

# **MATA KULIAH PROJECT MEMBUAT GAME DI GREENFOOT**

Praktikum Pemrograman Berorientasi Objek (PBO)

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



INF2153

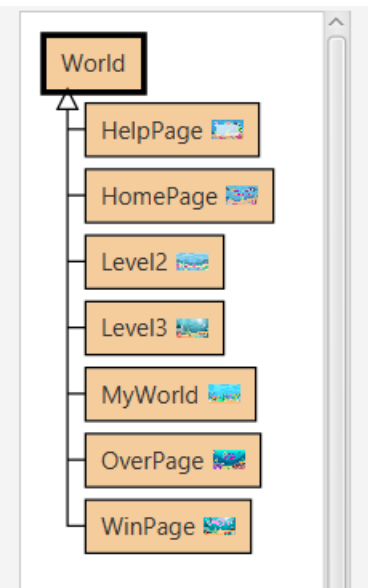
UJIAN AKHIR SEMESTER (UAS)

Oleh Kelompok 22

- |                          |                 |
|--------------------------|-----------------|
| 1. Muhammad Satya Satria | : 2211102441074 |
| 2. Alda Ramadani         | : 2211102441049 |
| 3. Siti Naila Fitriah    | : 2211102441110 |

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur  
Samarinda 2023

## Class World :



Kelas **World** memainkan peran yang penting dalam lingkungan Greenfoot karena menjadi wadah utama di mana permainan berlangsung. Ini menyediakan kerangka kerja yang diperlukan untuk menangani objek-objek, interaksi, dan logika permainan secara umum.

### Fungsi Utama :

- **Menjalankan Permainan:** Kelas **World** menyediakan metode **act()** yang dipanggil secara terus-menerus dan mengatur perilaku objek dalam permainan, seperti pembaruan posisi, interaksi, atau perubahan kondisi.
- **Mengelola Objek:** Kelas **World** bertanggung jawab untuk menambahkan objek-objek ke dalamnya menggunakan metode **addObject()**, menghapus objek menggunakan **removeObject()**, atau memeriksa interaksi antar objek.

## 1. HelpPage

### Output :

```
HelpPage - UAS-PBO-fish-fish
Class Edit Tools Options
HelpPage X
Compile Undo Cut Copy Paste Find... Close

import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class HelpPage here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class HelpPage extends World
{
    /**
     * Constructor for objects of class HelpPage.
     *
     */
    public HelpPage()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(600, 400, 1);
        prepare();
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
    private void prepare()
    {
        BackButton backButton = new BackButton();
        addObject(backButton, 81, 484);
        backButton.setLocation(192, 468);
        backButton.setLocation(88, 474);
    }
}
```



Penjelasan :

Kelas **HelpPage** bertujuan untuk menggambarkan dunia (world) yang menampilkan halaman bantuan (help page) dalam game. Laporan ini akan memberikan penjelasan tentang struktur kelas, metode yang digunakan, dan tujuan dari kelas tersebut.

**Deskripsi Kelas:** Kelas **HelpPage** merupakan kelas yang mewarisi **World** dari Greenfoot. Hal ini berarti kelas **HelpPage** adalah dunia (world) di mana elemen-elemen dalam halaman bantuan ditampilkan. Kelas ini digunakan untuk membangun tampilan yang menampilkan petunjuk atau informasi tentang permainan kepada pengguna.

#### Metode Utama:

1. **public HelpPage():** Constructor kelas ini membuat dunia dengan ukuran 950x550 piksel dan menyiapkan halaman bantuan dengan menyiapkan elemen-elemen yang diperlukan.
2. **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek yang diperlukan di dalam dunia (world). Di sini, **BackButton** (tombol kembali) ditambahkan ke halaman bantuan pada posisi tertentu.

**Tujuan Kelas:** Kelas **HelpPage** bertujuan untuk memberikan tampilan interaktif kepada pengguna dengan informasi atau panduan terkait permainan. Halaman ini menampilkan objek-objek seperti tombol kembali atau informasi-informasi yang membantu pemain dalam memahami aturan atau cara memainkan permainan.

