

MATA KULIAH PROJECT MEMBUAT GAME DI GREENFOOT

Praktikum Pemrograman Berorientasi Objek (PBO)

Dosen Pengampu : SAYEKTI HARITS SURYAWAN, S.Kom., M.Kom.



INF2153

UJIAN AKHIR SEMESTER (UAS)

Oleh Kelompok 11

- | | |
|--|------------------------|
| 1. Dyah Anggraeni Kusuma Dewi | : 2211102441133 |
| 2. Muhammad Albert Putra Firman | : 2211102441077 |
| 3. Nur Ade Afriyani | : 2211102441080 |

Teknik Informatika

Fakultas Sains & Teknologi

Universitas Muhammadiyah Kalimantan Timur

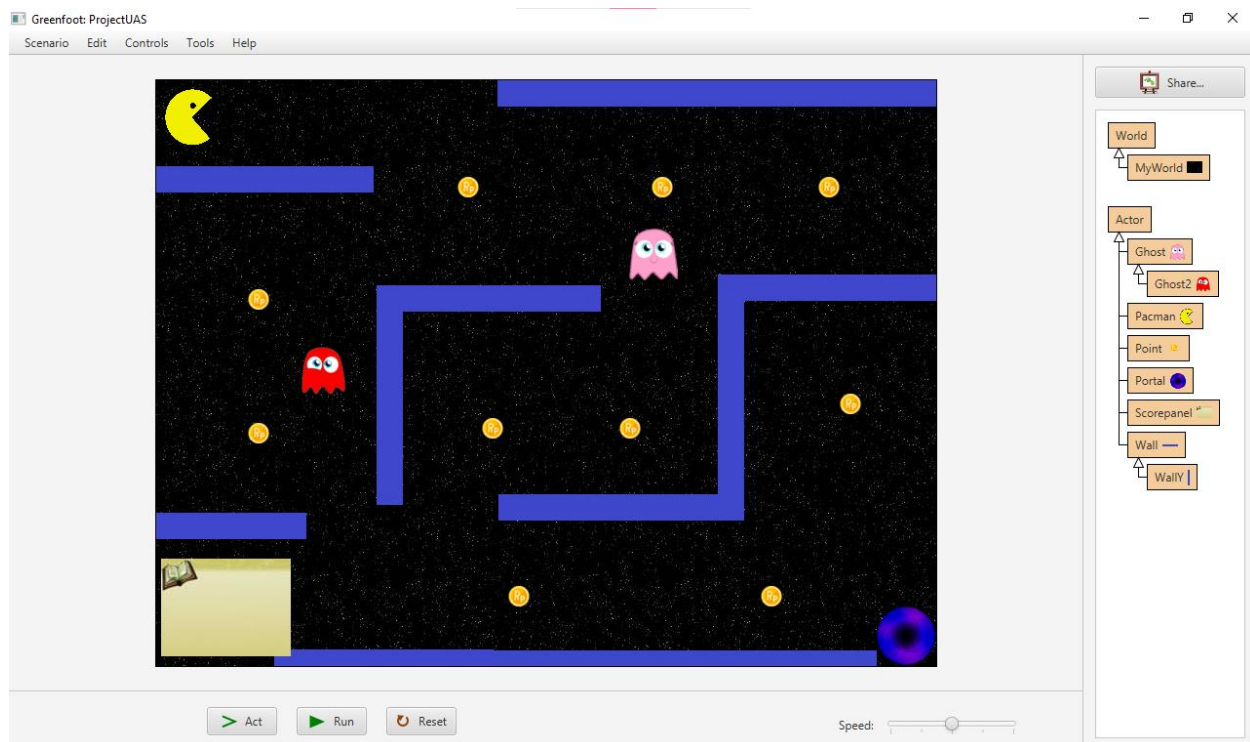
Samarinda 2023

Pada **Project UAS Praktikum Pemrograman Berorientasi Objek (PBO)** ini kami memutuskan untuk membuat Sebuah Game yang sederhana yaitu Game “**Pac Man**” yang dimana kami menggunakan Aplikasi yang biasa kami gunakan yaitu Software **GREENFOOT** yang telah diajarkan pada Semester 3 ini.

Berikut adalah Skenario yang telah kami terapkan pada pembuatan Aplikasi Game “**Pac Man**”.

