**รายงาน Mini Project**

วิชา Object Oriented Programming

Game : Among us jump!



จัดทำโดย

นาย อินทัช ชัยรักษ์สิริกุล

รหัสประจำตัวนักศึกษา 6104062636181 (ชั้นปีที่ 3)

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

**คำนำ**

โปรเจกต์เกมAmong us Jump! เป็นส่วนหนึ่งของการเรียน วิชา Object Oriented Programming ข้าพเจ้าตั้งใจจัดทำขึ้นเพื่อนำเสนอวิธีการเขียนโปรแกรมภาษาJava ด้วยโปรแกรม NetBeans 8.2 โดยนำความรู้จากสิ่งที่ข้าพเจ้าได้เรียนมาทั้งหมดมาสร้างเป็นผลงานชิ้นนี้

ผู้จัดทำจึงต้องขอขอบพระคุณอาจารย์สถิต ประสมพันธ์ ที่ให้ความรู้เกี่ยวกับการทmini Project ครั้งนี้ ผู้จัดทำหวังเป็นอย่างยิ่งว่าผู้ที่ใช้โปรแกรมนี้จะได้รับความรู้และประโยชน์จากโปรเจกต์ชิ้นนี้ ถ้าโครงงานเล่มนี้มีสิ่งใดผิดพลาดหรือไม่ถูกต้อง ขอ อภัยไว้ ณ ที่นี้ด้วยครับ

ผู้จัดทำ

นาย อินทัช ชัยรักษ์สิริกุล

บทที่ 1

**บทนำ**

**ที่มาและความสำคัญของโปรเจค**

ที่มาเกิดจากอาจารย์ได้ให้ไปศึกษาเกี่ยวกับเรื่องGUI และ Event handling โดยการให้สร้างโปรแกรมหรือเกมขึ้นมาโดยประยุกต์จากสิ่งที่อาจารย์เคยได้สอนและให้ลองศึกษา สิ่งสำคัญของโปรเจคชิ้นนี้คือการนำเนื้อหาต่างมามาประยุกต์ใช้และปรับปรุงแก้ไขจนเป็นงานชิ้นนี้

**ประเภทของโครงการ**

เป็นโครงการประเภทที่นำGUIมาใช้ในการออกแบบเกมอย่างง่าย

**ประโยชน์**

ได้ฝึกเขียนโค้ดที่เป็นงานชิ้นใหญ่ รู้จักการวิเคราะห์การทำงานของโปรแกรม การปรับปรุง แก้ไข ส่วนต่างๆให้ดีขึ้น

**ขอบเขตการทำงาน**

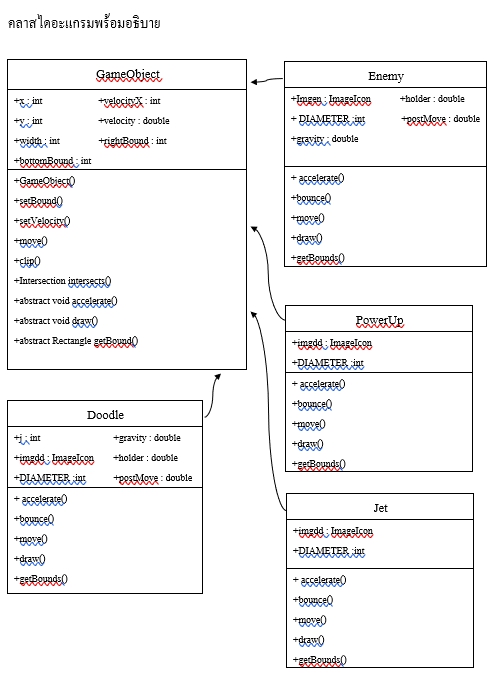
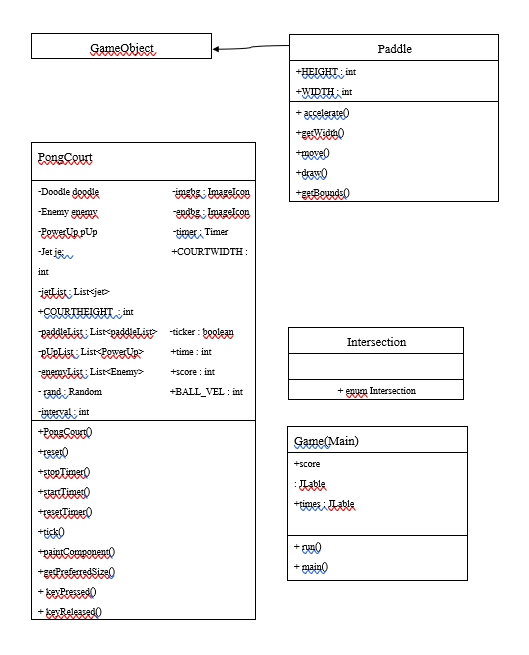
พัฒนาโปรแกรมด้วยภาษาJavaและใช้โปรแกรมNetBeans ในการเขียน โดยกำหนดวิธีการเขียนให้เป็นแบบ Object Oriented Programming โดยใช้เรื่อง GUI ในการทำงานชิ้นนี้