#### Analisis Tambahan:

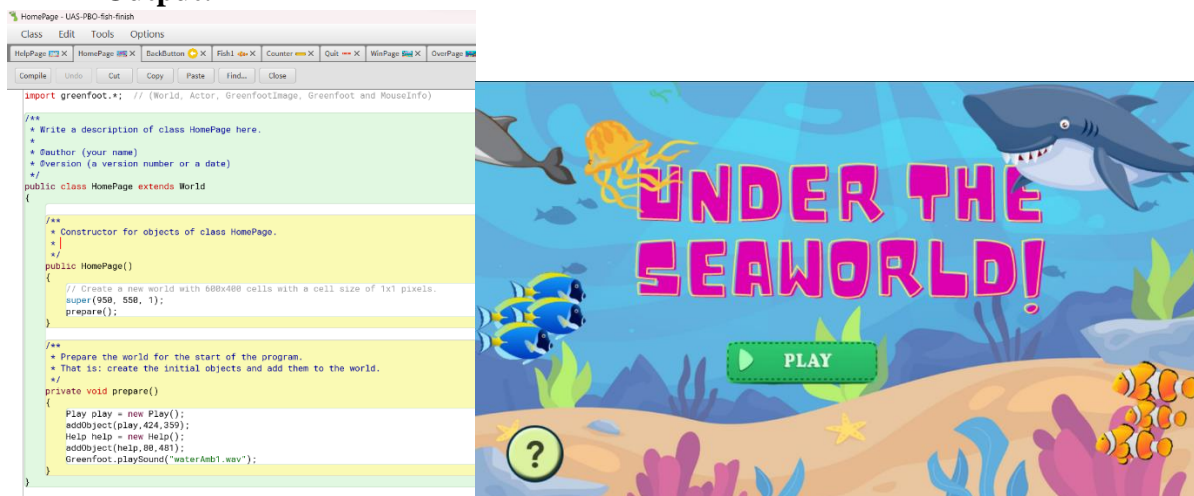
- Ukuran dunia yang besar (950x550 piksel) dapat memberikan ruang yang cukup untuk menampilkan konten bantuan dengan jelas dan terstruktur.
- Penggunaan objek **BackButton** menunjukkan adanya kemungkinan adanya fungsionalitas untuk kembali ke halaman sebelumnya atau menu utama dari halaman bantuan.

Kelas **HelpPage** memiliki peran penting dalam menyediakan informasi dan bantuan kepada pemain dalam memahami atau mempelajari permainan. Desain yang jelas dan navigasi yang baik akan meningkatkan pengalaman pengguna dalam menjelajahi halaman bantuan ini.

Harapannya, laporan ini memberikan gambaran tentang struktur, fungsi, dan tujuan dari kelas **HelpPage** dalam konteks permainan yang sedang dikembangkan.

## 2. HomePage

### Output:



Penjelasan :

**Deskripsi Kelas:**

Kelas **HomePage** adalah kelas yang mewarisi kelas **World** dari Greenfoot. Tujuan dari kelas ini adalah untuk menyediakan tampilan halaman utama (home page) dari permainan Anda. Ini adalah layar pertama yang akan ditampilkan kepada pengguna ketika permainan dimulai.

**Konstruktor:**

- **public HomePage():** Konstruktor kelas ini membuat dunia (world) dengan ukuran 950x550 piksel dan menginisialisasi tampilan halaman utama. **super(950, 550, 1)** digunakan untuk membuat dunia dengan ukuran yang diinginkan.

**Metode Utama:**

- **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek yang akan ditampilkan di dalam dunia (world). Pada contoh ini, dua objek (**Play** dan **Help**) ditambahkan ke halaman utama pada posisi tertentu. Selain itu, suara ("waterAmb1.wav") dimainkan menggunakan **Greenfoot.playSound()**.

**Tujuan Kelas:** Kelas **HomePage** bertujuan untuk memberikan tampilan awal kepada pemain ketika permainan dimulai. Di dalamnya terdapat objek-objek seperti tombol 'Play' untuk memulai permainan dan tombol 'Help' untuk mengarahkan pengguna ke halaman bantuan atau petunjuk permainan.