1. Objek bernama **Pac Man** ini dapat bergerak ke atas, bawah, kanan, dan kiri. Setiap **Pac Man** bergerak, maka **Pac Man** akan membuka dan menutup mulutnya. **Pac Man** juga bisa menghadap ke arah di saat dia berjalan.
2. Terdapat point yang berbentuk bulat berwarna kuning yang ketika akan dimakan oleh **Pac Man** maka Score akan bertambah 10.
3. Terdapat juga penghalang dinding yang tidak dapat ditembus oleh **Pac Man**.
4. **Pac Man** di dalam Game tersebut dapat melewati Ghost yang bergerak menghalangi jalan. Ketika **Pac Man** menabrak Ghost tersebut makan nyawa **Pac Man** akan berkurang dan akan kembali ke posisi awalnya.
5. Ketika **Pac Man** berhasil sampai ke Portal, maka Gmae akan berlanjut/naik ke level selanjutnya, yang dimana Ghostnya akan bertambah 1.

Dibawah ini Template awal Game kami.



PENJELASAN CODINGAN PADA GAME.

Untuk Codingan Game ini kami menerapkan banyak Code. Berikut ini adalah penjelasan Codingan pada setiap Subclass Actor dan World di Game Pac Man yang telah kami buat.

1. MyWorld.

Subclass myWorld pada world berperan sebagai latar pada Game. Kami memilih ukuran background latar angkasa.

```
public MyWorld()
{
    // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
    super(800, 600, 1);
    prepare();
}
```

Kemudian kami menambahkan beberapa objek yang dibutuhkan pada method prepare() seperti berikut ini.

```
private void prepare()
{
    Pacman pacman = new Pacman();
    addObject(pacman,38,38);
    Wall wall = new Wall();
    addObject (wall,110,101);
    Wall wall2 = new Wall();
    addObject (wall2,688,212);
    Wall wall3 = new Wall();
    addObject (wall3,444,597);
    Wall wall4 = new Wall();
    addObject (wall4,462,13);
    Wall wall5 = new Wall();
    addObject (wall5,687,13);
    Wall wall6 = new Wall();
    addObject (wall6,463,437);
    Wall wall7 = new Wall();
    addObject (wall7,343,223);
    Wall wall8 = new Wall();
    addObject (wall8,234,597);
    Wall wall9 = new Wall();
    addObject (wall9,626,597);
    Wall wall10 = new Wall();
    addObject (wall10,13,474);
}
```

Selanjutnya untuk menambahkan level pada Game, disini kami membuat method `setUpLevel()` yang dimana berfungsi jika level naik maka akan ditambahkan objek-objek baru seperti Actor Ghost dan Point baru yang telah tersebar pada world.

```
public void setUpLevel()
{
    if(currentLevel == 2){
        addObject(new Ghost2(),654,358);
        addObject(new Point(),320,109);
        addObject(new Point(),519,109);
        addObject(new Point(),690,109);
        addObject(new Point(),105,224);
        addObject(new Point(),372,528);
        addObject(new Point(),712,331);
        addObject(new Point(),486,356);
        addObject(new Point(),631,528);
        addObject(new Point(),105,361);
        addObject(new Point(),345,356);
    }else if(currentLevel == 3){
        addObject(new Ghost(),294,452);
        addObject(new Point(),320,109);
        addObject(new Point(),519,109);
        addObject(new Point(),690,109);
        addObject(new Point(),105,224);
        addObject(new Point(),372,528);
        addObject(new Point(),712,331);
        addObject(new Point(),486,356);
        addObject(new Point(),631,528);
        addObject(new Point(),105,361);
        addObject(new Point(),345,356);
    }
}
```

Kemudia kami membuat lagi method `increaseLevel()` yang berisi code untuk menaikkan level dan juga memanggil method `setUpLevel()` agar ketika level naik makan world akan ditambahkan objek baru dan point yang ada pada method `setUpLevel()`.

```
public void increaseLevel(){
    currentLevel++;
    setUpLevel();
}
```

2. Actor Pac Man.

Pac Man merupakan sebuah tokoh utama pada Game yang kami buat , yang bisa digerakkan oleh user. Untuk dapat menjalankan Game ini maka diperlukan method yang berisi dengan intruksi-intruksi yang dapat diperintahkan kepada Game. Berikut ini method yang telah kami buat.

a. `pacmanMove()`.

