

**PROJECT UJIAN AKHIR SEMESTER PEMROGRAMAN
BERBASIS OBJEK “MEMBUAT GAME DARI GREENFOOT”**



DISUSUN OLEH :

- 1. RACHMAT RIZKYY SAPUTRA (2211102441005)**
- 2. LIANAWATI SUWITO (2211102441024)**
- 3. NABIYA SHEVA ISWANDARI (2211102441006)**

2023/2024

DAFTAR ISI

DAFTAR ISI.....	ii
A. Pendahuluan	1
1. Deskripsi Singkat Proyek	1
2. Tujuan Pembuatan Game.....	1
3. Metodologi yang Digunakan	2
B. Desain Game.....	2
1. Konsep Permainan.....	2
2. Struktur Kode Awal	2
3. Pemilihan Konsep PBO/OOP yang Diterapkan	2
C. Implementasi Konsep PBO/OOP	3
1. Kelas dan Objek yang Dibangun	3
2. Pewarisan (Inheritance).....	4
3. Polimorfisme	4
4. Enkapsulasi	4
5. Interaksi Antar Objek	5
6. Penggunaan Overriding dan Overloading.....	5
D. Kesimpulan	6
1. Ringkasan Hasil Proyek	6
2. Pembelajaran yang Didapat.....	6
3. Saran untuk Pengembangan Selanjutnya	6
E. Lampiran	6
1. Snippet Kode Penting.....	6
2. Screenshots Gameplay	16

A. Pendahuluan

1. Deskripsi Singkat Proyek

Game "Death Target" adalah permainan di mana pemain memainkan peran seorang penembak yang harus bertahan hidup dari serangan zombie. Pemain menggunakan senjata untuk menembak zombie dan memperoleh poin. Tujuan utama adalah memberikan pengalaman bermain yang menegangkan dengan fokus pada kualitas grafis, desain menarik, sistem poin peringkat, dan integrasi suara yang memukau.

2. Tujuan Pembuatan Game

- a) Menghadirkan pengalaman bermain yang menegangkan: Game "Death Target" membangun pengalaman menegangkan dengan menghadirkan situasi yang menantang bagi pemain. Ini tercipta melalui:
 - 1) Penembakan dan Serangan Zombie: Pemain harus bertahan dengan menembakkan proyektil ke arah zombie yang mendekat, menciptakan ketegangan saat mencoba menghindari serangan dan bertahan hidup.
 - 2) Kesulitan Bertambah: Dengan berjalannya waktu, kecepatan munculnya zombie meningkat, meningkatkan tingkat kesulitan dan ketegangan dalam permainan.
- b) Menyajikan kualitas grafis dan desain yang menarik:
 - 1) Grafis yang Diperbaiki: Pemilihan gambar yang sesuai untuk karakter, senjata, dan musuh menambahkan aspek visual yang menarik bagi permainan.
 - 2) Desain Antar Muka Pengguna (UI): Penambahan elemen seperti tombol interaktif, tampilan skor, dan batas kesehatan memberikan pengalaman pengguna yang lebih baik.
- c) Mengimplementasikan sistem poin peringkat untuk meningkatkan interaktifitas:
 - 1) Poin Peringkat dan Uang dalam Permainan: Sistem poin skor dan uang memotivasi pemain untuk bermain lebih baik, mengejar skor tertinggi, dan mendapatkan akses ke peningkatan senjata.
 - 2) Penyimpanan Data Poin: Informasi skor dan uang ditampilkan pada antarmuka pengguna, memungkinkan pemain melacak kemajuan mereka dalam permainan.

- d) Integrasi suara yang memukau untuk meningkatkan pengalaman bermain:
- 1) Efek Suara yang Meningkatkan Atmosfer: Penggunaan suara seperti efek suara tembakan, serangan zombie, dan suara latar belakang, seperti musik atau efek lingkungan, menambahkan kedalaman pada pengalaman bermain.
 - 2) Memberikan Respons Tindakan: Suara yang dihasilkan oleh interaksi pemain dengan lingkungan, seperti suara tembakan atau suara kemenangan saat mencapai tujuan tertentu, memberikan respons langsung yang memperkaya pengalaman bermain.
 - 3) Dengan menyelaraskan aspek-aspek ini, game "Death Target" menciptakan pengalaman yang mendebarkan, visual yang menarik, sistem interaktif yang memotivasi, dan suasana yang diwarnai oleh suara-suara yang memukau, semuanya digabungkan untuk menciptakan pengalaman bermain yang memuaskan bagi pengguna.

3. Metodologi yang Digunakan

Penggunaan platform Greenfoot untuk mengembangkan permainan dengan bahasa pemrograman Java. Fokus pada penerapan konsep Pemrograman Berbasis Objek untuk membangun struktur permainan.