บทที่ 2

การพัฒนา

**เนื้อเรื่องย่อและวิธีการเล่น**

 เป็นการจำลองตัวละคร Among us jump ให้กระโดดไปแท่นสีเทาที่มี โดยวิธีการเล่นคือใช้ปุ่มลูกศรซ้ายขวาเพื่อบังคับทิศทางเพื่อไม่ให้ Among ตกลงบนพื้นหรือชนเข้ากับเจ้าไวรัสอวกาศ หรือเพื่อเก็บจรวดที่เป็นตัวเพิ่มพลัง โดยคะแนนของเกมนี้จะเพิ่มขึ้นตามความสูงที่สามารถกระโดดขึ้นไป และจะจบเกมก็ต่อเมื่อตัว Among us jump ตกลงสู่พื้น หรือเมื่อตัว Among us jump ชนกับไวรัสอวกาศในเกม

****

**อธิบาย Event handling**

เมื่อกดรัน เกมจะเริ่มเล่นโดยอัตโนมัติ หลังจากเริ่มเกมแล้วต้องคอยบังคับทิศทางไม่ให้ตัวละครไปชนกับเจ้าไวรัสอวกาศ บังคับตัวละครเพื่อไปเก็บจรวดที่เป็นตัวเพิ่มพลัง และต้องคอยบังคับตัวละครเพื่อไม่ให้ตกพื้นมิเช่นนั้นจะจบเกมทันที

**การพัฒนา Application**

พัฒนาโดยโปรแกรม NetBeans 8.2 โดยนำเนื้อหาทั้งหมดของวิชา Computer Programming 1 - 2 และวิชา Object Oriented Programming ทั้งหมดมาประยุกต์รวมกันเป็นผลงานชิ้นนี้

Class Game

package amongusjumpproject;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Game implements Runnable {

public static JLabel scores;

public static JLabel times;

@Override

public void run() {

// Top-level frame

final JFrame frame = new JFrame();

frame.setTitle("AMONG US Jump!!");

frame.setLocation(300,50);

frame.setVisible(true);

frame.setBackground(Color.WHITE);

// Main playing area

final PongCourt court = new PongCourt();

frame.add(court, BorderLayout.CENTER);

//resume game button

final JButton resume= new JButton("Resume game");

//final ImageIcon resume = new ImageIcon(this.getClass().getResource("Start.png"));

//final JButton resume = new JButton("Resume game");

//resume.setIcon(new ImageIcon("Start.png"));

resume.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

//w1.setVisible(false);

try {

Thread.sleep(1000);

court.restartTimer();

court.requestFocus();

resume.setVisible(false);

} catch (InterruptedException e1) {

e1.printStackTrace();

}

}

});

//Pause button

final JPanel panel = new JPanel();

frame.add(panel, BorderLayout.PAGE\_END);

final JButton Pause= new JButton("Pause");

Pause.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e){

court.stopTimer();

resume.setVisible(true);

}

});

// Reset button

final JButton reset = new JButton("New Game");

reset.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

court.startTimer();

court.reset();

}

});

panel.add(resume);

resume.setVisible(false);

panel.add(Pause);

panel.add(reset);

scores= new JLabel();

times = new JLabel();

scores.setText("Score " + court.score);

panel.add(scores);

times.setText(" Time ");

panel.add(times);

// Put the frame on the screen

frame.pack();

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

frame.setVisible(true);

frame.setResizable(false);

// Start the game running

court.reset();

}

/\*

\* Get the game started!

