Flying Hamster

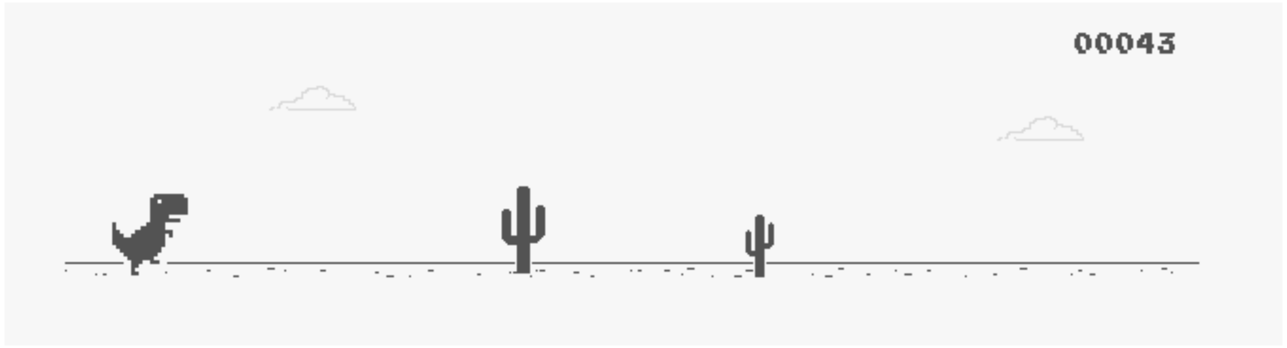
ICT 융합학부 이수진

Contents

1

동기

동기



소개

햄스터

hamster

바위

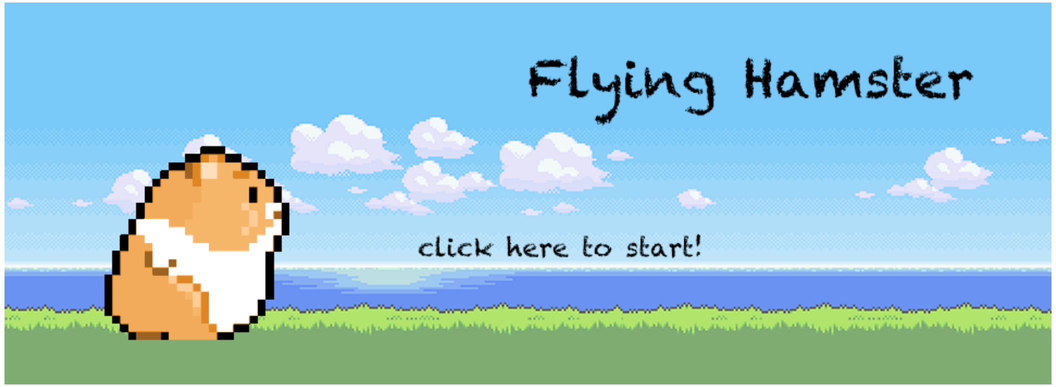
rock

귀신

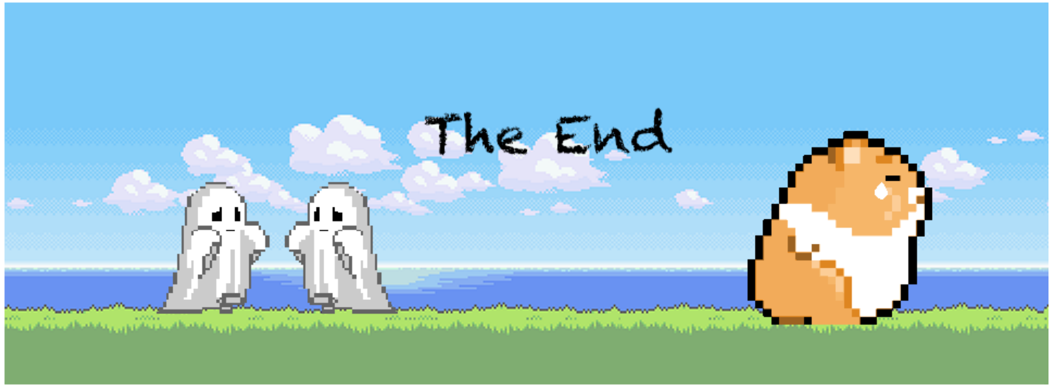
ghost

소개

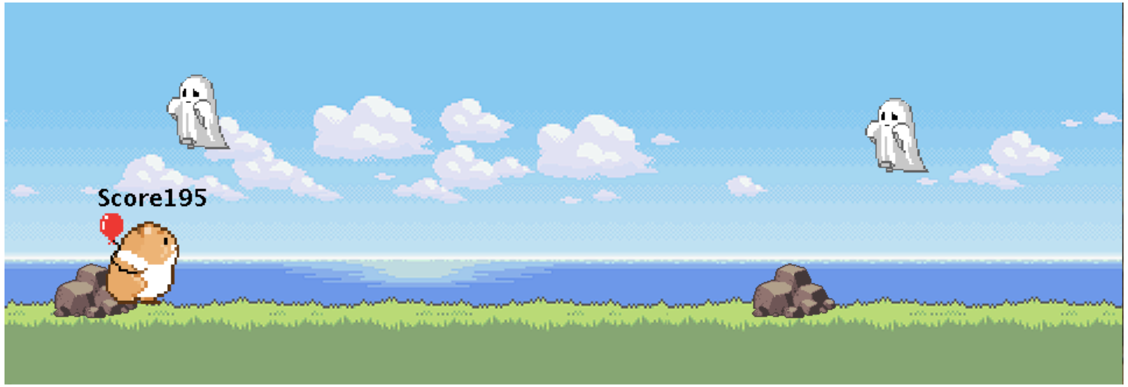
시작 화면

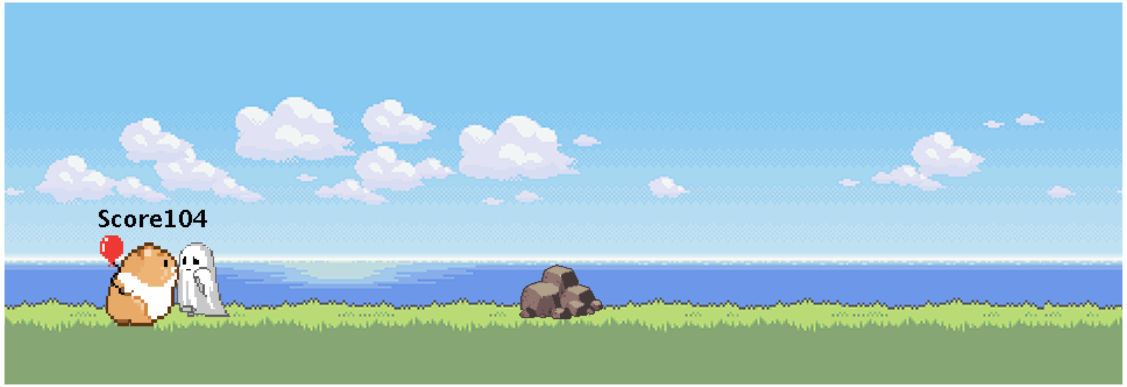


종료 화면



실행 화면

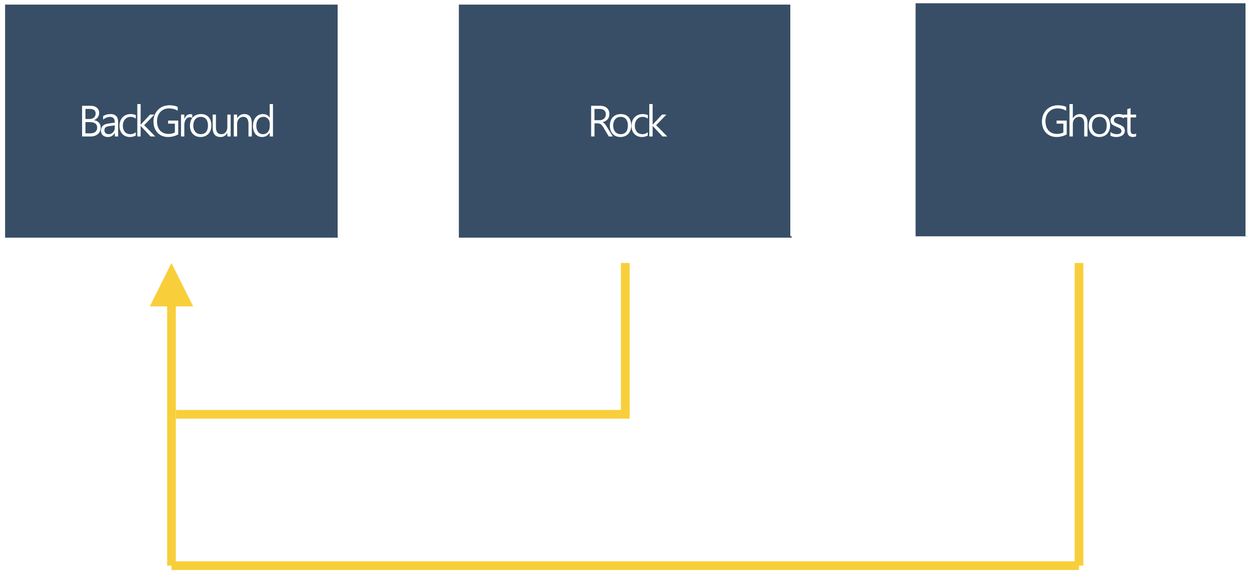




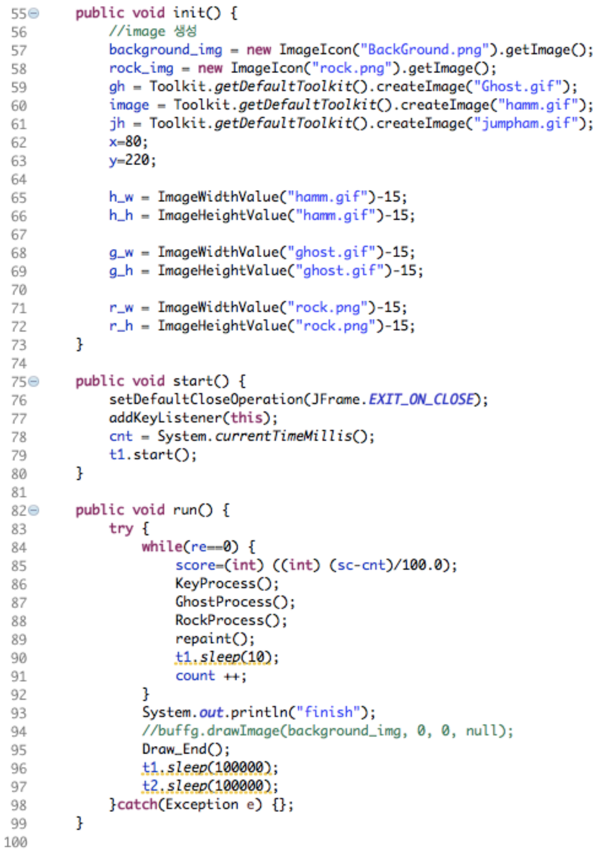
2

코드 설명

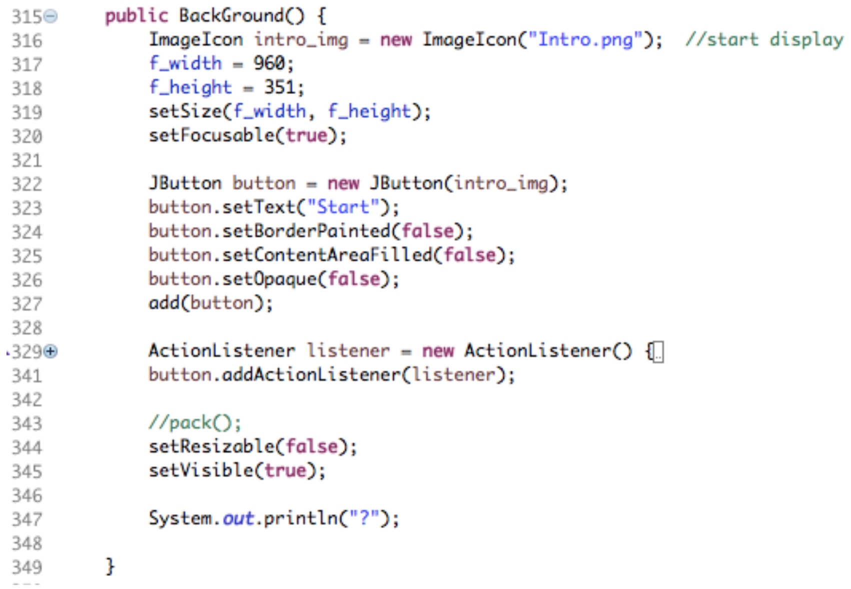