Di method ini berisi code kondisional `if(Greenfoot.isKeyDown())` yang di dalam argument tersebut diberi tombol kearah mana **Pac Man** akan berjalan, disini kami menulis arah "**right**", "**left**", "**down**", dan "**up**". Intruksi dari code kondisionalnya adalah `setLocation(getX(), getY())`; yang dimana code ini berfungsi untuk mengubah posisi setiap tombol arah yng ditekan. Perubahan posisi ini terjadi ketika penambahan atau pengurangan angka pada posisi X dan Y. Posisi X merupakan arah horizontal dan Posisi Y merupakan arah vertical. Lalu diberi code `setRotation()`; yang dimana berfungsi agar **Pac Man** menghadap kemana arah dia akan berjalan. Argumen pada code ini berupa derajat putaran.

```
public void pacmanMove()
{
    if(Greenfoot.isKeyDown("right")){
        setLocation(getX()+4, getY());
        setRotation(360);
        animate();
        Actor wall = getOneIntersectingObject(Wall.class);
        if (wall !=null)
        {
            setLocation (getX()-4,getY());
        }
    }
    if(Greenfoot.isKeyDown("left")){
        setLocation(getX()-4, getY());
        setRotation(180);
        animate();
        Actor wall = getOneIntersectingObject(Wall.class);
        if (wall !=null)
        {
            setLocation (getX()+4,getY());
        }
    }
    if(Greenfoot.isKeyDown("down")){
        setLocation(getX(), getY()+4);
        setRotation(90);
        animate();
        Actor wall = getOneIntersectingObject(Wall.class);
```

Setelah itu, agar **Pac Man** tidak dapat menembus dinding maka akan diberi code berikut, yang dimana code ini berfungsi untuk mengembalikan posisi **Pac Man** ketika menyentuh dinding sehingga seolah-olah **Pac Man** tidak bisa menembus dinding.

```
Actor wall = getOneIntersectingObject(Wall.class);  
if (wall !=null)  
{  
    setLocation (getX(),getY()+4);  
}
```

b. **animate().**

Method `animate()` ini berfungsi untuk menganimasikan actor **Pac Man**. Method ini dipanggil pada method `pacmanMove()` sehingga setiap **Pac Man** berjalan maka akan berganti gambar **Pac man** yang membuka mulut dan menutup mulut, seolah-olah **Pac Man** menggerakan mulutnya sendiri.

```
public void animate()  
{  
    if(getImage()==pacmanimage1){  
        setImage(pacmanimage2);  
    }  
    else{  
        setImage(pacmanimage1);  
    }  
}
```

c. **detectGhost()**

Method `detectGhost()` ini memiliki fungsi ketika Pac Man menabrak class Ghost maka Greenfoot akan memainkan sound effect yang berada pada Folder Sounds, lalu Pac Man akan diposisikan ke posisi awal dan kemudian memanggil method `removeLife()`.

```
public void detectGhost()  
{  
    if(isTouching(Ghost.class)){  
        Greenfoot.playSound("hurt03.wav");  
        setLocation(28,39);  
        removeLife();  
    }  
}
```

d. **removeLife()**

Pada method `removeLife()` ini berisi instruksi untuk mengurangi nyawa **Pac Man** dan mengatur ulang waktu untuk mundur ke 1000, lalu memanggil method `gameOver()` dan `showStatus()`.

```
public void removeLife()
{
    lives--;
    timer = 1000;
    gameOver();
    showStatus();
}
```

e. **gameOver()**

Method `gameOver()` berisi instruksi yang berfungsi untuk mengubah gambar **Pac Man** menjadi gambar Game Over lalu Greenfoot akan memainkan sound Game Over dan menghentikan Program tersebut. Method ini dipanggil dalam method `removeLife()` sehingga ketika disaat sisa nyawa telah habis maka Program akan menyatakan Game Over.

```
public void gameOver()
{
    if(lives <= 0){
        setImage(gameoverimage);
        setLocation(422,353);
        setRotation(360);
        Greenfoot.playSound("game over.wav");
        Greenfoot.stop();
    }
}
```