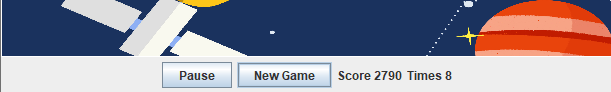
\*/

public static void main(String[] args) {

SwingUtilities.invokeLater(new Game());

}

}



Class GameObject

package amongusjumpproject;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

public abstract class GameObject {

int x; // x and y coordinates upper left

int y;

int width;

int height;

int velocityX; // Pixels to move each time move() is called.

double velocityY;

int rightBound; // Maximum permissible x, y values.

int bottomBound;

public GameObject(int x, int y, int velocityX, int velocityY, int width,

int height) {

this.x = x;

this.y = y;

this.velocityX = velocityX;

this.velocityY = velocityY;

this.width = width;

this.height = height;

}

public void setBounds(int width, int height) {

rightBound = width - this.width;

bottomBound = height - this.height;

}

public void setVelocity(int velocityX, double velocityY) {

this.velocityX = velocityX;

this.velocityY = velocityY;

}

// Move the object at the given velocity.

public void move() {

x += velocityX;

y += velocityY;

accelerate();

clip();

}

// Keep the object in the bounds of the court

public void clip() {

if (x < 0)

x = 0;

else if (x > rightBound)

x = rightBound;

//else if (y > bottomBound)

//y = bottomBound;

}

public Intersection intersects(GameObject other) {

if (other.x > x + width|| other.y > y + height|| other.x + other.width < x || other.y + other.height < y)

return Intersection.NONE;

// เซ็ทค่าให้มันอยู่ตรงกลาง

double dx = other.x + other.width /2 - (x + width /2);

double dy = other.y + other.height/2 - (y + height/2);

//ไว้ใช้คำนวนว่ามันintersectกันมั้ย

double theta = Math.atan2(dy, dx);

double diagTheta = Math.atan2(height, width);

if ( -diagTheta <= theta && theta <= diagTheta )

return Intersection.RIGHT;

if ( Math.PI - diagTheta <= theta || theta <= diagTheta - Math.PI )

return Intersection.LEFT;

if ( diagTheta <= theta && theta <= Math.PI - diagTheta )

return Intersection.DOWN;

if ( diagTheta - Math.PI <= theta && theta <= diagTheta && Doodle.postMove>Doodle.holder)

return Intersection.UP;

return Intersection.NONE;

}

public abstract void accelerate();

public abstract void draw(Graphics g);

public abstract Rectangle getBounds();

}

Class Intersection

package amongusjumpproject;

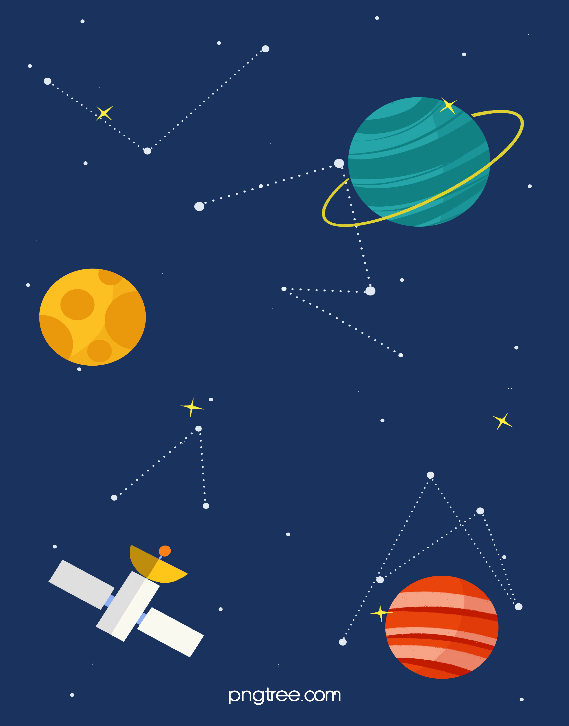
public enum Intersection {

NONE, UP, LEFT, DOWN, RIGHT

}

Class Pong Court

package amongusjumpproject;