**Analisis Tambahan:**

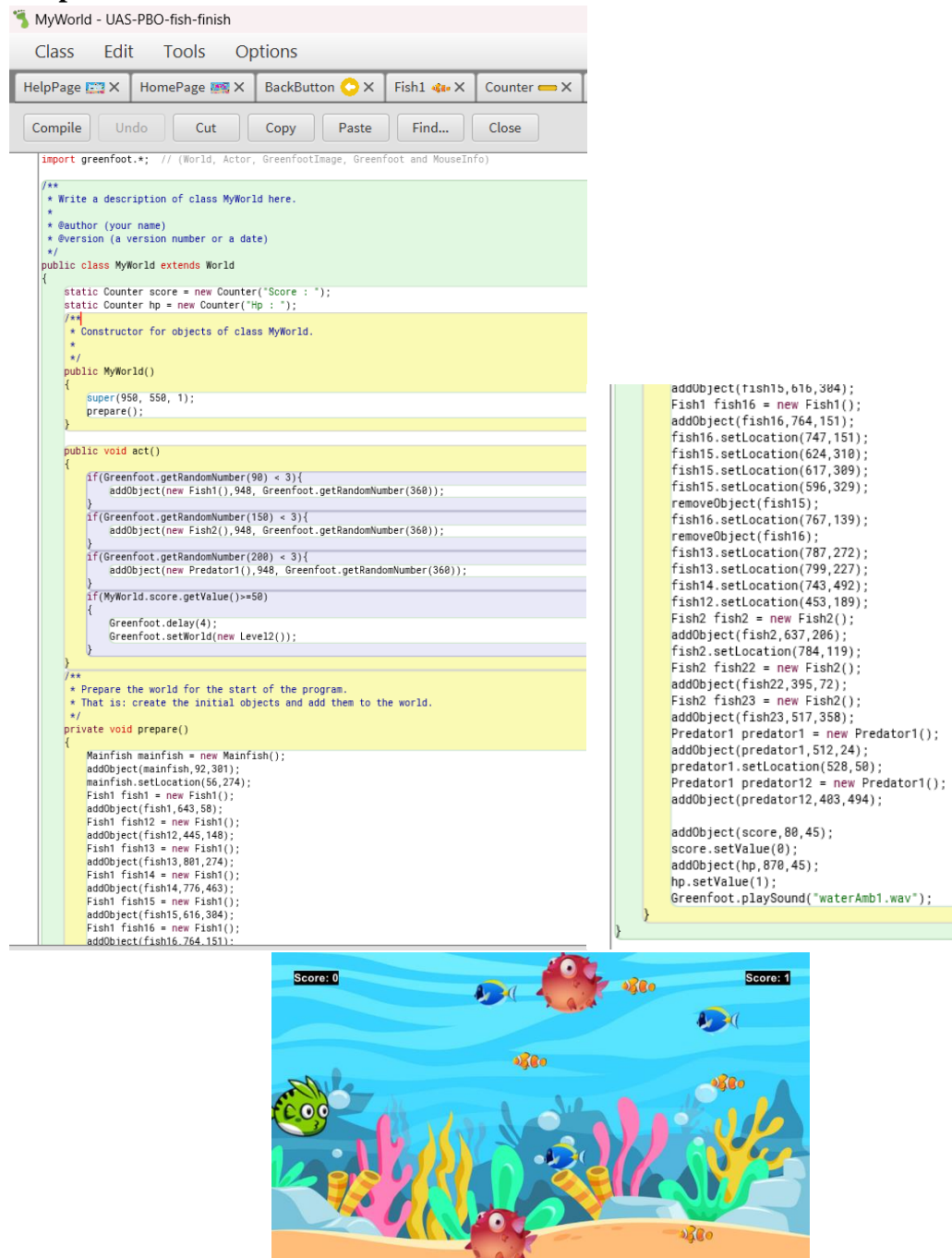
- Penambahan objek **Play** dan **Help** menunjukkan adanya fungsionalitas untuk memulai permainan dan akses ke halaman bantuan.
- Memainkan suara ("waterAmb1.wav") pada saat tampilan halaman utama dimuat dapat menambah atmosfer atau suasana dalam permainan.

