VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
INTERNATIONAL UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING

OBJECT-ORIENTED PROGRAMMING

Game Project

BASE ON PLANT AND ZOMBIES

Lecturer: Tran Thanh Tung

Member : Nguyen Thi Mai Huong (30%) ITITIU19128
Nguyen Quoc Bao (20%) ITITIU19081
Phan Le Dong (20%) ITITIU19102
Tran Khai Nguyen (30%) ITITIU19168

HO CHI MINH CITY, January 2021

Mnc lnc

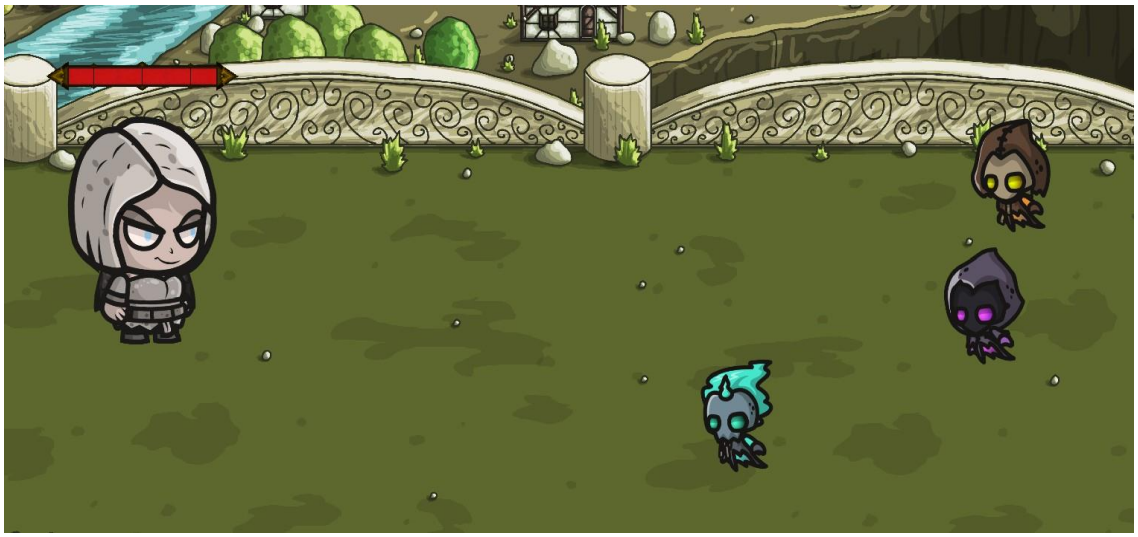
1 Introduction

1.1 Overview

- Idea of this game is came up with plant and zombie game.
- This project was designed based on Object – Oriented Programming method by Java language.

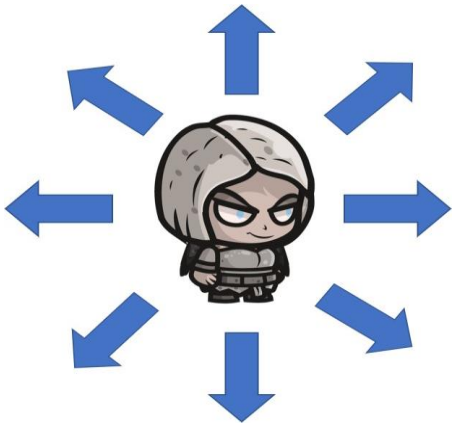
1.2 Goal

- You can commit around the map to avoid or attack several ghost with the aim purpose is protect your lives.



1.3 Rule

- In order to keep you alive, you have to control the main character move in one of the following directions : west, north-west, south, south-west, north, north-east, east or south-east.In order to attack with ghosts by sable.



- The main character has a red bar on the head to display the blood level.This bar divide to 10 equal part, if the collision between a ghost and the character happen, red bar will reduce one part. If ghost is killed by your character , you will receive 1 point.
- There are three type of ghosts: ice-ghost , soil-ghost and dark-ghost. They move from right to the left screen.