B. Desain Game

1. Konsep Permainan
 - a) Pemain berperan sebagai penembak yang harus bertahan hidup dari serangan zombie.
 - b) Penggunaan senjata untuk menembak zombie dan memperoleh poin.
2. Struktur Kode Awal
 - a) Objek MyWorld sebagai dunia permainan.
 - b) Objek Player sebagai karakter pemain.
 - c) Objek Zombie sebagai musuh yang harus dikalahkan.
 - d) Objek Counter sebagai penghitung skor, uang, dan waktu.
 - e) Objek HealthBar untuk menampilkan tingkat kesehatan pemain.
 - f) Objek PlayButton untuk memulai permainan.
3. Pemilihan Konsep PBO/OOP yang Diterapkan

- a) kelas dan objek yang terstruktur.
- b) Pewarisan digunakan dalam beberapa kelas untuk membangun hubungan logis antar kelas.
- c) Polimorfisme terlihat dalam beberapa perilaku objek yang bervariasi.
- d) Enkapsulasi digunakan untuk melindungi data dan memastikan akses yang aman.

C. Implementasi Konsep PBO/OOP

1. Kelas dan Objek yang Dibangun

- a) MyWorld: Dunia permainan yang memuat elemen-elemen utama seperti Player, Counter, HealthBar, WeaponButton, SuperPower, dan Zombie.
- b) Player: Mengontrol karakter utama pemain. Mengimplementasikan fungsi-fungsi untuk pergerakan, menembak, dan mendeteksi tabrakan dengan Zombie.
- c) Counter: Menampilkan informasi skor, uang, dan waktu permainan. Memanfaatkan fungsi `updateImage()` untuk mengupdate tampilan counter pada layar.
- d) HealthBar: Menampilkan tingkat kesehatan pemain. Memanfaatkan fungsi `loseHealth()` untuk mendeteksi tabrakan dengan Zombie dan mengurangi kesehatan pemain.
- e) PlayButton: Tombol untuk memulai permainan. Memanfaatkan fungsi `act()` untuk mendeteksi klik mouse dan memulai permainan.
- f) WeaponButton: Memanfaatkan kelas Counter untuk mengupdate upgrade senjata saat tombol ditekan. Menggunakan fungsi `updateImage()` untuk menampilkan tombol yang interaktif pada layar.
- g) SuperPower: Menangani penghitungan dan penggunaan kekuatan khusus. Fungsi `useSuper()` digunakan untuk mengakumulasi kekuatan super dan digunakan saat jumlah tertentu tercapai.
- h) Projectile: Objek yang ditembakkan oleh Player untuk melawan Zombie. Bergerak secara linier ke arah yang dituju.
- i) Zombie: Musuh dalam permainan yang bergerak ke arah Player dan dapat ditembak untuk mendapatkan skor.

2. Pewarisan (Inheritance)

Pewarisan adalah konsep dalam pemrograman berbasis objek di mana sebuah kelas dapat mewarisi sifat dan perilaku dari kelas lain. Contoh-contoh penggunaan pewarisan dalam game "Death Target":

- a) Kelas Zombie dan Player: Mungkin ada kelas "Zombie" dan "Player" yang mewarisi sifat dasar dari kelas "Character". Kedua kelas ini memiliki atribut dan perilaku yang mirip, seperti kemampuan bergerak dan berinteraksi dengan objek lain.
- b) Weapon Classes: Kelas senjata (weapon classes) dapat mewarisi sifat dasar dari kelas "Weapon". Misalnya, ada senjata-senjata spesifik seperti senapan, pistol, dan shotgun yang memiliki fungsi tembakan yang berbeda, tetapi memanfaatkan sifat dasar dari kelas induk "Weapon".

3. Polimorfisme

Polimorfisme memungkinkan objek untuk memiliki banyak bentuk, artinya, objek dari kelas yang sama dapat berperilaku berbeda. Contoh implementasi polimorfisme dalam game "Death Target":

- a) Perilaku Serangan: Objek yang melakukan serangan, seperti "Zombie" dan "Player", mungkin memiliki metode "attack()". Namun, setiap kelas tersebut mengimplementasikan metode ini dengan cara yang berbeda sesuai dengan jenis serangannya, seperti jarak serangan, kecepatan, atau dampak yang berbeda.
- b) Projectile Behavior: Objek "Projectile" (peluru) memiliki perilaku berbeda berdasarkan senjata yang digunakan. Polimorfisme memungkinkan penggunaan metode "shoot()" yang dapat menghasilkan gerakan peluru yang berbeda tergantung pada jenis senjata yang sedang digunakan.

