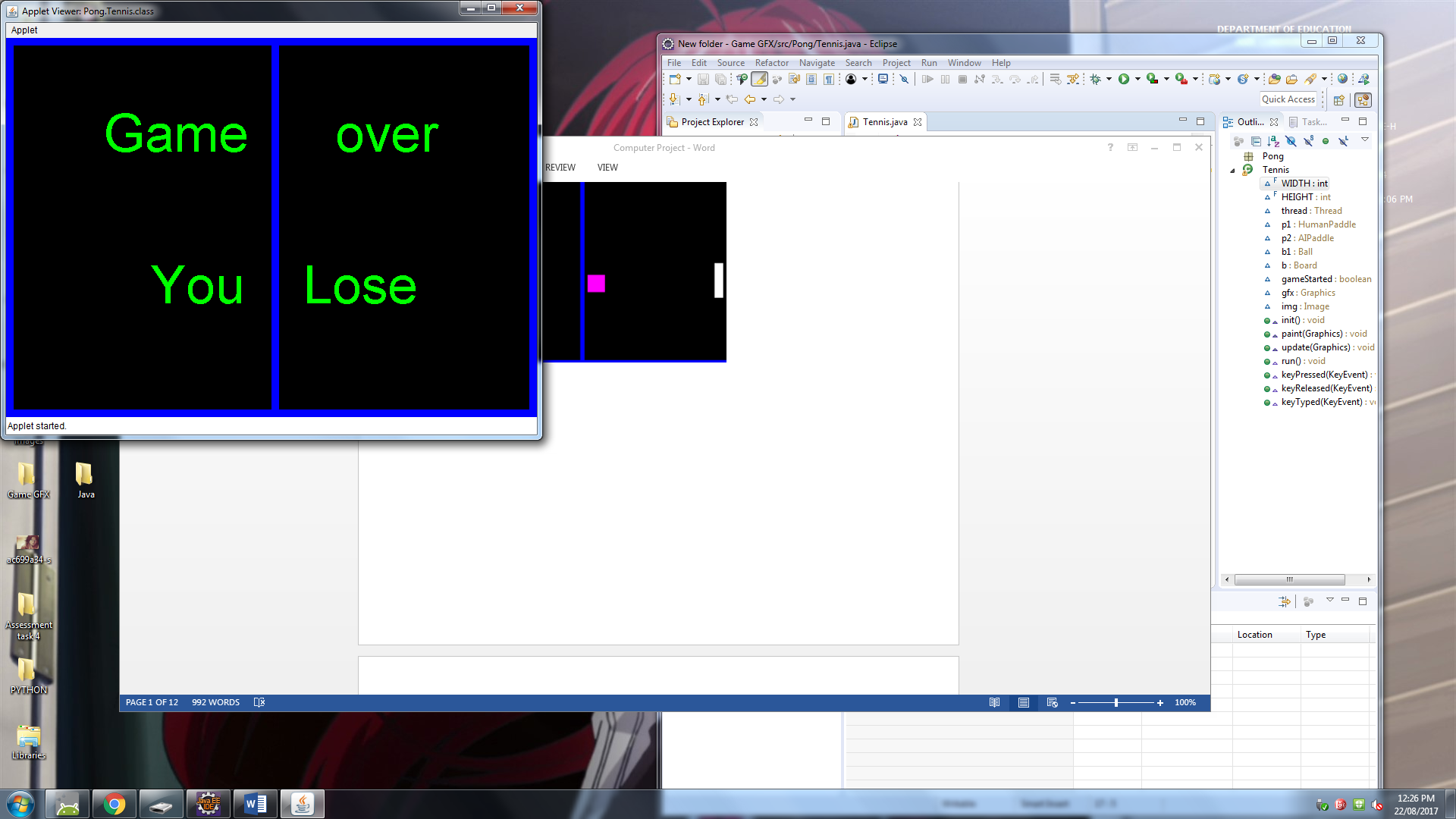
Intro



Game in action



Death Screen



Tennis.Java (main class)

package Pong;