Class



Code

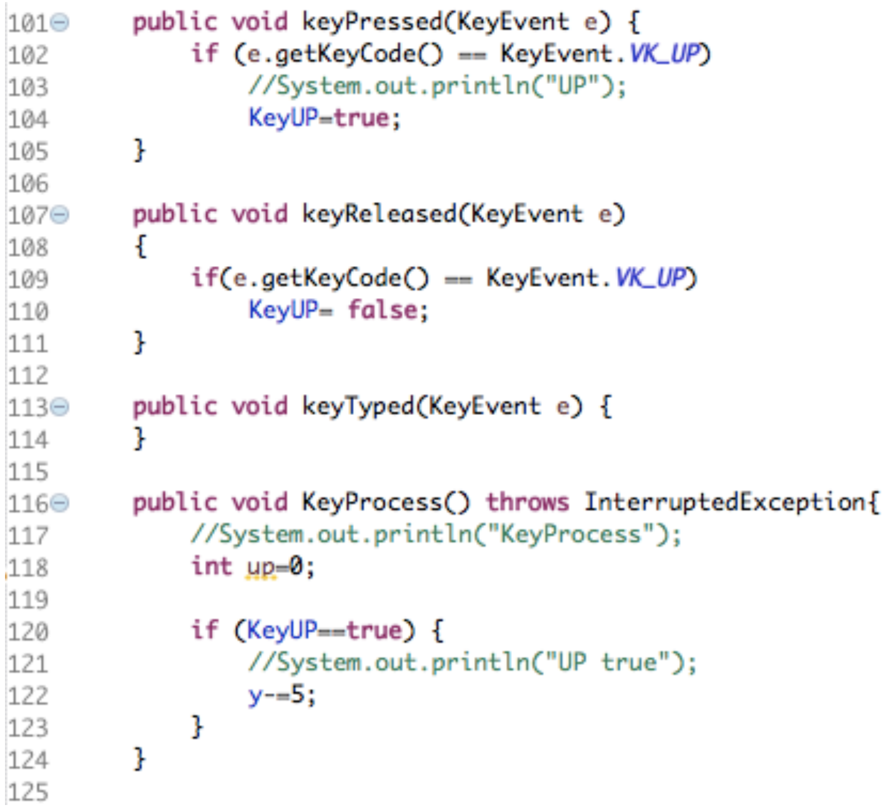


```java
public void init() {
    //image 생성
    background_img = new ImageIcon("BackGround.png").getImage();
    rock_img = new ImageIcon("rock.png").getImage();
    gh = Toolkit.getDefaultToolkit().createImage("Ghost.gif");
    image = Toolkit.getDefaultToolkit().createImage("hamm.gif");
    jh = Toolkit.getDefaultToolkit().createImage("jumpham.gif");
    x=80;
    y=220;

    h_w = ImageWidthValue("hamm.gif")-15;
    h_h = ImageHeightValue("hamm.gif")-15;

    g_w = ImageWidthValue("ghost.gif")-15;
    g_h = ImageHeightValue("ghost.gif")-15;

    r_w = ImageWidthValue("rock.png")-15;
    r_h = ImageHeightValue("rock.png")-15;
}

public void start() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    addKeyListener(this);
    cnt = System.currentTimeMillis();
    t1.start();
}

public void run() {
    try {
        while(re==0) {
            score=(int) ((int) (sc-cnt)/100.0);
            KeyProcess();
            GhostProcess();
            RockProcess();
            repaint();
            t1.sleep(10);
            count ++;
        }
        System.out.println("finish");
        //buffg.drawImage(background_img, 0, 0, null);
        Draw_End();
        t1.sleep(100000);
        t2.sleep(100000);
    }catch(Exception e) {};
}
```