4. Enkapsulasi

Enkapsulasi melibatkan pembungkusan (encapsulation) data dan metode yang beroperasi pada data ke dalam satu unit tunggal atau kelas. Ini membantu melindungi data dan memastikan akses yang aman. Contoh dalam game "Death Target":

- a) Private Attributes: Misalnya, atribut-atribut seperti "health" dari pemain atau "health" dari zombie mungkin ditetapkan sebagai private, yang berarti hanya metode internal dalam kelas yang memiliki akses langsung ke data tersebut.
- b) Getter dan Setter Methods: Enkapsulasi juga melibatkan penggunaan metode getter dan setter untuk mengakses dan mengubah data yang dilindungi. Misalnya, metode "getHealth()" dan "setHealth()" dapat digunakan untuk membaca dan mengubah nilai "health" dengan cara yang terkontrol.

5. Interaksi Antar Objek

Interaksi antar objek adalah inti dari banyak permainan, di mana objek saling berinteraksi untuk menciptakan gameplay yang dinamis. Contoh implementasi dalam game "Death Target":

- a) Collision Detection: Ketika proyektil bertabrakan dengan zombie atau pemain, terjadi interaksi yang memicu pengurangan kesehatan atau perubahan nilai skor.
- b) Pembaruan Antarmuka Pengguna: Interaksi antara pemain dan antarmuka pengguna terjadi saat skor atau uang diubah setelah interaksi tertentu dalam permainan, seperti menembak zombie atau meningkatkan senjata.

6. Penggunaan Overriding dan Overloading

- a) Overriding: Dalam konteks "Death Target", misalnya, metode "move()" pada kelas "Player" dapat di-override untuk menambahkan perilaku tambahan atau mengubah perilaku bawaan dari kelas "Actor" yang menjadi superclass.
- b) Overloading: Dalam hal ini, metode "fireProjectile()" pada kelas "Player" mungkin di-overload untuk menerima parameter yang berbeda tergantung pada tingkat upgrade senjata yang dimiliki pemain (misalnya, jumlah proyektil yang ditembakkan).

D. Kesimpulan

1. Ringkasan Hasil Proyek

Proyek "Death Target" berhasil menghadirkan pengalaman bermain yang menegangkan dengan integrasi konsep PBO/OOP.

2. Pembelajaran yang Didapat

- a) Penerapan konsep PBO/OOP membantu dalam mengorganisir dan memperjelas struktur permainan.
- b) Tantangan teknis yang dihadapi memberikan pelajaran berharga dalam pengembangan game.

3. Saran untuk Pengembangan Selanjutnya

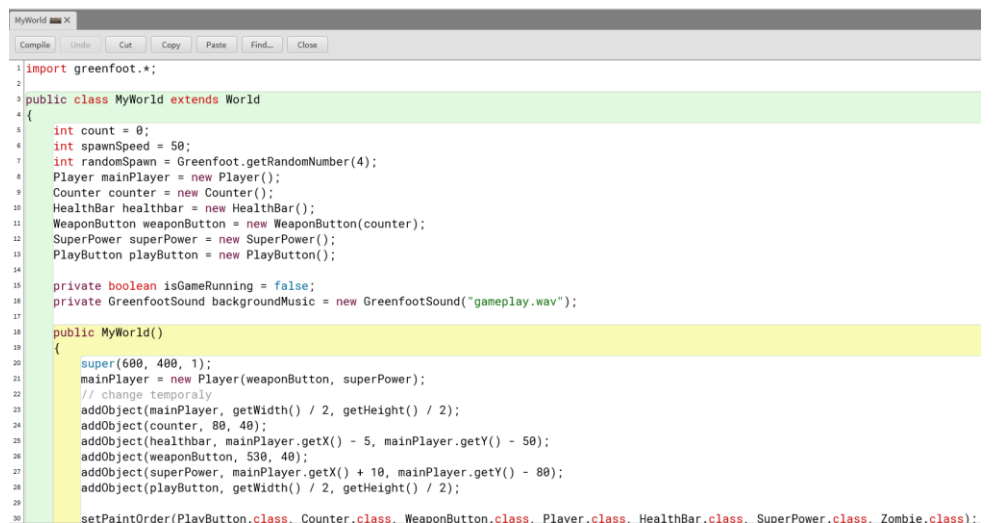
- a) Peningkatan lebih lanjut pada grafis dan desain untuk meningkatkan daya tarik visual.
- b) Penambahan fitur atau level baru untuk meningkatkan kompleksitas dan keunikan permainan.