f. **showStatus()**

Method `showStatus()` ini berisi code untuk menampilkan teks yang nantinya berfungsi untuk menampilkan level pada Game, jumlah nyawa yang tersisa, dan jumlah Score yang telah didapat.

```
public void showStatus()
{
    getWorld().showText("Level : "+level, 70, 520);
    getWorld().showText("Lives : "+lives, 70, 540);
    getWorld().showText("Score : "+score, 70, 560);
}
```


g. **eatPoint()**

Fungsi dari Method `eatPoint()` yaitu jika **Pac Man** memakan point maka Greenfoot akan memuatarkan sound `nom.wav`, dan akan menghapus point yang telah dimakan dan memanggil method `increaseScore()` dan timer akan ditambah 200.

```
public void eatPoint()
{
    if(isTouching(Point.class)){
        Greenfoot.playSound("nom.wav");
        removeTouching(Point.class);
        increaseScore();
        timer = timer + 200;
    }
}
```

h. **detectPortal()**

Fungsi Method ini yaitu jika actor **Pac Man** menyentuh Portal maka informasi level akan bertambah dan Greenfoot akan memainkan sound `cartoon-yuppie.wav`, lalu akan mengembalikan posisi **Pac Man** ke awal dan setelah itu akan menaikkan level pada `myWorld`, kemudian akan mengatur ulang kembali waktu mundur ke 1000.

```
public void detectPortal()
{
    if(isTouching(Portal.class)){
        level++;
        Greenfoot.playSound("cartoon-yuppie.wav");
        setLocation(28,39);
        MyWorld myworld = (MyWorld)getWorld();
        myworld.increaseLevel();
        timer = 1000;
    }
}
```

i. **youWin()**

Fungsi Method ini yaitu jika **Pac Man** dapat melewati 3 Level pada Game maka Greenfoot akan memutar sound `you win.wav` lalu gambar **Pac Man** akan berubah menjadi gambar Winner yang telah diposisikan berada di tengah world dan dirotasikan agar tidak salah rotasi. Kemudian Program Game akan berhenti.


```

public void youWin()
{
    if(level > 3){
        setImage(wingameimage);
        Greenfoot.playSound("you win.wav");
        setLocation(422,353);
        setRotation(360);
        Greenfoot.stop();
    }
}

```

j. **increaseScore()**

Berfungsi untuk menambahkan 10 score dan memanggil method showStatus(). Method ini dipanggil pada method eatPoints() sehingga ketika **Pac Man** memakan point yang ada pada Game maka score Game akan bertambah 10.

```

public void increaseScore()
{
    score+=10;
    showStatus();
}

```

k. **updateTimer()**

Method ini berfungsi untuk membuat waktu berjalan mundur. Kemudian saat waktu telah habis maka method akan memanggil removeLive() sehingga nyawa akan berkurang lalu **Pac Man** akan kembali ke posisi awal dengan disertai sound hurt03.mav.

```

public void updateTimer()
{
    timer--;
    getWorld().showText("Time Left = "+timer, 70, 580);
    if(timer < 1){
        removeLife();
        Greenfoot.playSound("hurt03.wav");
        setLocation(28,39);
    }
}

```

3. Actor Class Ghost.

Ghost memiliki peran sebagai musuh pada Game ini sehingga **Pac Man** harus berusaha untuk menghindar dari class ghost. Pada class ghost terdapat 2 Ghost yang berwarna pink dan merah. Keduanya memiliki method yang sama. Berikut method yang telah kami buat pada subclass Ghost.

a. **changeDirection().**

Method ini berfungsi untuk merubah pergerakan Ghost yang awalnya turun menjadi naik.

```
public void changeDirection()
{
    if (direction.equals("down")){
        direction = "up";
    }
}
```

