**Report on**

**Game Application : CSB RUN**

**Made by**

**Parunyu Anakitbumrung 6404062663223**

**Present to**

**Asst.Prof.Sathit Prasomphan**

**Object-Oriented Programming 040613204, Semester 1,**

**November 25, 2022**

**Game detail**

This game is about the student in computer science that run when he saw the police because he wants to steal the trophy cup in football club.

**How to play**

Use the “W”, “Spacebar” or “Up” to Jump

**Character**

This character is my friend his name is Tanaphat

this is main character to run over the object in game

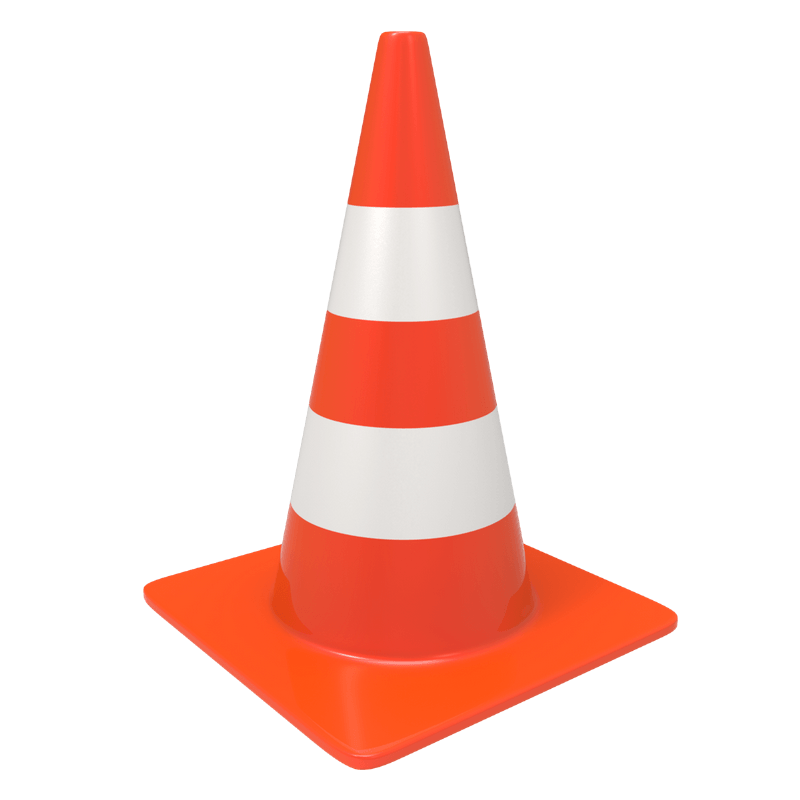
and he is Liverpool FC



This is a officer or police that want to

catch my friend Tanaphat

**Object**

****

**Logo

Description automatically generated**This is and object to jump over

This is Manchester United Balloon

that Tanaphat very hate it

**Place**

****

This is Menu page that have to select plat or exit

This is a road that main character will run

****

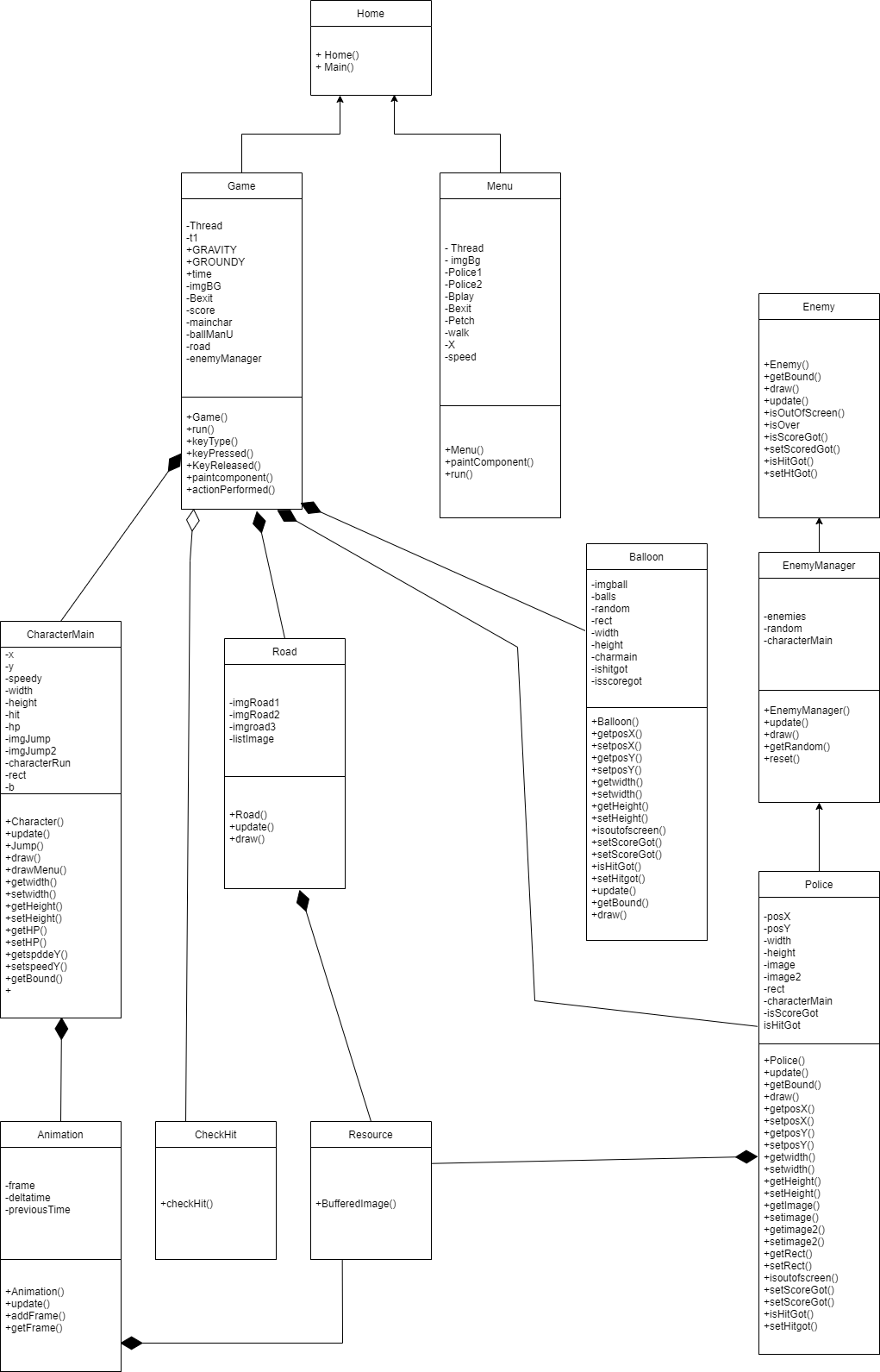
This is GameOver page that have to collect your scores and times

**Advantage**

1. **Just for Fun**
2. **Improve your skill**
3. **Learn how to escape the police**

**Table

Description automatically generatedRoad Map**

****

**Constructor**

Home() for calling Menu() and Game()

Menu() for create menu page

Game() for create game page and create

CharacterMain() , Road() , EnemyManager(), Balloon()

CharacterMain() add animation

Balloon() create random balloon

Road() create road and loop

Police() set images of police

Animation() create arraylist of character animation

Enemy() don’t have anything to do

EnemyManager() add enemy with random postion

**Encapsulation**

public class Balloon extends Enemy {

private class Ball{

        float posX;

        float posY;

    }

}

**Composition**

Text

Description automatically generated

**Polymorphism**

public class Balloon extends Enemy {

    public void draw(Graphics g){

        for(Ball ball: balls){

        g.drawImage(imgball,(int) ball.posX, (int)ball.posY,width,height ,null);