E. Lampiran

1. Snippet Kode Penting

Kode utama dari berbagai kelas seperti :

a) MyWorld



```
1 import greenfoot.*;
2
3 public class MyWorld extends World
4 {
5     int count = 0;
6     int spawnSpeed = 50;
7     int randomSpawn = Greenfoot.getRandomNumber(4);
8     Player mainPlayer = new Player();
9     Counter counter = new Counter();
10    HealthBar healthbar = new HealthBar();
11    WeaponButton weaponButton = new WeaponButton(counter);
12    SuperPower superPower = new SuperPower();
13    PlayButton playButton = new PlayButton();
14
15    private boolean isGameRunning = false;
16    private GreenfootSound backgroundMusic = new GreenfootSound("gameplay.wav");
17
18    public MyWorld()
19    {
20        super(600, 400, 1);
21        mainPlayer = new Player(weaponButton, superPower);
22        // change temporary
23        addObject(mainPlayer, getWidth() / 2, getHeight() / 2);
24        addObject(counter, 80, 40);
25        addObject(healthbar, mainPlayer.getX() - 5, mainPlayer.getY() - 50);
26        addObject(weaponButton, 530, 40);
27        addObject(superPower, mainPlayer.getX() + 10, mainPlayer.getY() - 80);
28        addObject(playButton, getWidth() / 2, getHeight() / 2);
29
30        setPaintOrder(PlayButton.class, Counter.class, WeaponButton.class, Player.class, HealthBar.class, SuperPower.class, Zombie.class);
```



```

MyWorld x
Compile Undo Cut Copy Paste Find... Close
30 setPaintOrder(PlayButton.class, Counter.class, WeaponButton.class, Player.class, HealthBar.class, SuperPower.class, Zombie.class);
31 }
32
33 public Player getPlayer()
34 {
35     return mainPlayer;
36 }
37
38 public void act()
39 {
40     count++;
41     spawnZombies();
42     System.out.println(spawnSpeed);
43
44     if (count % 600 == 0)
45     {
46         spawnSpeed--;
47     }
48
49     if (spawnSpeed < 2)
50     {
51         spawnSpeed = 1;
52     }
53
54     if (Greenfoot.mouseClicked(playButton) && !isGameRunning) {
55         setGameRunning(true);
56         removeObject(playButton);
57         // Add any additional initialization or actions you want when starting the game
58     }
59 }

```

```

MyWorld x
Compile Undo Cut Copy Paste Find... Close
60 // Check if the background music has finished playing and restart it
61 if (!backgroundMusic.isPlaying()) {
62     backgroundMusic.playLoop();
63 }
64
65
66 public void spawnZombies()
67 {
68     if (count % spawnSpeed == 0)
69     {
70         randomSpawn = Greenfoot.getRandomNumber(4);
71         switch (randomSpawn)
72         {
73             case 0:
74                 // spawn zombie randomly from the top side
75                 addObject(new Zombie(1, "small_zombie.png", mainPlayer, counter), Greenfoot.getRandomNumber(getWidth()), 0);
76                 break;
77             case 1:
78                 // spawn zombie randomly from the bottom side
79                 addObject(new Zombie(2, "zombieboss.png", mainPlayer, counter), Greenfoot.getRandomNumber(getWidth()), getHeight());
80                 break;
81             case 2:
82                 // spawn zombie randomly from the left side
83                 addObject(new Zombie(1, "small_zombie.png", mainPlayer, counter), 0, Greenfoot.getRandomNumber(getHeight()));
84                 break;
85             case 3:
86                 // spawn zombie randomly from the right side
87                 addObject(new Zombie(2, "zombieboss.png", mainPlayer, counter), getWidth(), Greenfoot.getRandomNumber(getHeight()));
88                 break;
89         }
90     }
91 }

```

```

MyWorld x
Compile Undo Cut Copy Paste Find... Close
92
93 public void setGameRunning(boolean running)
94 {
95     isGameRunning = running;
96 }
97
98 public boolean isGameRunning()
99 {
100     return isGameRunning;
101 }
102 }