****import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.List;

import java.util.Random;

import java.util.concurrent.\*;

public class PongCourt extends JPanel {

private Doodle doodle;

private Enemy enemy;

private PowerUp pUp;

private Jet je;

//homeplay homegames1 = new homeplay();

private final ImageIcon imgbg = new ImageIcon(this.getClass().getResource("ject4.png"));

private final ImageIcon endbg = new ImageIcon(this.getClass().getResource("over.jpg"));

private List<Jet> jetList;

private List<Paddle> paddleList;

private List<PowerUp> pUpList;

private List<Enemy> enemyList;

private List<Doodle> doodleList;

private Random rand= new Random();

private int interval = 20; // Milliseconds between updates.

private Timer timer; // Each time timer fires we animate one step.

final int COURTWIDTH = 600;

final int COURTHEIGHT = 600;

private boolean ticker=false; //Used to determine if ball is in powerup mode

public int times;

public int score;

final int BALL\_VEL = 6; // How fast does the ball move

public PongCourt() {

setBorder(BorderFactory.createLineBorder(Color.white));

setFocusable(true);

timer = new Timer(interval, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

times++;

// doodleList.add(doodle);

tick(); }//method เมื่อเริ่มเกมให้ตัวdoodle & paddle ขยับ

});

timer.start();

addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_LEFT)

doodle.setVelocity(-BALL\_VEL, doodle.velocityY);

else if (e.getKeyCode() == KeyEvent.VK\_RIGHT)

doodle.setVelocity(BALL\_VEL, doodle.velocityY);

else if (e.getKeyCode() == KeyEvent.VK\_R)

reset();

}

@Override

public void keyReleased(KeyEvent e) {

doodle.setVelocity(0, doodle.velocityY);

}

});

}

/\*\* Set the state of the state of the game to its initial value and

prepare the game for keyboard input. \*/

public void reset() {

paddleList= new CopyOnWriteArrayList<Paddle>();

pUpList=new CopyOnWriteArrayList<PowerUp>();

enemyList = new CopyOnWriteArrayList<Enemy>();

jetList = new CopyOnWriteArrayList<Jet>();

doodle = new Doodle(200, 400, 0, -5);

int y=600;

for(int x=0;x<10;x++){

paddleList.add(new Paddle(COURTWIDTH+525 - rand.nextInt(1000), y));

y-=90;

}

requestFocusInWindow();

score = 0;

times = 0;

Paddle.WIDTH=80;

}

//วิธีการเหล่านี้ใช้เมื่อหยุดชั่วคราวสำหรับคำแนะนำ / ดำเนินการต่อ

public void stopTimer(){

timer.stop();

}

public void startTimer(){

timer.start();

}

public void restartTimer(){

timer.restart();

}

/\*\* Update the game one timestep by moving the ball and the paddle. \*/

void tick() {

//เพิ่ม powerups ทุกๆ scores หาร 8000 ลงตัว หรือเมื่อ score เท่ากับ 2000

if((score%5000==0 && score>0) || score == 2000 ){

pUpList.add(new PowerUp(rand.nextInt(550),0, 0, 2));

}

if(score%3000 == 0 && score>0){

enemyList.add(new Enemy(rand.nextInt(550),0,0,2));

}

if(score%1700 == 0 && score>0){

jetList.add(new Jet(rand.nextInt(550),0,0,2));

}

doodle.setBounds(getWidth(), getHeight());

doodle.move();

//Add intersects/move all paddles

for(Paddle p:paddleList){

doodle.bounce(p.intersects(doodle));

p.setBounds(getWidth(), getHeight());

p.move();

}

repaint(); // Repaint indirectly calls paintComponent.