        }

    }

public CharacterMain() {

public void draw(Graphics g) {

        if ((int) y != 280) {

            if (getHit()) {

                g.drawImage(ImageJump2, (int) x, (int) y, (int) width, (int) height, null);

            } else {

                g.drawImage(ImageJump, (int) x, (int) y, (int) width, (int) height, null);

            }

        } else {

            g.drawImage(charecterRun.getFrame(), (int) x, (int) y, (int) width, (int) height, null);

            setHit(false);

        }

        g.setColor(Color.white);

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

        g.setColor(Color.red);

        if (HP > 0)

            g.fillRect(0, 0, HP \* 5, 20);

    }

}

Have a same paint method but another function

**Abstract**

public abstract class Enemy {

    public Enemy() {

    }

    public abstract Rectangle getBound();

    public abstract void draw(Graphics g);

    public abstract void update();

    public abstract boolean isOutOfScreen();

    public abstract boolean isOver();

    public abstract boolean isScoreGot();// 1 time

    public abstract void setScoreGot(boolean isScoreGot);

    public abstract boolean isHitGot();

    public abstract void setHitGot(boolean isHitGot);

}

This class create a template of methods

public class Police extends Enemy {

    @Override

    public boolean isOutOfScreen() {

        return (posX + width < 0);

    }

    @Override

    public boolean isOver() {

        return (characterMain.getX() > getPosX());

    }

    @Override

    public boolean isScoreGot() {

        return isScoreGot;

    }

    @Override

    public void setScoreGot(boolean isScoreGot) {

        this.isScoreGot = isScoreGot;

    }

    @Override

    public boolean isHitGot() {

        return isHitGot;

    }

    @Override

    public void setHitGot(boolean isHitGot) {

        this.isHitGot = isHitGot;

    }

@Override

    public void update(){

        posX-=2;

        rect.x = posX;

        rect.y = posY;

        rect.width = width;

        rect.height = height;

    }

    @Override

    public Rectangle getBound(){

        return rect;

    }

    @Override

    public void draw(Graphics g){

        // g.drawRect(posX, posY,width,height);

        g.drawImage(image, posX, posY,width,height, null);

    }

And when you extend abstract class you must override

**Inheritance**

class Home extends JFrame implements ActionListener {

public static void main(String[] args) {

        Home frame = new Home();

        frame.setSize(1000, 500);

        frame.setTitle("CSB RUN");

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

        frame.setLocationRelativeTo(null);

        frame.setVisible(true);

    }

}

Class Home extend Jframe

So you can use all methods in JFrame

For example :