b. **atWall().**

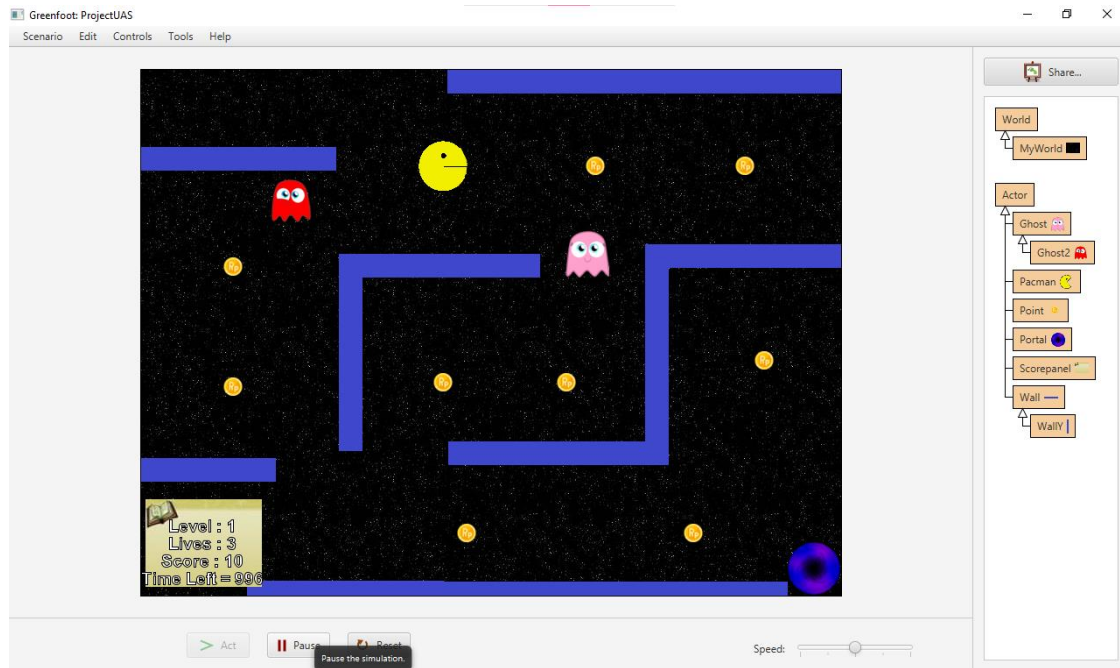
Method ini berfungsi jika menabrak class dinding maka kecepatan Ghost akan berkurang sementara, kemudian method akan memanggil changeDirection() agar Ghost mengubah arah pergerakannya.

```
public void atWall()
{
    Actor wall = getOneIntersectingObject(Wall.class);
    if (wall != null){
        speed = -speed;
        changeDirection();
    }
}
```