*Kelas **HomePage** berperan penting dalam memberikan tampilan awal yang menarik serta memberikan akses cepat kepada pengguna untuk memulai permainan atau mendapatkan bantuan jika diperlukan.*

Penjelasan di atas memberikan gambaran tentang fungsi dan tujuan kelas **HomePage** dalam konteks permainan yang sedang dikembangkan.

### 3. MyWorld (Level 1)

#### Output :



Penjelasan :

#### Deskripsi Kelas:

Kelas **MyWorld** adalah kelas yang mewarisi **World** dari Greenfoot. Tujuannya adalah untuk menjadi dunia utama permainan Anda. Di dalamnya, elemen-elemen permainan ditampilkan, dan logika permainan dieksekusi.

#### Konstruktor:

- **public MyWorld():** Konstruktor kelas ini membuat dunia (world) dengan ukuran 950x550 piksel dan menginisialisasi tampilan untuk memulai permainan.

**Metode Utama:**

- **public void act():** Metode ini dipanggil secara terus-menerus selama permainan berlangsung. Di dalamnya terdapat logika untuk menambahkan objek-objek ke dunia (world) secara acak (seperti **Fish1**, **Fish2**, dan **Predator1**). Selain itu, terdapat logika yang memeriksa skor (**MyWorld.score.getValue()**) dan bila mencapai nilai tertentu (50 dalam contoh ini), permainan akan pindah ke **Level2**.

**Metode Prepare:**

- **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek awal yang akan ditampilkan di dalam dunia (world). Dalam contoh ini, terdapat beberapa objek **Mainfish**, **Fish1**, **Fish2**, dan **Predator1** yang ditambahkan ke dunia. Selain itu, **Counter** untuk skor (**score**) dan poin darah (**hp**) juga ditambahkan ke dunia pada lokasi tertentu.

**Tujuan Kelas:** Kelas **MyWorld** bertujuan untuk menjadi lingkungan permainan utama di mana pemain akan berinteraksi. Ini mencakup penambahan objek, pengaturan skor, dan memulai logika permainan.

**Analisis Tambahan:**

- Penambahan objek **Mainfish**, **Fish1**, **Fish2**, dan **Predator1** di dalam metode **prepare** menunjukkan bahwa ini adalah elemen-elemen dasar dalam permainan yang akan ditampilkan di dunia.
- Penggunaan **Greenfoot.playSound()** menunjukkan adanya pemutaran suara ("waterAmb1.wav") untuk menambah atmosfer atau efek dalam permainan.

*Kelas **MyWorld** memiliki peran utama dalam menginisialisasi permainan, menambahkan objek-objek awal, dan memulai logika permainan dalam lingkungan dunia yang dibuat.*

Penjelasan ini memberikan gambaran tentang fungsi, tujuan, dan bagaimana kelas **MyWorld** berperan dalam permainan yang sedang dikembangkan.

## 4. Level 2

### Output ;



### Penjelasan:

**Deskripsi Kelas:** Kelas **Level2** adalah kelas yang mewarisi **World** dari Greenfoot. Tujuannya adalah menjadi level kedua dalam permainan Anda. Di dalamnya, elemen-elemen permainan ditampilkan, dan logika permainan lanjutan dieksekusi.

### Konstruktor:

- **public Level2():** Konstruktor kelas ini membuat dunia (world) dengan ukuran 950x550 piksel dan menginisialisasi tampilan untuk level kedua permainan.

**Metode Utama:**

- **public void act():** Metode ini dipanggil secara terus-menerus selama permainan berlangsung. Di dalamnya terdapat logika untuk menambahkan objek-objek ke dunia (world) secara acak (seperti **Fish1**, **Fish2**, **Fish3**, **Predator1**, **Predator2**). Selain itu, terdapat logika yang memeriksa skor (**Level2.score.getValue()**) dan bila mencapai nilai tertentu (100 dalam contoh ini), permainan akan pindah ke **Level3**.