frame is class Home but it can use the method setSize

setSize is one of JFrame method

GUI

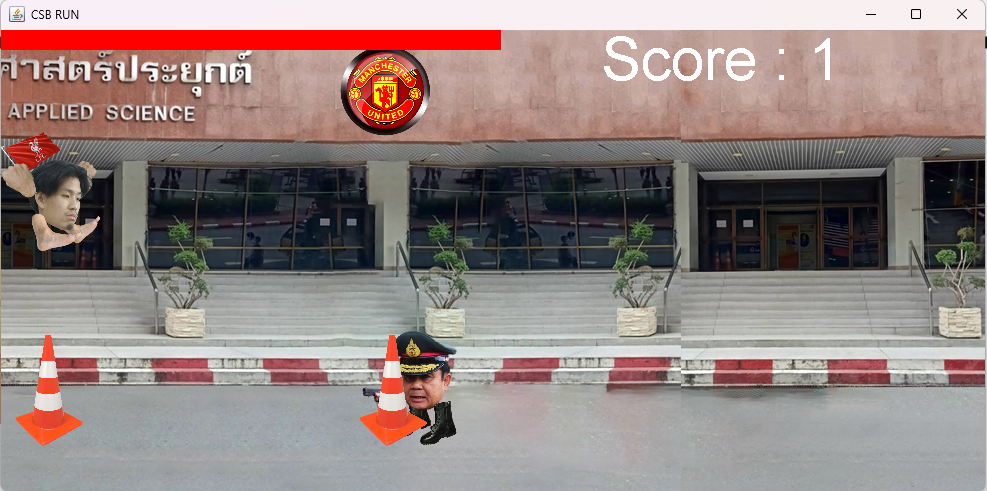
Graphical user interface, website

Description automatically generated

Component is

-Have a Button

-Have a image for background



Component is

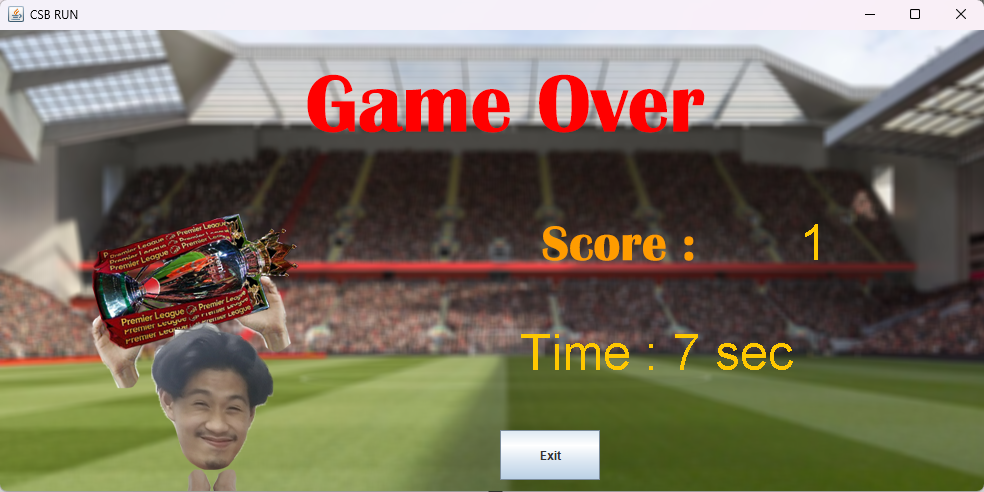
-Have character image

-Have a image for background

-Have a object Image that damage your character

-Have a Rectangle HP bar

-Have a String Score



Component is

-Have a Button

-Have a Score String and times

**Event Handling**

class Home extends JFrame implements ActionListener {

    Menu menu;

    Game game;

    @Override

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == menu.Bplay) {

            this.setLocationRelativeTo(null);

            this.remove(menu);

            this.setSize(1000, 500);

            game = new Game();

            this.add(game);

            game.addKeyListener(game);

            game.requestFocusInWindow();

        } else if (e.getSource() == menu.Bexit) {

            System.exit(0);

        }

        this.validate();

        this.repaint();

    }

}

actionPerformed

use to set the button command when you hit play and exit button

it will do the different command

**Algorithm**

if(enemies.get(0).isOver() && !e.isScoreGot()){

                    if(!characterMain.getHit())

                        Score++;

                    e.setScoreGot(true);

}

if(e.getBound().intersects(characterMain.getBound()) &&

!e.isHitGot() && enemies.get(0).isOver()){

                    characterMain.jump();

                    characterMain.setHP(characterMain.getHP()-20);

                    e.setHitGot(true);

                    characterMain.setHit(true);

}

This algorithm is used to count the score when we jump over the object

The problem is if we didn’t use this algorithm, it will count a lot score when we jump 1 object

First, we must check, did we jump over object already?

Second, we must check, did you got score already?

If we jump and not have a score it will it will count a score

But if you got a score, you will do nothing.

And Next is check hit already

The problem is if you hit 1 object you lose a lot of HP

The Algorithm is

First we check character and object intersects or not ?

Second we check, is it hit already ?

Third we check we pass the object or not ?

If it true all it will -20 HP

else it will do nothing

**The Problem**

* Have a lot of damage when hit 1 object
* Very hard to coding a beautiful structure if you create all of class in the same class it will hard to edit
* Have a little time to coding a game because it has a lot of exams
* Don’t know how to change a character

**Highlight of my game**

I use my friend to create a character, So I very happy when I saw his face