2 Class

2.1 AFrameOnImage class

```
public class AFrameOnImage {

    private boolean enablePaintRect = false;

    private int []DimsBounds = new int[4];

    public AFrameOnImage(int xOnImage, int yOnImage, int w, int h){...6 lines }
    public void VisibleRectDebug(boolean enable){...3 lines }
    public int[] GetBounds(){...3 lines }
    public void Paint(int x, int y, BufferedImage image, Graphics2D g2, int anchor, flo
    private void PaintBound(Graphics2D g){...3 lines }
}
```

2.

```
public class Animation {

    private long beginTime = 0;

    private long mesure = 20;

    private AFrameOnImage[] frames;
    private int NumOfFrame = 0;
    private int CurrentFrame = 0;

    public Animation(long mesure){...3 lines }
    public int get_numframe(){...3 lines }
    public void Update_Me(long deltaTime){...10 lines }
    public void AddFrame(AFrameOnImage sprite){...7 lines }
    public void PaintAnims(int x, int y, BufferedImage image, Graphics2D g2, int anchor
    public void PaintAnims(int i, int i0, BufferedImage[] Character, Graphics2D g2, int
}
```

2.3 GameScreen class

```
public abstract class GameScreen extends JFrame implements KeyListener{

    public static int KEY_PRESSED = 0;
    public static int KEY_RELEASED = 1;

    public int CUSTOM_WIDTH  = 500;
    public int CUSTOM_HEIGHT = 500;

    private GameThread G_Thread;

    public static int MASTER_WIDTH = 500, MASTER_HEIGHT = 500;

    public GameScreen(){...4 lines }
    public void RegisterImage(int id, BufferedImage image){...3 lines }
    public BufferedImage getImageWithID(int id){...3 lines }
    public GameScreen(int w, int h){...8 lines }
    private void InitScreen(){...8 lines }
    public void BeginGame(){...3 lines }
    private void InitThread(){...4 lines }
    @Override
    public void keyTyped(KeyEvent e) {}
    @Override
    public void keyPressed(KeyEvent e) {
        KEY_ACTION(e, GameScreen.KEY_PRESSED);
    }
    @Override
    public void keyReleased(KeyEvent e) {...3 lines }
    public abstract void GAME_UPDATE(long deltaTime);
    public abstract void GAME_PAINT(Graphics2D g2);
    public abstract void KEY_ACTION(KeyEvent e, int Event);

}
```

Create a game screen using JFrame in java.

2.4 Object class

```
public class Objects {

    private float posX, posY;
    private float w, h;
    private int balance_x;
    private int balance_y;

    public Objects() {...3 lines }
    public Objects(float x, float y, float w, float h) {...6 lines }
    public boolean isCollisionHappenWith(float x, float y) {...5 lines }
    public boolean isCollisionHappenWith(float x, float y, float w, float h) {...5 lines
    public boolean end_map()
    {...6 lines }
    public void setPos(float x, float y) {...4 lines }
    public void setPosX(float x) {...3 lines }
    public void setPosY(float y) {...3 lines }
    public float getPosX() {...3 lines }
    public float getPosY() {...3 lines }
    public float getW() {...3 lines }
    public float getH() {...3 lines }
    public void increasePosX(float m) {...3 lines }
    public void increasePosY(float m) {...3 lines }
    public void decreasePosX(float m) {...3 lines }
    public void decreasePosY(float m) {...3 lines }
}
```

2.5 GameThread class

```
public class GameThread extends JPanel implements Runnable{

    private GameScreen context;
    private Thread thread;
    private Graphics ThisGraphics;
    public static int FPS = 70;
    private BufferedImage buffImage;
    private int MasterWidth, MasterHeight;
    public static float scaleX_ = 1, scaleY_ = 1;
    public GameThread(GameScreen context){...9 lines }
    public void StartThread(){...3 lines }
    public void paint(Graphics g){...9 lines }
    private void UpdateSize(){...9 lines }
    @Override
    public void run() {...41 lines }
}
```

Create game looper.

2.6 HP class

```
class HP extends Objects {
    private BufferedImage HP_bar;
    private BufferedImage HP_in[];
    private int temp;

    public HP(int x, int y){...19 lines }
    public void update_among(long deltaTime){...3 lines }
    public void paint_among(Graphics2D g2 , int x, int y , int hp){...5 lines }
}
```

Create the blood bar of character.