```

b) Player

```
Player X
Compile Undo Cut Copy Paste Find... Close

1 import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
2
3 /**
4  * Write a description of class Player here.
5  *
6  * @author (your name)
7  * @version (a version number or a date)
8  */
9 public class Player extends Actor
10 {
11     private int speed = 3;
12     int time = 0;
13     WeaponButton weaponButton;
14     SuperPower superPower;
15     int superTimer;
16     GreenfootSound laserSound = new GreenfootSound("laser.wav");
17
18     public Player()
19     {
20     }
21
22
23     public Player(WeaponButton weaponButton, SuperPower superPower)
24     {
25         this.superPower = superPower;
26         this.weaponButton = weaponButton;
27     }
28
29
30     public void act()
31     {
32         time++;
33         turnAround();
34         moveAround();
35         fireProjectile();
36         hitByZombie();
37         superPowerUsed();
38     }
39
40     public void turnAround()
41     {
42         if (Greenfoot.getMouseInfo() != null)
43         {
44             turnTowards(Greenfoot.getMouseInfo().getX(), Greenfoot.getMouseInfo().getY());
45         }
46     }
47
48     public void moveAround()
49     {
50         if (Greenfoot.isKeyDown("a"))
51         {
52             setLocation(getX() - speed, getY());
53         }
54
55         if (Greenfoot.isKeyDown("w"))
56         {
57             setLocation(getX(), getY() - speed);
58         }
59     }
60 }
```

```
Player X
Compile Undo Cut Copy Paste Find... Close

60 if (Greenfoot.isKeyDown("d"))
61 {
62     setLocation(getX() + speed, getY());
63 }
64
65 if (Greenfoot.isKeyDown("s"))
66 {
67     setLocation(getX(), getY() + speed);
68 }
69 }
70
71 public void fireProjectile()
72 {
73     if (Greenfoot.isKeyDown("space") && weaponButton.weaponUpgrade == 1)
74     {
75         Projectile projectile = new Projectile("yellow.png");
76         getWorld().addObject(projectile, getX(), getY());
77         projectile.setRotation(getRotation());
78         projectile.move(25);
79         laserSound.play();
80     }
81
82     if (Greenfoot.isKeyDown("space") && weaponButton.weaponUpgrade == 2)
83     {
84         Projectile projectile = new Projectile("yellow.png");
85         getWorld().addObject(projectile, getX(), getY());
86         projectile.setRotation(getRotation() - 5);
87         projectile.move(20);
88         laserSound.play();
89     }
90 }
```