//This ticker variable is used to determine if the ball is in powerup mode

if(ticker==false)

doodle.gravity+=0.3;

Game.scores.setText("Score " + score);

Game.times.setText("Times " + times/60);

}

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);// Paint background, border

g.drawImage(imgbg.getImage(),0,0,700,700,this);

g.drawImage(doodle.imgdd.getImage(), doodle.x, doodle.y,50,50,this);

for(Jet je : jetList){

g.drawImage(je.imgjet.getImage(), je.x,je.y,50,50, this);

je.y+=4;

if(doodle.intersects(je)==Intersection.UP ||doodle.intersects(je)==Intersection.DOWN|| doodle.intersects(je)==Intersection.RIGHT || doodle.intersects(je)==Intersection.LEFT){

ticker=true;

doodle.gravity=6;

jetList.remove(je);

}

}

if(doodle.y<200){

ticker=false;;

doodle.gravity=10;

}

//วาดตัว powerUps

for(PowerUp pU:pUpList){

g.drawImage(pU.imgrocket.getImage(), pU.x, pU.y,50,60, this);

pU.y+=4;

//ถ้ามัน intersects กับ ball, enter powerUp mode จะทำให้บอลเด้งขึ้นไปข้างบน.

if(doodle.intersects(pU)==Intersection.UP ||doodle.intersects(pU)==Intersection.DOWN

|| doodle.intersects(pU)==Intersection.RIGHT || doodle.intersects(pU)==Intersection.LEFT){

ticker=true;

doodle.gravity=4;

pUpList.remove(pU);

}

}

//ปิด powerup once the ball gets too high

if(doodle.y<10){

ticker=false;;

doodle.gravity=10;

}

//วาดสัตว์ประหลาด

for(Enemy em : enemyList){

//em.draw(g);

g.drawImage(em.imgen.getImage(), em.x, em.y,75,75,this);

em.y+=4;

if(doodle.intersects(em) == Intersection.UP || doodle.intersects(em) == Intersection.DOWN

|| doodle.intersects(em) == Intersection.RIGHT || doodle.intersects(em) == Intersection.LEFT){

enemyList.remove(em);

for(Paddle p: paddleList)

paddleList.remove(p);

g.drawImage(endbg.getImage(),0,0,700,700,this);

g.setFont(new Font("Verdana", Font.PLAIN, 30));

g.setColor(Color.white);

g.drawString("GAME OVER !!", COURTWIDTH/2-130, 40);

g.drawString("Your Final Score: " + score, COURTWIDTH/2-150, 70);

g.drawString("Times : " + times/60, COURTWIDTH/2-120, 100);

timer.stop();

}

}

//จบเกมเมื่อลูกบอลตกออกนอกเฟรม

if(doodle.y> COURTHEIGHT){

for(Paddle p: paddleList)

paddleList.remove(p);

g.drawImage(endbg.getImage(),0,0,700,700,this);

g.setFont(new Font("Verdana", Font.PLAIN, 30));

g.setColor(Color.white);

g.drawString("GAME OVER !!", COURTWIDTH/2-130, 40);

g.drawString("Your Final Score: " + score, COURTWIDTH/2-150, 70);

g.drawString("Times : " + times/60, COURTWIDTH/2-120, 100);

timer.stop();

}

if(doodle.postMove<doodle.holder ){

for(Paddle p: paddleList){

p.y+=8;

score+=1;//คะแนนจะเพิ่มขึ้นเมื่อลูกบอลสูงขึ้น ความไม่เท่าเทียมกันทำให้คะแนนเพิ่มขึ้นเท่านั้น

// เมื่อลูกบอลถึงจุดสูงสุดใหม่

if(score>4000){ p.setWidth(65); }

if(score>6000){ p.setWidth(55); }

if(score>10000){ p.setWidth(45); }

if(score>12000){ p.setWidth(37); }

if(p.y>COURTHEIGHT-10){

paddleList.remove(p);

paddleList.add(new Paddle(rand.nextInt(1000), -10));

if(rand.nextInt(8)==0){//ถ้าสุ่มแล้วได้8จะทำให้paddleเลื่อน

if(rand.nextInt(2)==0) { paddleList.get(paddleList.size()-1).setVelocity(2, 0);

else { paddleList.get(paddleList.size()-1).setVelocity(-2, 0); }

}

}

}

for(Paddle p: paddleList){ p.draw(g); }

}

@Override

public Dimension getPreferredSize() { return new Dimension(COURTWIDTH, COURTHEIGHT); }