2.7 SeflGame class

```
public class SeflGame extends GameScreen{

    private static int WIDTH = 3800;
    private static int HEIGHT = 2100;
    Random generator = new Random(19900828);

    private backGround BG;
    private character cha;
    private enemy ene[];
    private enemy2 ene2[];
    private enemy3 ene3[];
    private HP hp;
    private boolean is_lose;
    private int point;

    public SeflGame() {...31 lines }
    public void start()
    {...3 lines }
    @Override
    public void GAME_UPDATE(long deltaTime) {...89 lines }
    @Override
    public void GAME_PAINT(Graphics2D g2) {...17 lines }

    @Override
    public void KEY_ACTION(KeyEvent e, int Event) {
        cha.key_among(e, Event);
    }

}
```

Create main interface to play game.

2.8 BackGround class

```
class backGround extends Objects{

    private BufferedImage background;

    public backGround(){...7 lines }
    public int get_width(){...3 lines }
    public int get_heigh(){...3 lines }
    public void update_among(long deltaTime){...3 lines }
    public void paint_among(Graphics2D g2){...3 lines }
}
```

Createa background of game.

2.9 Button class

```
class button extends Objects implements MouseListener{
    private BufferedImage but;
    private int scale;
    public button(int x, int y, int w, int h){...11 lines }
    public void paint_button(Graphics2D g2){...3 lines }

    @Override
    public void mouseClicked(MouseEvent e) {...4 lines }

    @Override
    public void mousePressed(MouseEvent e) {...3 lines }

    @Override
    public void mouseReleased(MouseEvent e) {...3 lines }

    @Override
    public void mouseEntered(MouseEvent e) {...3 lines }

    @Override
    public void mouseExited(MouseEvent e) {...3 lines }
}
```

Create button in game.

2.10 Enemy class

```
class enemy extends Objects{

    public static int STAGE = 0;
    private static final int ALIVE = 0;
    private static final int DEAD = 1;

    private BufferedImage left;
    private final Animation left_anim;
    private BufferedImage die;
    private final Animation die_anim;
    private static final float SPEED = (float) 10.0;
    private int time_dead;
    private int frame ;
    private boolean dead;

    enemy(int x, int y){...26 lines }
    public void is_dead()
    {...3 lines }
    public int getx(){...3 lines }
    public int gety(){...3 lines }
    public void update_enemy(long deltaTime){...5 lines }
    public void paint_enemy(Graphics2D g2){...12 lines }

}
```

Create a enemy in game.