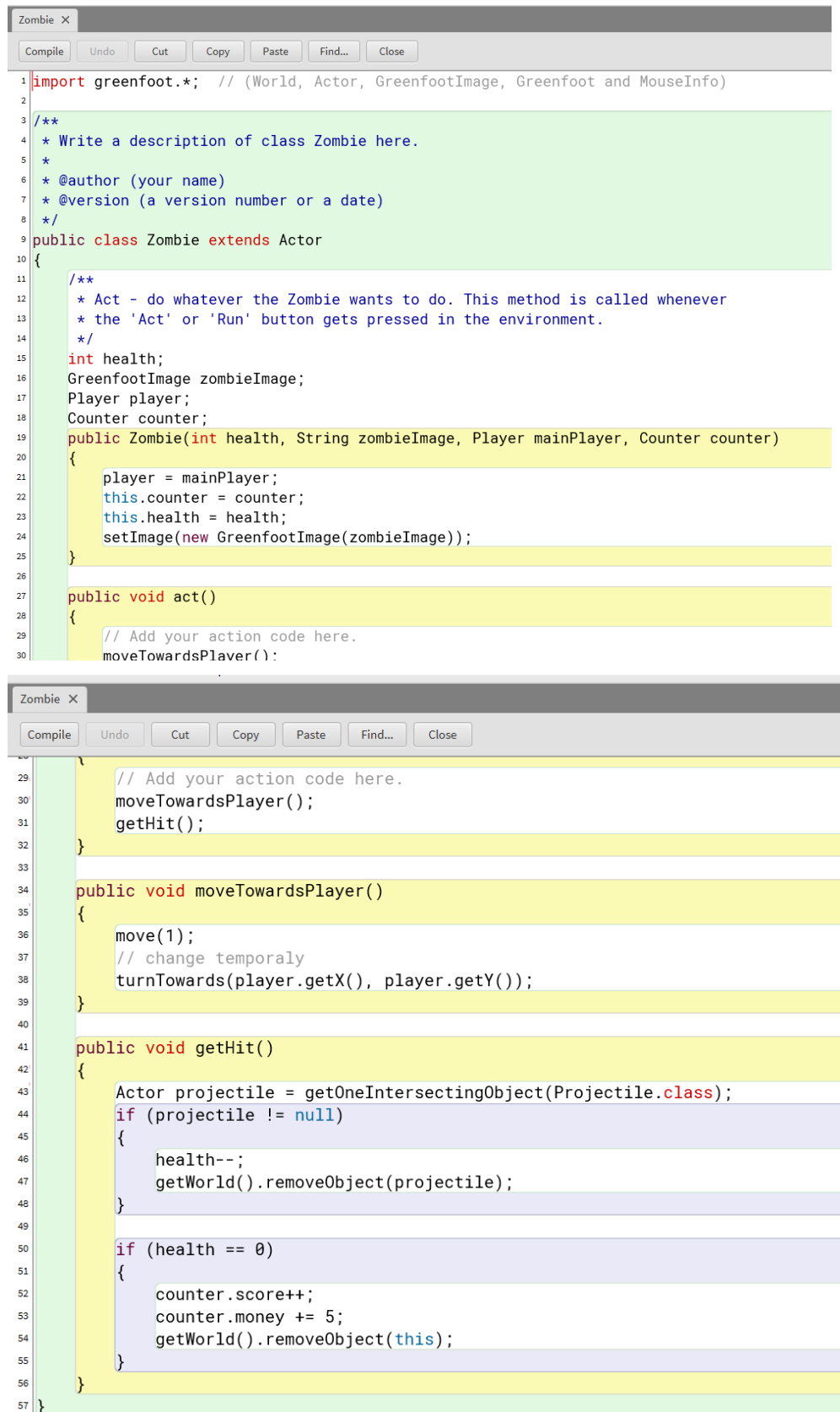
```
Player X
Compile Undo Cut Copy Paste Find... Close

90 Projectile projectile2 = new Projectile("yellow.png");
91 getWorld().addObject(projectile2, getX(), getY());
92 projectile2.setRotation(getRotation() + 5);
93 projectile2.move(20);
94 }
95
96 if (Greenfoot.isKeyDown("space") && weaponButton.weaponUpgrade == 3)
97 {
98     Projectile projectile = new Projectile("yellow.png");
99     getWorld().addObject(projectile, getX(), getY());
100     projectile.setRotation(getRotation() - 5);
101     projectile.move(20);
102     laserSound.play();
103
104     Projectile projectile2 = new Projectile("yellow.png");
105     getWorld().addObject(projectile2, getX(), getY());
106     projectile2.setRotation(getRotation() + 5);
107     projectile2.move(20);
108
109     Projectile projectile3 = new Projectile("yellow.png");
110     getWorld().addObject(projectile3, getX(), getY());
111     projectile3.setRotation(getRotation());
112     projectile3.move(20);
113 }
114 }
115
116 public void superPowerUsed()
117 {
118     if (superPower.superCount > 99 && superTimer < 30)
```

```
Player X
Compile Undo Cut Copy Paste Find... Close
119 {
120     Projectile projectile = new Projectile("blue.png");
121     getWorld().addObject(projectile, getX(), getY());
122     projectile.setRotation(getRotation() - 60);
123     projectile.move(20);
124     laserSound.play();
125
126     Projectile projectile2 = new Projectile("blue.png");
127     getWorld().addObject(projectile2, getX(), getY());
128     projectile2.setRotation(getRotation() + 60);
129     projectile2.move(20);
130
131     Projectile projectile3 = new Projectile("blue.png");
132     getWorld().addObject(projectile3, getX(), getY());
133     projectile3.setRotation(getRotation());
134     projectile3.move(20);
135
136     Projectile projectile4 = new Projectile("blue.png");
137     getWorld().addObject(projectile4, getX(), getY());
138     projectile4.setRotation(getRotation() - 180);
139     projectile4.move(20);
140
141     Projectile projectile5 = new Projectile("blue.png");
142     getWorld().addObject(projectile5, getX(), getY());
143     projectile5.setRotation(getRotation() + 120);
144     projectile5.move(20);
145
146     Projectile projectile6 = new Projectile("blue.png");
147     getWorld().addObject(projectile6, getX(), getY());
148     projectile6.setRotation(getRotation() - 120);
```

```
Player X
Compile Undo Cut Copy Paste Find... Close
146     Projectile projectile6 = new Projectile("blue.png");
147     getWorld().addObject(projectile6, getX(), getY());
148     projectile6.setRotation(getRotation() - 120);
149     projectile6.move(20);
150
151     superTimer++;
152 }
153
154 if (superTimer > 29)
155 {
156     superPower.superCount = 0;
157     superTimer = 0;
158 }
159
160
161 public boolean hitByZombie()
162 {
163     Actor zombie = getOneObjectAtOffset(0, 0, Zombie.class);
164
165     if (zombie != null)
166     {
167         return true;
168     }
169     else
170     {
171         return false;
172     }
173 }
174 }
```

c) Zombie



```

1 import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
2
3 /**
4  * Write a description of class Zombie here.
5  *
6  * @author (your name)
7  * @version (a version number or a date)
8  */
9 public class Zombie extends Actor
10 {
11     /**
12     * Act - do whatever the Zombie wants to do. This method is called whenever
13     * the 'Act' or 'Run' button gets pressed in the environment.
14     */
15     int health;
16     GreenfootImage zombieImage;
17     Player player;
18     Counter counter;
19     public Zombie(int health, String zombieImage, Player mainPlayer, Counter counter)
20     {
21         player = mainPlayer;
22         this.counter = counter;
23         this.health = health;
24         setImage(new GreenfootImage(zombieImage));
25     }
26
27     public void act()
28     {
29         // Add your action code here.
30         moveTowardsPlayer();
31
32     }
33
34     public void moveTowardsPlayer()
35     {
36         move(1);
37         // change temporary
38         turnTowards(player.getX(), player.getY());
39     }
40
41     public void getHit()
42     {
43         Actor projectile = getOneIntersectingObject(Projectile.class);
44         if (projectile != null)
45         {
46             health--;
47             getWorld().removeObject(projectile);
48         }
49
50         if (health == 0)
51         {
52             counter.score++;
53             counter.money += 5;
54             getWorld().removeObject(this);
55         }
56     }
57 }