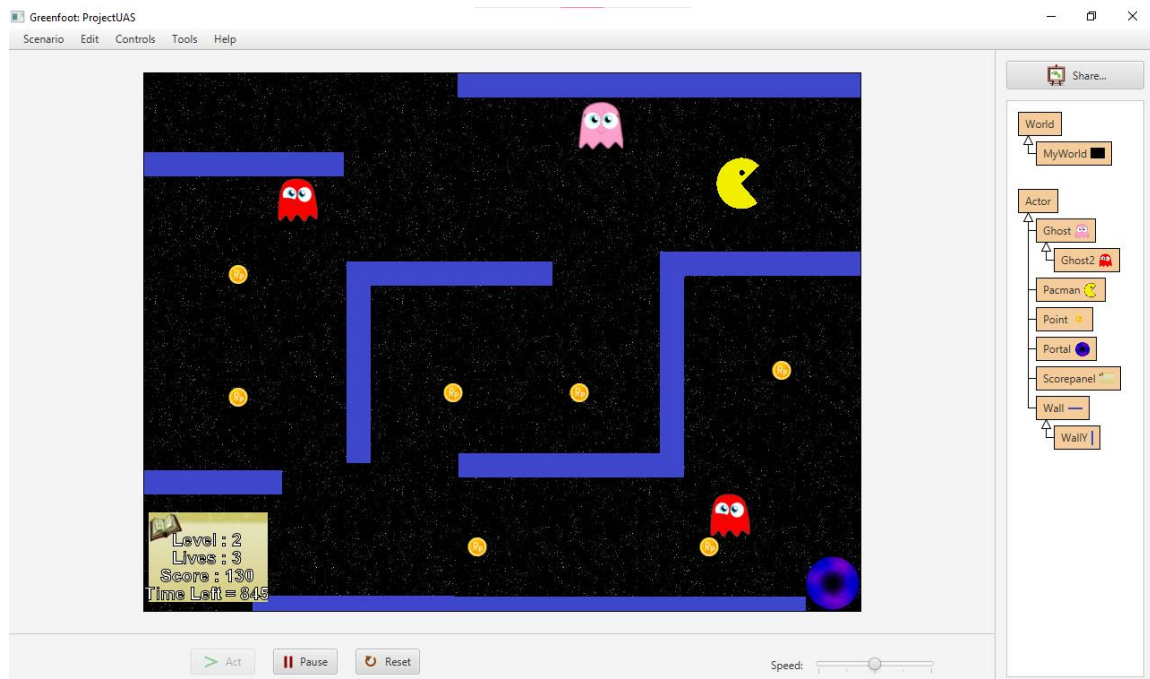
TAMPILAN GAME SAAT DIJALANKAN

Disaat Game dijalankan, Game akan berada di Level 1 dengan terdapat 2 Ghost yang bergerak ke atas dan bawah untuk menghalangi jalan **Pac Man** dan terdapat points yang tersebar.

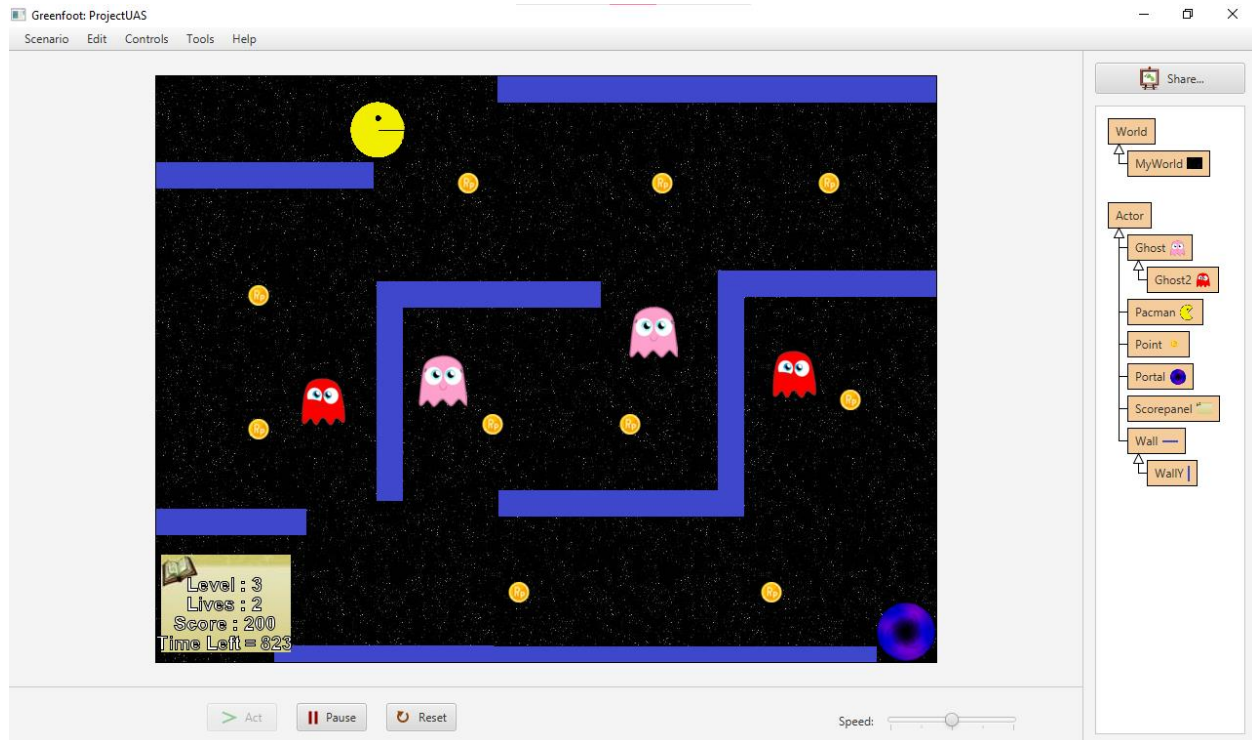
Berikut tampilan Game di Level 1.



Jika **Pac Man** berhasil menghindari Ghost dan masuk ke dalam Portal maka Game akan berlanjut ke Level 2 dengan tambahan Ghost dan Points yang telah dimakan akan ditambahkan kembali.