**Metode Prepare:**

- **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek awal yang akan ditampilkan di dalam dunia (world). Di dalamnya terdapat beberapa objek **Mainfish**, **Fish1**, **Fish2**, **Fish3**, **Predator1**, dan **Predator2** yang ditambahkan ke dunia. Selain itu, **Counter** untuk skor (**score**) dan poin darah (**hp**) juga ditambahkan ke dunia pada lokasi tertentu.

**Tujuan Kelas:** Kelas **Level2** bertujuan untuk menjadi level kedua dalam permainan, memberikan tantangan baru dengan objek-objek yang berbeda dan meningkatkan kesulitan permainan dari level sebelumnya.

**Analisis Tambahan:**

- Penambahan objek-objek spesifik seperti **Fish1**, **Fish2**, **Fish3**, **Predator1**, dan **Predator2** menunjukkan bahwa level ini memiliki variasi elemen permainan yang lebih kompleks dibandingkan level sebelumnya.
- Penggunaan **Greenfoot.playSound()** menunjukkan adanya pemutaran suara ("waterAmb1.wav") untuk mempertahankan atmosfer atau efek dalam permainan.

*Kelas **Level2** adalah bagian penting dari progresi permainan dan memberikan tingkat kesulitan yang lebih tinggi dibandingkan dengan level sebelumnya.*

Penjelasan ini memberikan gambaran tentang fungsi, tujuan, dan bagaimana kelas **Level2** berperan dalam permainan yang sedang dikembangkan.



## 5. Level 3

### Output :

```
Level3 - UAS-PBO-fish-finish
Class Edit Tools Options
HelpPage X HomePage X BackButton X Fish1 X Counter X
Compile Undo Cut Copy Paste Find... Close

/**
 * Write a description of class Level3 here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Level3 extends World
{
    static Counter score = new Counter("Score : ");
    static Counter hp = new Counter("hp : ");

    /**
     * Constructor for objects of class Level3.
     */
    public Level3()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(950, 550, 1);
        prepare();
    }

    public void act()
    {
        if(Greenfoot.getRandomNumber(160)< 3){
            addObject(new Fish1(), 948, Greenfoot.getRandomNumber(240));
        }
        if(Greenfoot.getRandomNumber(150)< 3){
            addObject(new Fish2(), 948, Greenfoot.getRandomNumber(360));
        }
        if(Greenfoot.getRandomNumber(180)< 3){
            addObject(new Fish3(), 948, Greenfoot.getRandomNumber(400));
        }
        if(Greenfoot.getRandomNumber(500)< 3){
            addObject(new Predator1(), 948, Greenfoot.getRandomNumber(360));
        }
        if(Greenfoot.getRandomNumber(450)< 3){
            addObject(new Predator2(), 948, Greenfoot.getRandomNumber(500));
        }
        if(Greenfoot.getRandomNumber(600)< 3){
            addObject(new Predator3(), 948, Greenfoot.getRandomNumber(140));
        }
        if(Level3.score.getValue()==200)
        {
            Greenfoot.delay(4);
            Greenfoot.setWorld(new WinPage());
        }
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
    private void prepare()
    {
        Mainfish mainfish = new Mainfish();
        addObject(mainfish, 32, 271);
        Fish1 fish1 = new Fish1();
        addObject(fish1, 635, 76);
        Fish1 fish12 = new Fish1();
        addObject(fish12, 685, 446);
        Fish2 fish2 = new Fish2();
        addObject(fish2, 429, 167);
        Fish2 fish22 = new Fish2();
        addObject(fish22, 735, 346);
        Fish3 fish3 = new Fish3();
        addObject(fish3, 768, 102);
        Fish3 fish32 = new Fish3();
        addObject(fish32, 553, 524);
        Predator2 predator2 = new Predator2();
        addObject(predator2, 626, 229);
        Predator1 predator1 = new Predator1();
        addObject(predator1, 735, 494);
        Predator3 predator3 = new Predator3();
        addObject(predator3, 563, 400);
        predator3.setLocation(709, 356);
        Fish3 fish33 = new Fish3();
        addObject(fish33, 212, 54);
        predator2.setLocation(433, 48);

        addObject(score, 80, 45);
        score.setValue(0);
        addObject(hp, 870, 45);
        hp.setValue(0);
        Greenfoot.playSound("waterAmb1.wav");
    }
}
```



Penjelasan:

**Deskripsi Kelas:** Kelas **Level3** adalah kelas yang mewarisi **World** dari Greenfoot. Tujuannya adalah menjadi level ketiga dalam permainan Anda. Di dalamnya, elemen-elemen permainan ditampilkan, dan logika permainan lanjutan dieksekusi.