}

Class Paddle

package amongusjumpproject;

import java.awt.\*;

public class Paddle extends GameObject {

final static int HEIGHT = 15;

static int WIDTH = 75;

public Paddle(int courtwidth, int courtheight) {

super((courtwidth - WIDTH) / 2, courtheight - HEIGHT - 20, 0, 0, WIDTH, HEIGHT);

}

//switch direction if it's a moving paddle

@Override

public void accelerate() {

if(x<0){

velocityX=2;

}

if(x> rightBound){

velocityX=-2;

}

}

@Override

public void draw(Graphics g) {

g.setColor(Color.lightGray);

g.fillRect(x, y, WIDTH, HEIGHT);

}

public void setWidth(int x){

this.WIDTH=x;

}

@Override

public void move(){

super.move();

}

@Override

public Rectangle getBounds() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

Class Doodle

package amongusjumpproject;

****import java.awt.Color;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

import javax.swing.ImageIcon;

public class Doodle extends GameObject {

int j;

final ImageIcon imgdd = new ImageIcon(this.getClass().getResource("among.png"));

final static int DIAMETER = 21;

public double gravity =0.8;

public static double holder;//score before

public static double postMove=0;//score after

public Doodle(int x, int y, int velocityX, int velocityY) {

super(x, y, 0, velocityY, DIAMETER, DIAMETER);

}

@Override

public void accelerate() {

if (x < 0)

x=rightBound;

else if (x > rightBound)

x=0;

}

// Bounce the ball, if necessary

public boolean bounce(Intersection i) {

switch (i) {

case NONE: break;//if i = NONE break

case UP: velocityY = -4; gravity=0.8; return true;

}

return false;

}

@Override

public void move(){

holder=this.y;

super.move();

this.y+=this.velocityY+gravity;

postMove=this.y;

}

@Override

public void draw(Graphics g){

//g.setColor(Color.GREEN);

//g.fillOval(x, y, DIAMETER, DIAMETER);

//g.setColor(Color.BLACK);

}

@Override

public Rectangle getBounds(){

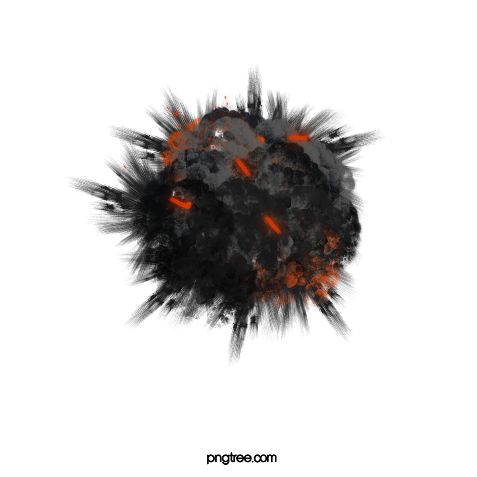
return new Rectangle(x,y,32,32);

}

}

}

Class Enemy

package amongusjumpproject;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

import javax.swing.ImageIcon;

public class Enemy extends GameObject{

final ImageIcon imgen = new ImageIcon(this.getClass().getResource("corona.png"));

final static int DIAMETER = 21;

public double gravity =0.8;

public static double holder;//score before

public static double postMove=0;//score after

public Enemy(int x, int y, int velocityX, int velocityY) {

super(x, y, 0, velocityY, DIAMETER, DIAMETER); }

@Override

public void move(){ super.move(); } @Override

public void accelerate() { }@Override

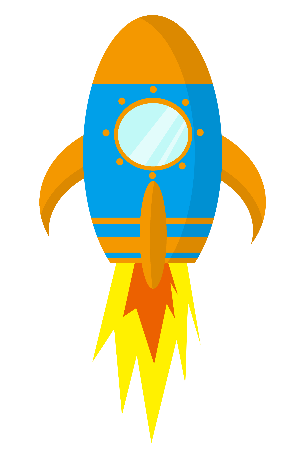
public void draw(Graphics g){ }@Override

public Rectangle getBounds(){ return new Rectangle(x,y,32,32);