```

d) Counter

```
Counter X
[Compile] [Undo] [Cut] [Copy] [Paste] [Find...] [Close]

1 import greenfoot.*;
2
3 public class Counter extends Actor {
4     int score;
5     int money;
6     int time;
7
8     public Counter() {
9         // updateImage();
10    }
11
12    public void act() {
13        time++;
14        updateImage();
15    }
16
17    private void updateImage() {
18        World world = getWorld();
19        MyWorld myWorld = (MyWorld) world;
20
21        GreenfootImage image = new GreenfootImage(120, 50); // Ukuran gambar disesuaikan
22        // int borderThickness = 5; // Ketebalan border
23
24        // Mengisi seluruh area dengan warna ungu
25        image.setColor(new Color(95, 43, 178)); // Warna ungu
26        image.fill(); // Mengisi seluruh area dengan warna ungu
27
28        // Menggambar teks berwarna putih di dalam border
29        image.setColor(Color.WHITE); // Warna putih
```

```
Counter X
[Compile] [Undo] [Cut] [Copy] [Paste] [Find...] [Close]

12 public void act() {
13     time++;
14     updateImage();
15 }
16
17 private void updateImage() {
18     World world = getWorld();
19     MyWorld myWorld = (MyWorld) world;
20
21     GreenfootImage image = new GreenfootImage(120, 50); // Ukuran gambar disesuaikan
22     // int borderThickness = 5; // Ketebalan border
23
24     // Mengisi seluruh area dengan warna ungu
25     image.setColor(new Color(95, 43, 178)); // Warna ungu
26     image.fill(); // Mengisi seluruh area dengan warna ungu
27
28     // Menggambar teks berwarna putih di dalam border
29     image.setColor(Color.WHITE); // Warna putih
30     image.setFont(new Font("Arial", true, false, 18)); // Ukuran font disesuaikan
31     image.drawString("Score: " + score + "\nCash: " + money + "\nTime" + time, 10, 20);
32
33     // Menggambar border di atas teks untuk efek border
34     image.setColor(new Color(128, 0, 128)); // Warna ungu
35     // Menggambar border di atas teks
36
37     setImage(image);
38 }
39 }
```

e) HealthBar

```

HealthBar X
Compile Undo Cut Copy Paste Find... Close Source Code
1 import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
2
3 /**
4  * Write a description of class HealthBar here.
5  *
6  * @author (your name)
7  * @version (a version number or a date)
8  */
9 public class HealthBar extends Actor
10 {
11     int health = 50;
12     GreenfootSound defeatSound = new GreenfootSound("defeat.wav");
13
14     public HealthBar()
15     {
16         setImage(new GreenfootImage(52, 12));
17         getImage().drawRect(0, 0, 51, 11);
18         getImage().setColor(Color.RED);
19         getImage().fillRect(1, 1, health, 10);
20     }
21
22     public void act()
23     {
24         setImage(new GreenfootImage(80, 12));
25         getImage().drawRect(0, 0, 51, 11);
26         getImage().setColor(Color.RED);
27         getImage().fillRect(1, 1, health, 10);
28
29         World world = getWorld();
30
31         getImage().setColor(Color.RED);
32         getImage().fillRect(1, 1, health, 10);
33
34         World world = getWorld();
35         MyWorld myWorld = (MyWorld) world;
36
37         setLocation(myWorld.getPlayer().getX() - 5, myWorld.getPlayer().getY() - 20);
38         loseHealth();
39     }
40
41     public void loseHealth()
42     {
43         World world = getWorld();
44         MyWorld myWorld = (MyWorld) world;
45
46         if (myWorld.getPlayer().hitByZombie())
47         {
48             health--;
49         }
50
51         if (health <= 0)
52         {
53             getWorld().showText("You Lose! \n You survived for " + (myWorld.getPlayer().time / 60) + " seconds", getWorld().getWidth() / 2, getWorld().getHeight() / 2);
54             defeatSound.play();
55             Greenfoot.stop();
56         }
57     }
58 }

```

f) PlayButton

```

PlayButton X
Compile Undo Cut Copy Paste Find... Close Source Code
1 import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
2
3 /**
4  * Write a description of class PlayButton here.
5  *
6  * @author (your name)
7  * @version (a version number or a date)
8  */
9 public class PlayButton extends Actor
10 {
11     private GreenfootImage playImage = new GreenfootImage("playbutton.png");
12
13     public PlayButton()
14     {
15         setImage(playImage);
16     }
17
18     public void act()
19     {
20         if (Greenfoot.mouseClicked(this))
21         {
22             MyWorld world = (MyWorld) getWorld();
23             world.setGameRunning(true);
24             world.removeObject(this); // Remove the button after it's clicked
25         }
26     }
27 }