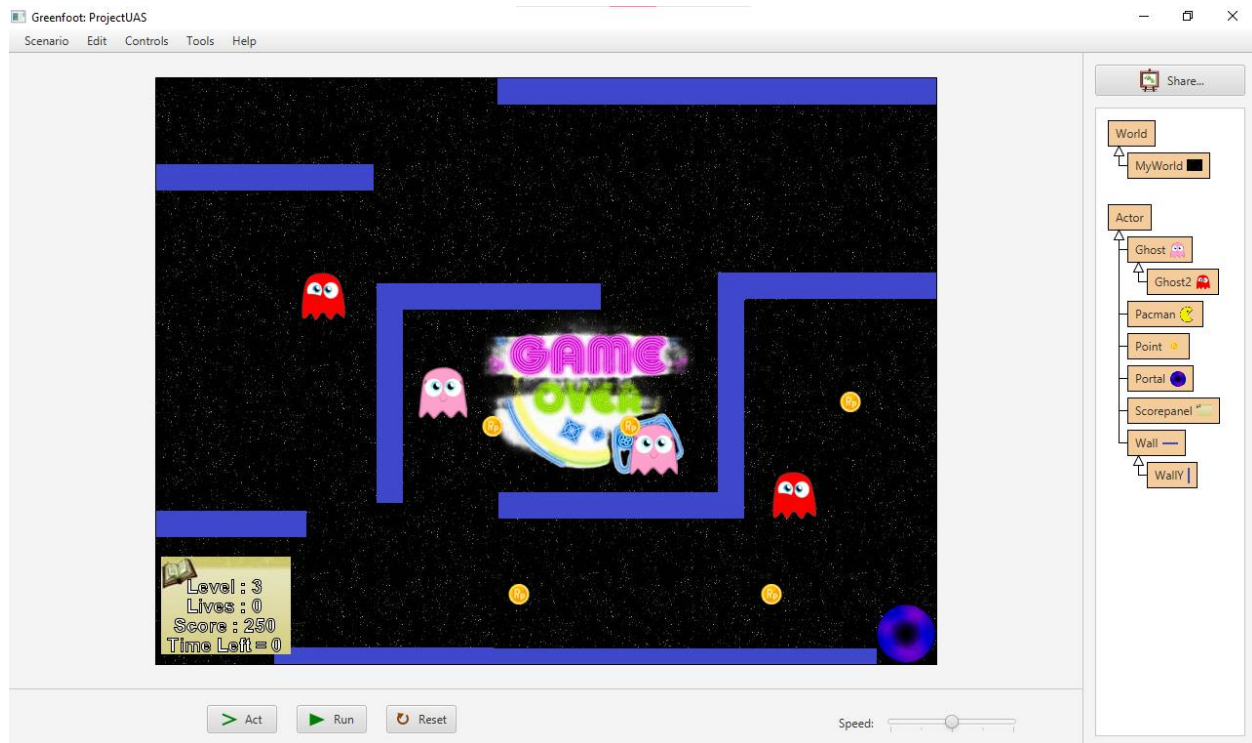
Dan jika **Pac Man** berhasil melewati Level 2 maka Game akan naik ke Level 3 dengan tambahan yang sama yaitu Ghost dan Point akan tersebar.



Kemudian jika **Pac Man** telah berhasil melewati semua level maka Game akan menyatakan bahwa user telah memenangkan Game ini seperti tampilan berikut.



Akan tetapi jika **Pac Man** kehilangan ke-3 nyawanya maka Game akan menampilkan gambar Game Over yang dimana telah menyatakan bahwa user telah kalah dalam permainan. Berikut adalah tampilannya.



Demikian Laporan yang telah kami buat berdasarkan Game Pac Man buatan kami melalui Software Greenfoot. Laporan ini dibuat untuk menyelesaikan/memenuhi Ujian Akhir Semester (UAS) kami pada Mata Kuliah Praktikum Pemrograman Berorientasi Objek (PBO). Kami mohon maaf jika ada kekeliruan dan kesalahan dalam pembuatan Project kami, Semoga Project Game kami bisa memenuhi nilai Akhir kami dengan baik, dan juga semoga Game kami bisa menghibur orang yang memainkannya, Terimakasih.