//java is importing data for defined repositories

import java.applet.\*;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

public class Tennis extends Applet implements Runnable, KeyListener{

**//fields / Global Variables.**

final int WIDTH = 700, HEIGHT = 500;

Thread thread;

HumanPaddle p1;

AIPaddle p2;

Ball b1;

Board b;

boolean gameStarted;

Graphics gfx;

Image img;

**//initial variables.**

public void init() {

this.resize(WIDTH, HEIGHT);// WIDTH and HEIGHT taken from the final int above.

gameStarted = false;

this.addKeyListener(this);// initials keyListener to the class 'Tennis'.

p1 = new HumanPaddle(1);

b1 = new Ball();

p2 = new AIPaddle(2, b1);

b = new Board();

img = createImage(WIDTH, HEIGHT);//Double buffering.

gfx = img.getGraphics();//Double buffering.

thread = new Thread(this);

thread.start();//starts thread. Thread controls the programs timing.

}

**//Draws defined object on the screen.**

public void paint(Graphics g) {

//Background

gfx.setColor(Color.black);

gfx.fillRect(0, 0, WIDTH, HEIGHT);

//Game over/Death Screen code.

if (b1.getX() < -10 || b1.getX() > 710) {

**//if the ball touches the end zone then initial this code.**

gfx.setColor(Color.GREEN);

gfx.setFont(new Font("Apple Casual", Font.PLAIN, 70));

gfx.drawString(" Game over", 110, 150);

gfx.drawString("You Lose", 190, 350);

}

else {

**//if the ball doesn't touches the end zone then initial this code.**

p1.draw(gfx);

b1.draw(gfx);

p2.draw(gfx);

}

**//Start Screen**

**// At the start of the game this will happen.**

if(!gameStarted) {

gfx.setColor(Color.WHITE);

gfx.setFont(new Font("TimesRoman", Font.PLAIN, 40));

gfx.drawString("UNBEATABLE PONG", 150, 80);

gfx.drawString("Press The Enter Key To Start", 120, 150);

}

**//else draw the borders.**

else {

b.draw(gfx);

}

**//drawing double Buffering.**

g.drawImage(img, 0, 0, this);

}

//update paint method.

public void update(Graphics g) {

paint(g);

}

public void run() {

for(;;) {

**//if game started = true, initial move and check paddle collision methods from ball, AlPaddle and HumanPaddle.**

if(gameStarted) {

p1.move();

p2.move();

b1.move();

b1.checkPaddleCollision(p1, p2);

}

repaint();

**//Thread timing.**

try {

Thread.sleep(10);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

**//KeyListener code. Uses predefined variables in linked classes to move p1(paddle 1).**

public void keyPressed(KeyEvent e) {

if(e.getKeyCode() == KeyEvent.VK\_UP) { //Move up, press the up key which changes the UpAccel to true.

p1.setUpAccel(true);

}

else if(e.getKeyCode() == KeyEvent.VK\_DOWN) {

**//Move down, press the down key which changes the DownAccel to true.**

p1.setDownAccel(true);

} else if(e.getKeyCode() == KeyEvent.VK\_ENTER) {

**//To set gameStarted which starts the program.**

gameStarted = true;

}

}

public void keyReleased(KeyEvent e) {

**//When up key is released stop UpAccel.**

if(e.getKeyCode() == KeyEvent.VK\_UP) {

p1.setUpAccel(false);

}

else if(e.getKeyCode() == KeyEvent.VK\_DOWN) { //When down key is released stop DownAccel.

p1.setDownAccel(false);

}

}

public void keyTyped(KeyEvent e) {

}

}

**Ball.Java**

package Pong;

import java.awt.Color;

import java.awt.Graphics;

public class Ball {

**//Fields and Global variables.**

double xVel, yVel, x , y;

**//Constructor.**

public Ball() {

x = 350;

y = 250;

yVel = getRandomSpeed() \* getRandomDirection();

xVel = getRandomSpeed() \* getRandomDirection();