2.11 Character class

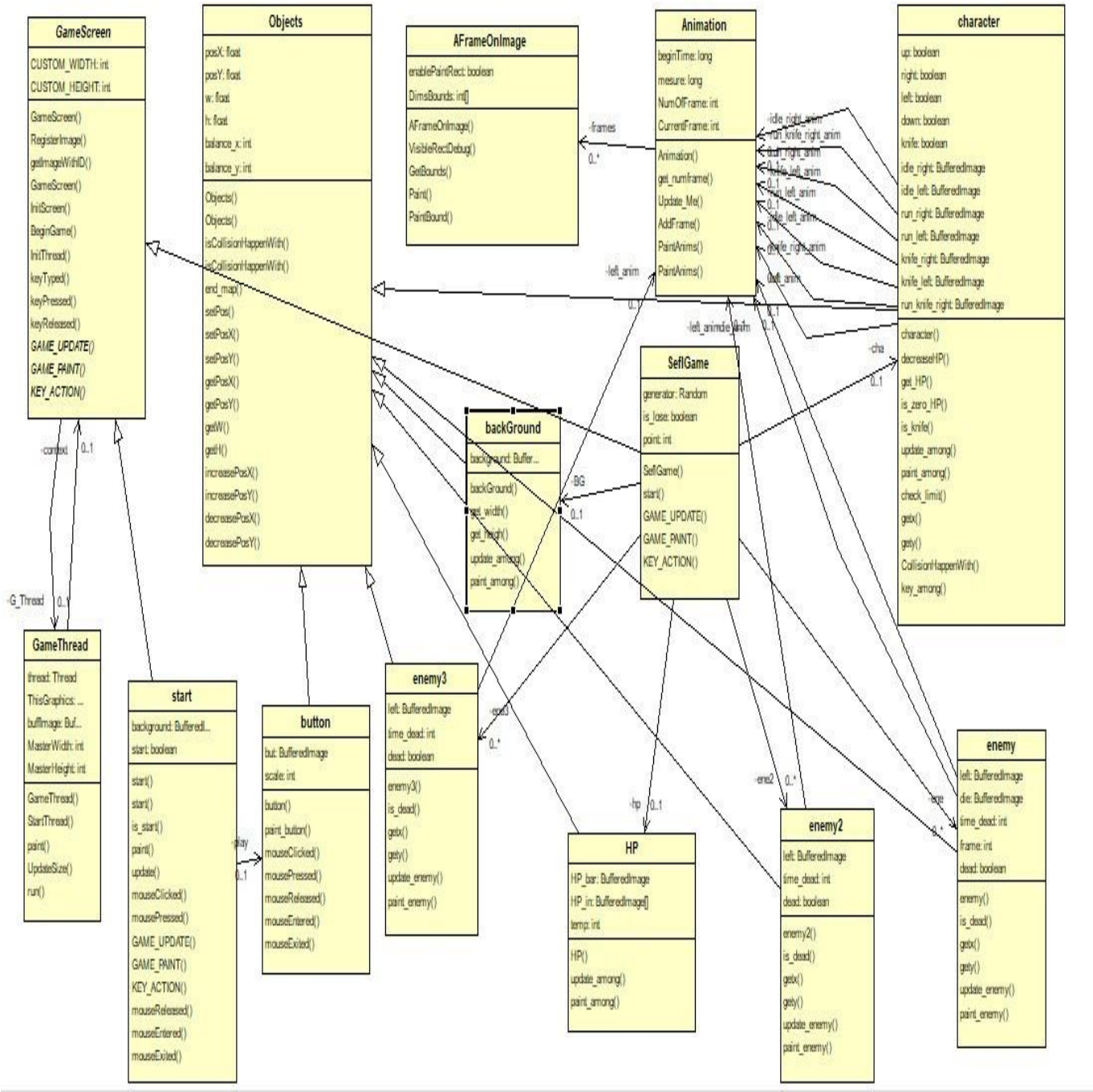
```
public character(int x, int y){...54 lines }
public void decreaseHP(int a){...3 lines }
public int get_HP()
{...3 lines }
public boolean is_zero_HP(){...5 lines }
public boolean is_knife(){...3 lines }
public void update_among(long deltaTime){...9 lines }
public void paint_among(Graphics2D g2){...52 lines }
public boolean check_limit(){...51 lines }
public int getx(){...3 lines }
public int gety(){...3 lines }
public boolean CollisionHappenWith(float x, float y, float w, float h){...5 lines }

public void key_among(KeyEvent e, int Event){...185 lines }
```

2.12 Start class

```
class start extends GameScreen implements MouseListener{

    private BufferedImage background;
    private button play;
    private boolean start;
    public start(int x, int y){...12 lines }
    public static void main(String[] args) {...4 lines }
    public void start()
    {...3 lines }
    public boolean is_start(){...3 lines }
    public void paint(Graphics2D g2) {...4 lines }
    public void update(long deltaTime){...3 lines }
    @Override
    public void mouseClicked(MouseEvent e) {...10 lines }
    @Override
    public void mousePressed(MouseEvent e) {...4 lines }
    @Override
    public void GAME_UPDATE(long deltaTime) {...3 lines }
    @Override
    public void GAME_PAINT(Graphics2D g2) {...3 lines }

    @Override
    public void KEY_ACTION(KeyEvent e, int Event) {

    }
```

Start game interface.