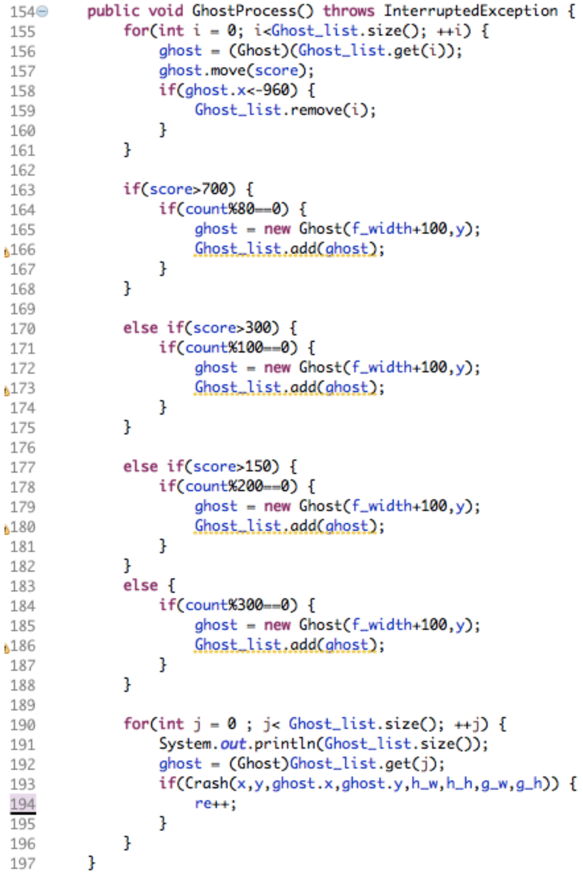
```java
public BackGround() {
    ImageIcon intro_img = new ImageIcon("Intro.png");  //start display
    f_width = 960;
    f_height = 351;
    setSize(f_width, f_height);
    setFocusable(true);

    JButton button = new JButton(intro_img);
    button.setText("Start");
    button.setBorderPainted(false);
    button.setContentAreaFilled(false);
    button.setOpaque(false);
    add(button);

    ActionListener listener = new ActionListener() {
    button.addActionListener(listener);

    //pack();
    setResizable(false);
    setVisible(true);

    System.out.println("?");

}
```