```

g) WeaponButton

```

WeaponButton X
Compile Undo Cut Copy Paste Find... Close

1 import greenfoot.*;
2
3 public class WeaponButton extends Actor {
4     Counter counter;
5     int weaponUpgrade = 1;
6
7     public WeaponButton(Counter counter) {
8         this.counter = counter;
9         updateImage();
10    }
11
12    public void act() {
13        if (Greenfoot.mousePressed(this) && counter.money > 99) {
14            counter.money -= 100;
15            weaponUpgrade++;
16        }
17
18        if (weaponUpgrade > 3)
19        {
20            weaponUpgrade = 3;
21        }
22    }
23
24    public void updateImage() {
25        GreenfootImage image = new GreenfootImage(110, 55); // Ukuran gambar disesuaikan
26        int borderThickness = 3; // Ketebalan border
27
28        // Mengisi seluruh area dengan warna ungu sebagai background
29        image.setColor(new Color(95, 43, 178)); // Warna ungu
30    }
31
32    public void act() {
33        if (Greenfoot.mousePressed(this) && counter.money > 99) {
34            counter.money -= 100;
35            weaponUpgrade++;
36        }
37
38        if (weaponUpgrade > 3)
39        {
40            weaponUpgrade = 3;
41        }
42    }
43
44    public void updateImage() {
45        GreenfootImage image = new GreenfootImage(110, 55); // Ukuran gambar disesuaikan
46        int borderThickness = 3; // Ketebalan border
47
48        // Mengisi seluruh area dengan warna ungu sebagai background
49        image.setColor(new Color(95, 43, 178)); // Warna ungu
50        image.fill(); // Mengisi seluruh area dengan warna ungu
51
52        // Menggambar teks berwarna putih di dalam background ungu
53        image.setColor(Color.WHITE); // Warna putih
54        image.setFont(new Font("Arial", true, false, 18)); // Ukuran font disesuaikan
55        image.drawString("Weapon\nUpgrade", 10, 25);
56
57        setImage(image);
58    }
59
60 }

```

h) SuperPower


```

SuperPower X
Compile Undo Cut Copy Paste Find... Close

1 import greenfoot.*;
2
3 public class SuperPower extends Actor
4 {
5     final int SUPER_POWER_LIMIT = 100;
6     int superCount;
7     int count;
8
9
10    public SuperPower()
11    {
12        setImage(new GreenfootImage(SUPER_POWER_LIMIT + 2, 12));
13        getImage().drawRect(0, 0, SUPER_POWER_LIMIT + 1, 11);
14        getImage().setColor(Color.BLUE);
15        getImage().fillRect(1, 1, superCount, 10);
16    }
17
18    public void act()
19    {
20        setImage(new GreenfootImage(SUPER_POWER_LIMIT + 2, 12));
21        getImage().drawRect(0, 0, SUPER_POWER_LIMIT + 1, 11);
22        getImage().setColor(Color.BLUE);
23        getImage().fillRect(1, 1, superCount, 10);
24
25        World world = getWorld();
26        MyWorld myWorld = (MyWorld) world;
27
28        setLocation(myWorld.getPlayer().getX() + 5, myWorld.getPlayer().getY() - 35);
29
30        // Check for the space key press to use the superpower
31
32
33
34
35
36
37
38    public void useSuper()
39    {
40        // Play the laser sound when the superpower is used
41        // laserSound.play();
42        count++;
43
44        if (count % 5 == 0)
45        {
46            superCount++;
47        }
48
49        if (superCount > SUPER_POWER_LIMIT)
50        {
51            superCount = 0;
52        }
53    }
54
55 }

```

i) Projectile

```
Projectile X
Compile Undo Cut Copy Paste Find... Close
1 import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
2
3 /**
4  * Write a description of class Projectile here.
5  *
6  * @author (your name)
7  * @version (a version number or a date)
8  */
9 public class Projectile extends Actor
10 {
11     /**
12      * Act - do whatever the Projectile wants to do. This method is called whenever
13      * the 'Act' or 'Run' button gets pressed in the environment.
14      */
15     GreenfootImage bulletImage;
16
17     public Projectile(String bulletImage)
18     {
19         setImage(new GreenfootImage(bulletImage));
20         getImage().scale(20, 20);
21     }
22
23     public void act()
24     {
25         move(10);
26         if (isAtEdge())
27         {
28             getWorld().removeObject(this);
29         }
30     }
31 }
```

2. Screenshots Gameplay