**Konstruktor:**

- **public Level3():** Konstruktor kelas ini membuat dunia (world) dengan ukuran 950x550 piksel dan menginisialisasi tampilan untuk level ketiga permainan.

**Metode Utama:**

- **public void act():** Metode ini dipanggil secara terus-menerus selama permainan berlangsung. Di dalamnya terdapat logika untuk menambahkan objek-objek ke dunia (world) secara acak (seperti **Fish1**, **Fish2**, **Fish3**, **Predator1**, **Predator2**, **Predator3**). Selain itu, terdapat logika yang memeriksa skor (**Level3.score.getValue()**) dan bila mencapai nilai tertentu (200 dalam contoh ini), permainan akan pindah ke **WinPage**.

**Metode Prepare:**

- **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek awal yang akan ditampilkan di dalam dunia (world). Di dalamnya terdapat beberapa objek **Mainfish**, **Fish1**, **Fish2**, **Fish3**, **Predator1**, **Predator2**, **Predator3** yang ditambahkan ke dunia. Selain itu, **Counter** untuk skor (**score**) dan poin darah (**hp**) juga ditambahkan ke dunia pada lokasi tertentu.

**Tujuan Kelas:** Kelas **Level3** bertujuan untuk menjadi level ketiga dalam permainan, memberikan tantangan yang lebih tinggi dan kompleksitas yang lebih besar dibandingkan dengan level sebelumnya.

**Analisis Tambahan:**

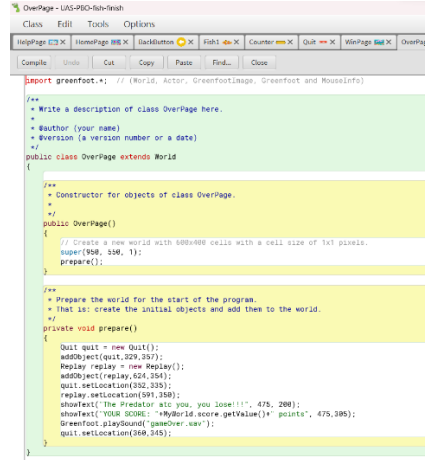
- Penambahan objek-objek spesifik seperti **Fish1**, **Fish2**, **Fish3**, **Predator1**, **Predator2**, **Predator3** menunjukkan bahwa level ini memiliki variasi elemen permainan yang lebih kompleks dibandingkan level sebelumnya.
- Penggunaan **Greenfoot.playSound()** menunjukkan adanya pemutaran suara ("waterAmb1.wav") untuk mempertahankan atmosfer atau efek dalam permainan.

*Kelas **Level3** adalah bagian penting dari progresi permainan dan menawarkan tingkat kesulitan yang lebih tinggi serta tantangan yang lebih besar dibandingkan dengan level sebelumnya.*

Penjelasan ini memberikan gambaran tentang fungsi, tujuan, dan bagaimana kelas **Level3** berperan dalam permainan yang sedang dikembangkan.