Code

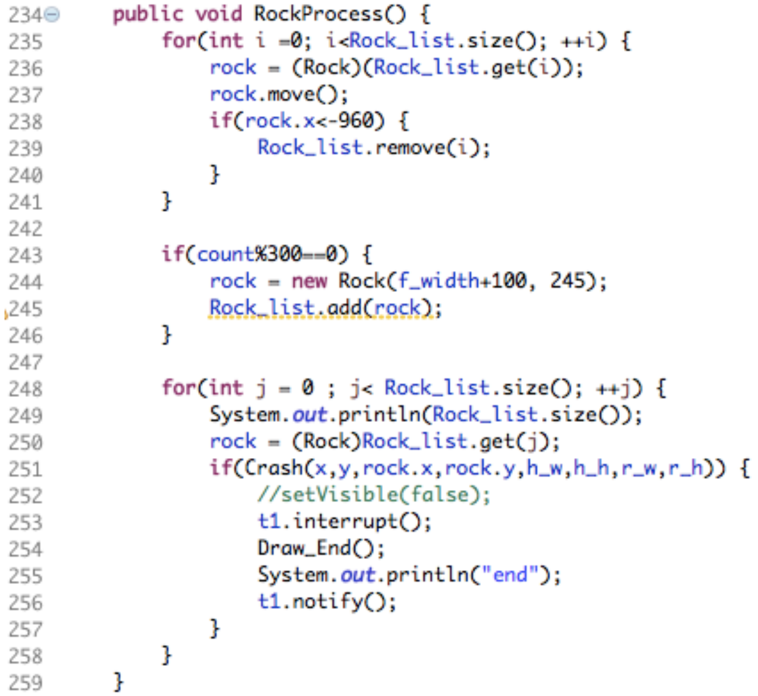


```java
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_UP)
            //System.out.println("UP");
            KeyUP=true;
    }

    public void keyReleased(KeyEvent e)
    {
        if(e.getKeyCode() == KeyEvent.VK_UP)
            KeyUP= false;
    }

    public void keyTyped(KeyEvent e) {
    }

    public void KeyProcess() throws InterruptedException{
        //System.out.println("KeyProcess");
        int up=0;

        if (KeyUP==true) {
            //System.out.println("UP true");
            y-=5;
        }
    }