3 Class diagram



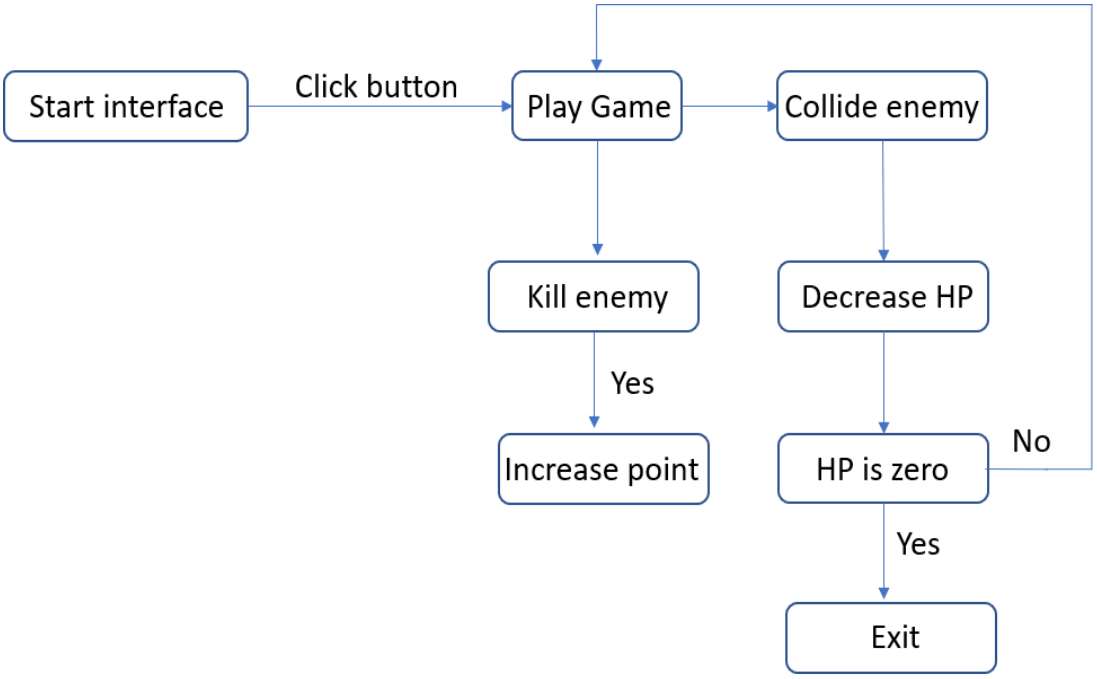
4 Animation

- We have a list of single image with the same size.

- Using CSS Sprites Generator tool (https://www.toptal.com/developers/css/sprite-generator/) to cre- ate sprite.

- Save each frame of image to array.
- Draw frame by frame after fixed times and you will see the change of character.

5 Game stage diagram



Explain the code:

How to load image ? and character movement

As for the movement of the character and monster it is a series of images of the same size and every time it loads the next image on the screen.

```
public character(int x, int y){
    super(x, y , 430, 570);

    try {
        idle_right = ImageIO.read(new File("Assets/idle.png"));
        idle_left = ImageIO.read(new File("Assets/idle_left.png"));
        run_right = ImageIO.read(new File("Assets/run_right.png"));
        run_left = ImageIO.read(new File("Assets/run_left.png"));
        knife_right = ImageIO.read(new File("Assets/knife_right.png"));
        knife_left = ImageIO.read(new File("Assets/knife_left.png"));
        run_knife_right = ImageIO.read(new File("Assets/run_knife_right.png"));
    } catch (IOException ex) {}
```

That is the movement of the image, and the movement of the character, we create a frame to place the image and use the coordinates to move the frame as you like

```
public boolean check_limit(){
    if(this.getx()>=-240 && this.getx() <= 3800-240 && this.gety() >= 250 && this.gety() <= 1210)
    {
        System.out.print("X = ");
        System.out.print(this.getx());
        System.out.print("Y = ");
        System.out.print(this.gety());
        System.out.print("\n");
       // System.out.println("END LEFT");
        return true;
    }
    else
    {
        System.out.print("NewX = ");
        System.out.println(this.getPosX());
        System.out.print("NewY = ");
        System.out.println(this.getPosY());
        if(this.getx() <= -240 && this.gety() <= 250)
        {
            this.setPosX(- 480);
            this.setPosY(60);
        }
        else if(this.getx() <= -240 && this.gety() >= 1210)
        {
            this.setPosX(- 480);
            this.setPosY(1020);
        }
        else if(this.getx() >= 3800 -240 && this.gety() <= 250)
        {
            this.setPosX(3320);
            this.setPosY(60);
        }
        else if(this.getx() >= 3800 -240 && this.gety() >= 1210)
        {
            this.setPosX(3320);
            this.setPosY(1020);
        }
        else if(this.getx() >= 3800 -240)
```

```
        }
        else if(this.getx() >= 3800 -240 && this.gety() >= 1210)
        {
            this.setPosX(3320);
            this.setPosY(1020);
        }
        else if(this.getx() >= 3800 -240)
            this.setPosX(3320);
        else if(this.gety() <= 250)
            this.setPosY(60);
        else if(this.gety() >= 1210)
            this.setPosY(1020);
        else if(this.getx() <= -240)
        {
            this.setPosX(- 480);
        }

    }
    return false;
}
public int getx(){
    return (int)this.getPosX() + 240;
}
public int gety(){
    return (int)this.getPosY() + 190;
}
public boolean CollisionHappenWith(float x, float y, float w, float h){
    if( x < this.getPosX() + this.getW() && x + w > this.getPosX() && y < this.getPosY() + 120 + this.getH() && h + y > this.getPosY() + 120)
        return true;
    return false;
}
```

How when the character touches the monster will lose blood ?