## 5. OverPage

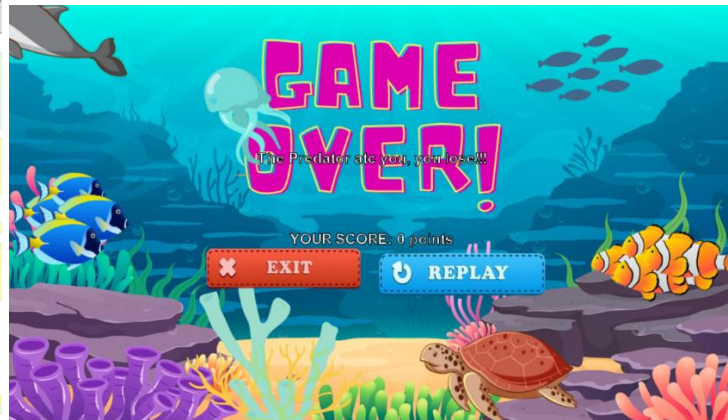
### Output :



```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class OverPage here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class OverPage extends World
{
    /**
     * Constructor for objects of class OverPage.
     *
     * @param OverPage()
     */
    public OverPage()
    {
        // Create a new world with 600x600 cells with a cell size of 1x1 pixels.
        super(600, 600, 1);
        prepare();
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
    private void prepare()
    {
        Quit quit = new Quit();
        addObject(quit, 325, 257);
        Replay replay = new Replay();
        addObject(replay, 425, 255);
        quit.setLocation(325, 255);
        replay.setLocation(425, 255);
        showText("The Predator ate you, you lose!!!", 475, 200);
        showText("YOUR SCORE: " + MyWorld.score.getValue() + " points", 475, 205);
        Greenfoot.playSound("gameOver.wav");
        quit.setLocation(325, 255);
    }
}
```



Penjelasan :

**Deskripsi Kelas:** Kelas **OverPage** adalah kelas yang mewarisi **World** dari Greenfoot. Tujuannya adalah untuk menampilkan halaman akhir permainan setelah pemain kalah atau gagal dalam permainan.

**Konstruktor:**

- **public OverPage():** Konstruktor kelas ini membuat dunia (world) dengan ukuran 950x550 piksel dan menginisialisasi tampilan untuk halaman akhir permainan.

**Metode Prepare:**

- **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek yang akan ditampilkan di dalam dunia (world). Dalam halaman akhir permainan ini, terdapat dua objek, yaitu **Quit** dan **Replay**, yang mungkin berfungsi sebagai tombol untuk keluar dari permainan atau memulai ulang permainan. Selain itu, terdapat pesan teks yang memberitahukan pemain bahwa mereka kalah ("The Predator attack you, you lose!!!") dan menampilkan skor terakhir yang didapat oleh pemain ("YOUR SCORE: " + MyWorld.score.getValue() + " points"). Terakhir, terdapat pemutaran suara ("gameOver.wav") untuk memberikan efek suara saat permainan berakhir.

**Tujuan Kelas:** Kelas **OverPage** bertujuan untuk memberikan tampilan yang informatif setelah permainan berakhir dengan hasil kekalahan. Ini mungkin menampilkan opsi untuk keluar dari permainan atau memulai ulang permainan.

**Analisis Tambahan:**

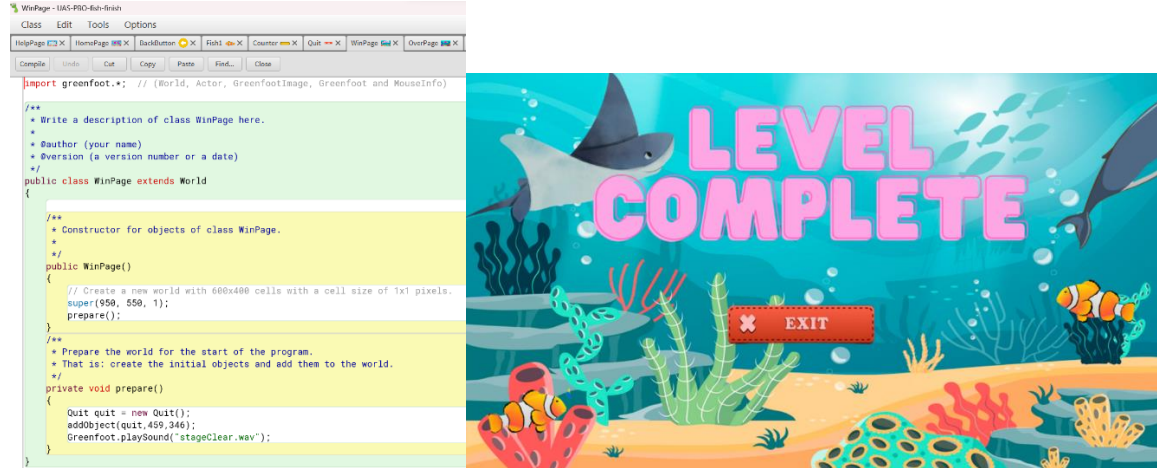
- Penggunaan objek **Quit** dan **Replay** menunjukkan adanya opsi untuk mengakhiri permainan atau memulai kembali permainan.
- Pesan teks yang ditampilkan memberitahukan pemain tentang hasil akhir permainan dan skor terakhir yang berhasil dicapai.