```

Code

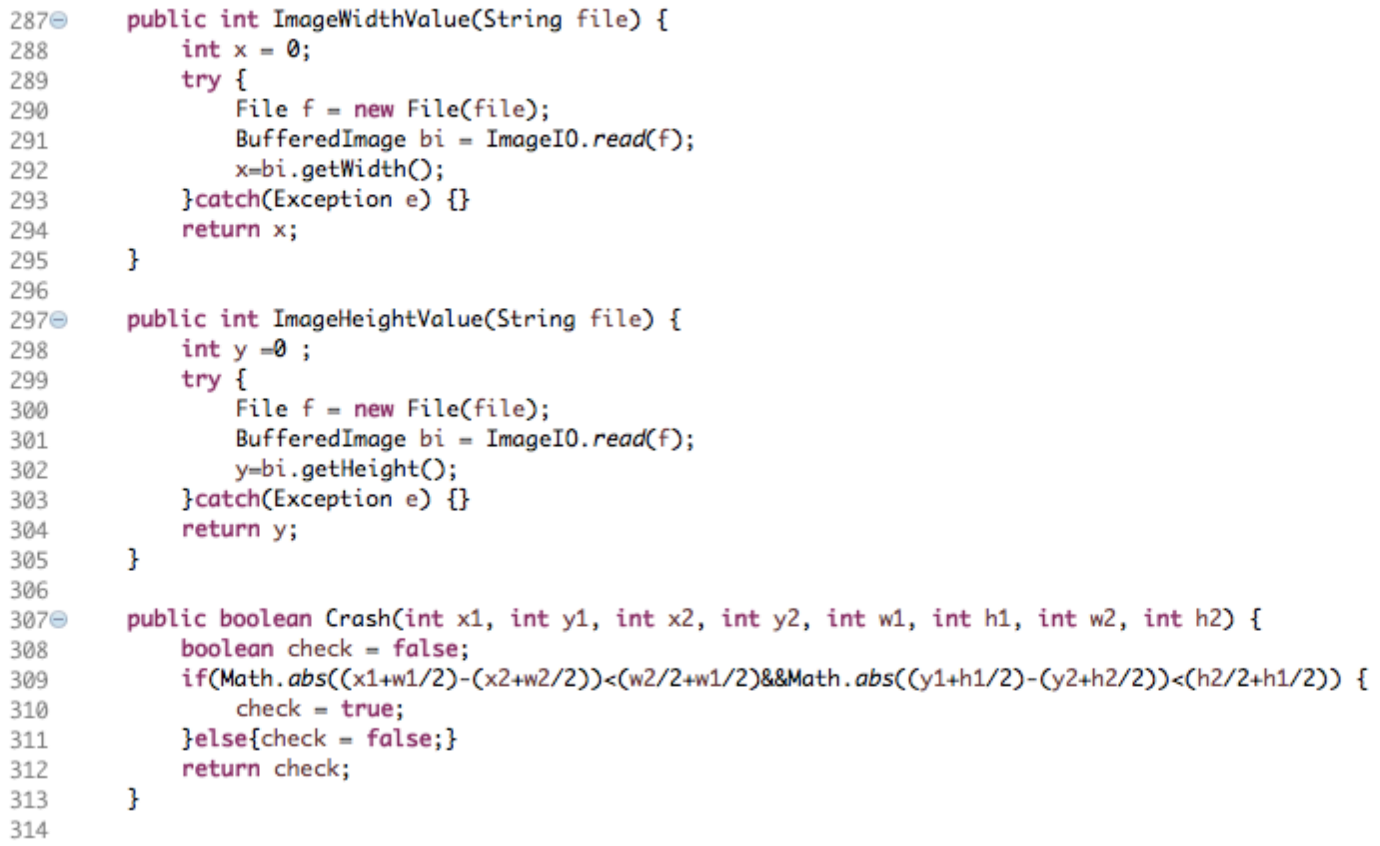


```java
public void GhostProcess() throws InterruptedException {
    for(int i = 0; i<Ghost_list.size(); ++i) {
        ghost = (Ghost)(Ghost_list.get(i));
        ghost.move(score);
        if(ghost.x<-960) {
            Ghost_list.remove(i);
        }
    }

    if(score>700) {
        if(count%80==0) {
            ghost = new Ghost(f_width+100,y);
            Ghost_list.add(ghost);
        }
    }

    else if(score>300) {
        if(count%100==0) {
            ghost = new Ghost(f_width+100,y);
            Ghost_list.add(ghost);
        }
    }

    else if(score>150) {
        if(count%200==0) {
            ghost = new Ghost(f_width+100,y);
            Ghost_list.add(ghost);
        }
    }
    else {
        if(count%300==0) {
            ghost = new Ghost(f_width+100,y);
            Ghost_list.add(ghost);
        }
    }

    for(int j = 0 ; j< Ghost_list.size(); ++j) {
        System.out.println(Ghost_list.size());
        ghost = (Ghost)Ghost_list.get(j);
        if(Crash(x,y,ghost.x,ghost.y,h_w,h_h,g_w,g_h)) {
            re++;
        }
    }
}
```



```java
public void RockProcess() {
    for(int i =0; i<Rock_list.size(); ++i) {
        rock = (Rock)(Rock_list.get(i));
        rock.move();
        if(rock.x<-960) {
            Rock_list.remove(i);
        }
    }

    if(count%300==0) {
        rock = new Rock(f_width+100, 245);
        Rock_list.add(rock);
    }

    for(int j = 0 ; j< Rock_list.size(); ++j) {
        System.out.println(Rock_list.size());
        rock = (Rock)Rock_list.get(j);
        if(Crash(x,y,rock.x,rock.y,h_w,h_h,r_w,r_h)) {
            //setVisible(false);
            t1.interrupt();
            Draw_End();
            System.out.println("end");
            t1.notify();
        }
    }
}
```

Code



```
public int ImageWidthValue(String file) {
    int x = 0;
    try {
        File f = new File(file);
        BufferedImage bi = ImageIO.read(f);
        x=bi.getWidth();
    }catch(Exception e) {}
    return x;
}

public int ImageHeightValue(String file) {
    int y =0 ;
    try {
        File f = new File(file);
        BufferedImage bi = ImageIO.read(f);
        y=bi.getHeight();
    }catch(Exception e) {}
    return y;
}

public boolean Crash(int x1, int y1, int x2, int y2, int w1, int h1, int w2, int h2) {
    boolean check = false;
    if(Math.abs((x1+w1/2)-(x2+w2/2))<(w2/2+w1/2)&&Math.abs((y1+h1/2)-(y2+h2/2))<(h2/2+h1/2)) {
        check = true;
    }else{check = false;}
    return check;
}

```

3

시연

4

보완할 점

보완해야 할 점

Ending 장면이 안들어감

유령과 햄스터의 충돌 여부를 사진 크기로 파악함으로써 발생하는 오차

감사합니다