When the monster touches the character or the monster is guillotined by the character, we create frames containing pictures of both of them if the two frames touch the monster will die

```
public void GAME_UPDATE(long deltaTime) {
    if(cha.is_zero_HP())
    {
         System.exit(0);
    }
    System.out.println(this.point);
    cha.update_among(deltaTime);
    for(int i = 0;i<3;i++)
    {
        int value_y = generator.nextInt(1150) + 60;
        int value_x = generator.nextInt(1000);
        ene[i].update_enemy(deltaTime);
        if(cha.is_knife())
        {
            if(cha.CollisionHappenWith((int)ene[i].getx() - 200, (int)ene[i].gety(), (int)ene[i].getW(), (int)ene[i].getH()))
            {
                ene[i].is_dead();
                //System.out.println("enemy dead");
                ene[i] = new enemy(3800 + value_x, value_y);
                this.point++;
            }
        }
```

And when the character is touched by a monster, he will lose his blood as well

```
if(cha.CollisionHappenWith((int)ene[i].getx(), (int)ene[i].gety(), (int)ene[i].getW(), (int)ene[i].getH()))
{
    cha.decreaseHP(1);
    //System.out.print(cha.get_HP());
    ene[i] = new enemy(3800 + value_x, value_y);
}
if(ene[i].end_map() == true)
{
    //System.out.println("enemy leave");
    ene[i] = new enemy(3800 + value_x, value_y);
}
```

When hp = 0 means the game will be out and the health bar is we also use a series of images to make the blood loss movement of the core.

```
public void decreaseHP(int a){
    this.HP -= a;
}
public int get_HP()
{
    return this.HP;
}
public boolean is_zero_HP(){
    if(this.HP == 0)
        return true;
    else return false;
}
public boolean is_knife(){
    return knife;
```

```
    public HP(int x, int y){
        super(x,y,512,128);
        HP_in = new BufferedImage[11];
        try {
            HP_bar = ImageIO.read(new File("Assets/HP_bar.png"));

            HP_in[0] = ImageIO.read(new File("Assets/HP0.png"));
            HP_in[1] = ImageIO.read(new File("Assets/HP1.png"));
            HP_in[2] = ImageIO.read(new File("Assets/HP2.png"));
            HP_in[3] = ImageIO.read(new File("Assets/HP3.png"));
            HP_in[4] = ImageIO.read(new File("Assets/HP4.png"));
            HP_in[5] = ImageIO.read(new File("Assets/HP5.png"));
            HP_in[6] = ImageIO.read(new File("Assets/HP6.png"));
            HP_in[7] = ImageIO.read(new File("Assets/HP7.png"));
            HP_in[8] = ImageIO.read(new File("Assets/HP8.png"));
            HP_in[9] = ImageIO.read(new File("Assets/HP9.png"));
            HP_in[10] = ImageIO.read(new File("Assets/HP10.png"));
        } catch (IOException ex) {}
    }
    public void update_among(long deltaTime){

    }
    public void paint_among(Graphics2D g2 , int x, int y , int hp){
       // System.out.println(hp);
        g2.drawImage(HP_in[hp], x , y, this.HP_in[hp].getWidth(), this.HP_in[hp].getHeight()-50, null);
        g2.drawImage(HP_bar, x, y, this.HP_bar.getWidth(), this.HP_bar.getHeight()-50, null);
    }
}
```

6 Conclusion

6.1 Result

- The start interface.

- The game interface.

6.2 Limited

- Beside the success of build the game with basic rules, our project still has many case that can not be solved:
- Do not have Save Game, Save Score, input the Name for the user and display the score on user interface just only show on console.log
- This game just for 1 player, still have not modified for 2 players.
- There is only 1 level. We have not yet build more upgrade levels.
- Our game has not yet complete, lack of sound and menu to guide user to control main character.