}

**//Speed of the ball when starting.**

public double getRandomSpeed() {

return(Math.random()\*3 + 2);

}

**//Direction of the Ball when starting.**

public int getRandomDirection() {

int rand = (int)(Math.random() \* 2);

if(rand == 1)

return 1;

else

return -1;

}

**//Drawing ball on the screen.**

public void draw(Graphics g) {

g.setColor(Color.MAGENTA);

g.fillRect((int)x-10, (int)y-10, 40, 40);

}

**//Giving the Paddles the ability to deflect the ball and at what angle.**

public void checkPaddleCollision(Paddle p1, Paddle p2) {

if( x<= 50) {

if( y >= p1.getY() && y <= p1.getY() + 80)

xVel = -xVel;

}

else if(x >= 640){

if(y >= p2.getY() && y <= p2.getY() + 80)

xVel = -xVel;

}

}

**//Settings the board logic and collisions.**

public void move() {

x += xVel;

y += yVel;

if (y < 10)

yVel = -yVel;

if (y > 490)

yVel = -yVel;

}

**//setting GetX as a int.**

public int getX() {

return(int)x;

}

//setting GetY as a int.

public int getY() {

return(int)y;

}

}

HumanPaddle.java

package Pong;

import java.awt.Color;

import java.awt.Graphics;

public class HumanPaddle implements Paddle { //Implements Paddle Interface form Paddle

**//Fields and Global variables**

double y, yVel;

boolean upAccel, downAccel;

final double GRAVITY = 0.94;

int player, x;

**//Constructor.**

**//Starting positions of HumanPaddle.**

public HumanPaddle(int player) {

upAccel = false; downAccel = false;

y = 210; yVel = 0;

if(player == 1)

x = 20;

else

x = 680;

}

**//Drawing Paddle.**

public void draw(Graphics g) {

g.setColor(Color.WHITE);

g.fillRect(x, (int)y, 20, 80);

}

**//Move method, establishing movement speeds and gravity.**

public void move() {

if(upAccel) {

yVel -= 2;

}

else if (downAccel) {

yVel += 2;

}

else if (!upAccel && !downAccel) {

yVel \*= GRAVITY;

}

if(yVel >= 5)

yVel = 5;

else if(yVel <= -5)

yVel= -5;

y += yVel;

if(y < 0)

y = 0;

if(y > 420)

y = 420;

}

public void setUpAccel(boolean input) {

upAccel = input;

}

public void setDownAccel(boolean input) {

downAccel = input;

}

**//Setting getY as a int to use in this form.**

public int getY() {

return (int)y;

}

}

AIPaddle.java

package Pong;

import java.awt.Color;

import java.awt.Graphics;

public class AIPaddle implements Paddle {

**//Fields and Global variables**

double y, yVel;

final double GRAVITY = 0.94;

int player, x;

Ball b1;

**//Constructor**

public AIPaddle(int player, Ball b) {

b1 = b;

y = 210; yVel = 0;

if(player == 1)

x = 20;

else

x = 660;

}

**//Drawing the ALPaddle.**

public void draw(Graphics g) {

g.setColor(Color.WHITE);

g.fillRect(x, (int)y, 20, 80);

}

**//Making the AIPaddle follow the ball.**

public void move() {

y = b1.getY() - 40;

if(y < 0)

y = 0;

if(y > 420)

y = 420;

}

**//Setting getY as a int.**

public int getY() {

return (int)y;

}

}

Paddle.java

package Pong;

import java.awt.Graphics;

public interface Paddle {

public void draw(Graphics g);

public void move();

public int getY();

}

Board.java

package Pong;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Graphics;

public class Board {

**//Drawing Boards around the game.**

public void draw(Graphics g) {

g.setColor(Color.blue);

**//Vertical lines**

g.fillRect(350, 0, 10, 500);

g.fillRect(0, 0, 10, 700);

g.fillRect(690, 0, 10, 500);

**//horizontal lines**

g.fillRect(0, 0, 700, 10);

g.fillRect(0, 490, 700, 10);

}}