*Kelas **OverPage** adalah bagian akhir dari pengalaman permainan dan menampilkan informasi akhir kepada pemain setelah permainan berakhir.*

Penjelasan ini memberikan gambaran tentang fungsi, tujuan, dan bagaimana kelas **OverPage** berperan dalam permainan yang sedang dikembangkan.

## 6. WinPage

### Output :



Penjelasan:

**Deskripsi Kelas:** Kelas **WinPage** adalah kelas yang mewarisi **World** dari Greenfoot. Tujuannya adalah untuk menampilkan halaman kemenangan setelah pemain berhasil menyelesaikan permainan atau mencapai kondisi kemenangan tertentu.

**Konstruktor:**

- **public WinPage():** Konstruktor kelas ini membuat dunia (world) dengan ukuran 950x550 piksel dan menginisialisasi tampilan untuk halaman kemenangan permainan.

**Metode Prepare:**

- **private void prepare():** Metode ini dipanggil dari konstruktor untuk menyiapkan objek-objek yang akan ditampilkan di dalam dunia (world). Dalam halaman kemenangan ini, terdapat satu objek yaitu **Quit**, yang mungkin berfungsi sebagai tombol untuk keluar dari permainan setelah pemain menyelesaikan permainan.

**Tujuan Kelas:** Kelas **WinPage** bertujuan untuk memberikan tampilan yang informatif setelah pemain berhasil menyelesaikan permainan atau mencapai kondisi kemenangan tertentu.

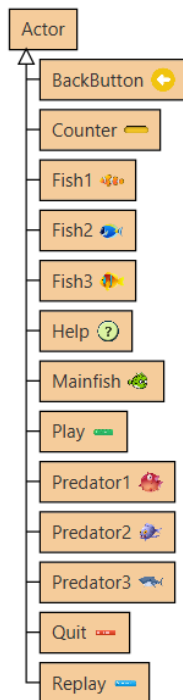
**Analisis Tambahan:**

- Penggunaan objek **Quit** menunjukkan adanya opsi untuk mengakhiri permainan setelah pemain berhasil menyelesaikan permainan.

*Kelas **WinPage** adalah bagian akhir dari pengalaman permainan dan menampilkan informasi kemenangan kepada pemain setelah berhasil menyelesaikan permainan.*

Penjelasan ini memberikan gambaran tentang fungsi, tujuan, dan bagaimana kelas **WinPage** berperan dalam permainan yang sedang dikembangkan.

## Class Actor :



Kelas **Actor** adalah kelas dasar yang digunakan untuk mewakili objek dalam permainan. Objek **Actor** dapat memiliki perilaku yang diimplementasikan melalui metode **act()** serta dapat berinteraksi dengan objek lain di dalam dunia permainan. Hal ini membuatnya menjadi komponen penting dalam pembangunan permainan menggunakan platform Greenfoot.

### Fungsi Utama :

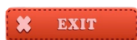
- **Interaksi dalam Dunia:** Objek yang merupakan turunan dari kelas **Actor** dapat berinteraksi dengan objek lain di dalam dunia (kelas **World**).
- **Metode act():** Setiap objek **Actor** memiliki metode **act()** yang dapat diimplementasikan untuk menentukan perilaku atau logika objek tersebut. Metode ini dipanggil secara teratur selama permainan berjalan.



- Tombol Back



- Tombol Help



- Tombol Exit



- Tombol Play



- Fish 1



- Fish 2



- Fish 3



- Mainfish



- Predator 1



- Predator 2



- Pedator 3