}

}

Class PowerUp

****package amongusjumpproject;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

import javax.swing.ImageIcon;

public class PowerUp extends GameObject {

final ImageIcon imgrocket = new ImageIcon(this.getClass().getResource("rocket.png"));

final static int DIAMETER = 16;

//An object to represent a powerUp

public PowerUp(int x, int y, int velocityX, int velocityY) {

super(x, y, 0, velocityY, DIAMETER, DIAMETER); }

@Override

public void accelerate() { }

@Override

public void draw(Graphics g) { }

@Override

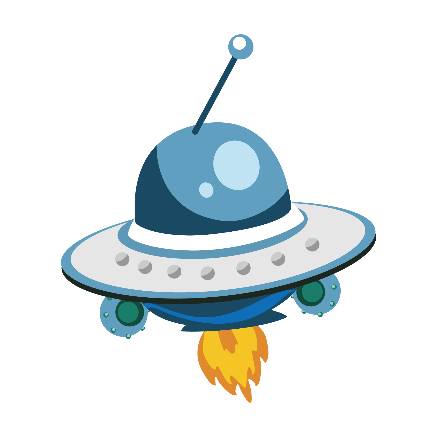
public void move(){ super.move(); }

@Override

public Rectangle getBounds() {

return new Rectangle(x,y,100,1000); }

Class Jet

package amongusjumpproject;

import java.awt.Graphics;

import java.awt.Rectangle;

import javax.swing.ImageIcon;

public class Jet extends GameObject{

final ImageIcon imgjet = new ImageIcon(this.getClass().getResource("jetspack.png"));

final static int DIAMETER = 16;

public Jet(int x, int y, int velocityX, int velocityY) {

super(x, y, 0, velocityY, DIAMETER, DIAMETER); }

@Override

public void accelerate() { }

@Override

public void draw(Graphics g) { }

@Override

public void move(){

super.move();

}

@Override

public Rectangle getBounds() { return new Rectangle(x,y,32,32); }

}

**บทที่ 3**

สรุป

**ปัญหาที่พบระหว่างการพัฒนา**

1. ไม่สามารถนำรูปลง Buttonได้ ผมจำได้ว่าอาจารย์เคยบอกผมแล้ว แต่ผมนึกไม่ออกแล้วไปหาในอินเทอร์เน็ตก็ยังไม่สามารถมาแปลงลงcodeตัวเองได้ครับ

2.การสุ่มแท่น เพื่อให้แท่นที่เอาไว้ให้กระโดดสุ่มเกิดบนหน้าจอ เพราะว่าไม่รู้ว่าจะทำยังไงให้มันสุ่มตำแหน่งการเกิดขึ้นมา จึงไปหาข้อมูลเพิ่มเติมในเน็ต

3.ผมยังไม่ค่อยเก่งเรื่องนี้มากเลยต้องไปหาในเน็ตเลยทำให้ค่อนข้างที่จะทำช้า

**จุดเด่นของโปรแกรม**

โปรแกรมที่เป็นรูปแบบเกมช่วยให้ผู้เล่นฝึกทักษะต่างๆ ฝึกสมาธิ ฝึกความว่องไวของสายตา ช่วยให้ผู้เล่นผ่อนคลายและสนุก นอกจากนี้ยังช่วยให้ผู้เขียนได้รับความรู้จากการลองผิดลองถูกในการทำโปรแกรมนี้ขึ้นมาเพื่อเป็นแนวทางในการพัฒนาตัวเองต่อไป

**คำแนะนำสำหรับผู้สอน**

อาจารย์เป็นอาจารย์ที่ตั้งใจในการสอนมากและพยายามให้เด็กเข้าใจผมเลยชอบที่จะเรียนกับอาจารย์แต่บางครั้งผมก็จะยังตามอาจารย์ไม่ค่อยทันเลยต้องถามเพื่อนบ้าง หรือถามอาจารย์ซ้ำอีกรอบบ้างผมเป็นคนแนวเข้าใจอะไรยากนิดหน่อยครับ