



# **PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

**INF2153**

## **Project UAS: Block Breaker**

Dosen Pengampu:

*Sayekti Harits Suryawan, S.Kom, M.Kom.*

Oleh : Kelompok 24

- 1. Muhammad Albi Alnur                      2211102441092*
- 2. Reza Lukia Zikri                              2211102441063*
- 3 Abdurrahman Ibnu Syahid   2211102441179*

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2023

# Project UAS: Game Block Breaker

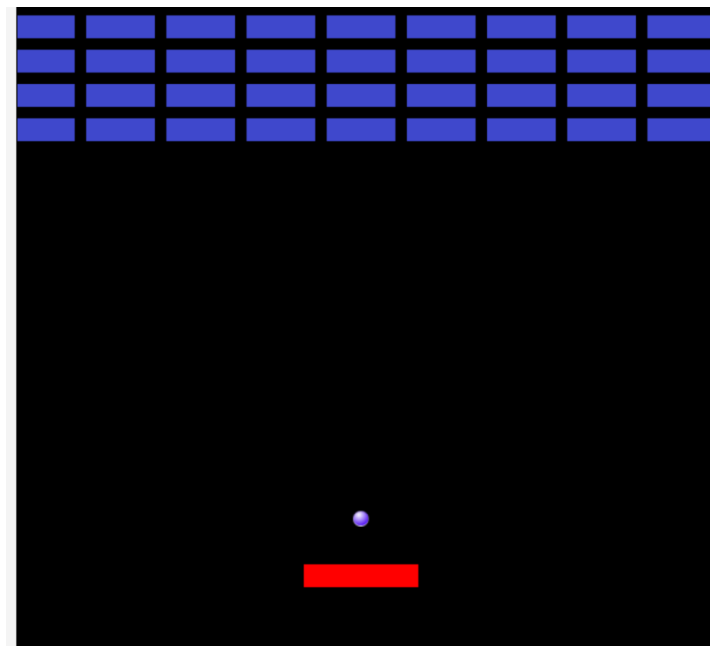
---

Pada project Ujian Akhir Semester ini kelompok kami yaitu kelompok 24 memilih untuk membuat game sederhana yang terinspirasi game jadul jaman arcade yaitu Brick Breaker. Kami membuatnya dengan software Greenfoot yang telah di pelajari di semester 3 ini dengan Bahasa kodingan menggunakan Bahasa java. Kelompok kami menamai project kami yaitu Bloke Breaker.

Berikut ini adalah beberapa skenario yang kami terapkan pada project game kami yaitu;

1. Terdapat 3 Objek yang ada dalam game ini yaitu “**Player, Block dan Ball**”.
2. Tujuan dari game ini adalah menghancurkan **Block** yang berabris diatas menggunakan **Ball**.
3. **Ball** memiliki kemampuan untuk memantul di segala tempat kecuali area di bawah **Player**.
4. **Player** adalah Objek yang dikendalikan oleh pemain game ini, **Player** dapat bergerak ke samping kanan maupun kiri.
5. **Player** memiliki fungsi sebagai penghalang agar **Ball** jatuh.
6. Pemain dapat kalah jika **Ball** jatuh ke area bawah player.

Ini adalah tampilan awal game project yang kami buat ketika game di buka;



## Penjelasan

Sudah disebutkan sebelumnya kelompok kami menggunakan Bahasa java dengan menggunakan software Greenfoot. Berikut adalah kode pada setiap subclass world dan actor yang telah kami buat di game ini;

### 1. World subclass MyWorld

```
public Player player = new Player();
public MyWorld()
{
    // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
    super(610, 600, 1);
    prepare();
}
```

- MyWorld()

Dalam method MyWord selain memasukan angka untuk mengatur ukuran dunia gamenya, di masukan juga method prepare.

```
public void prepare()
{
    getBackground().setColor(Color.BLACK);
    getBackground().fill();

    addObject(player, 300, 500);
    for(int i = 0; i < 9; i++)
        for(int j = 0; j < 4; j++)
            addObject(new Block(), 30 + 70 * i, 20 + 30 * j);
    Ball ball = new Ball(player, this);
    addObject(ball, 300, 450);
}
```

- Prepare()

Dalam methodode prepare kami memasukan kodingan untuk memunculkan **Player** dan **Ball** di dunia gamenya sesuai koordinat. Lalu kami menggunakan fungsi perulangan untuk memunculkan Bloke agar berbaris diatas sesuai koordinatnya.

### 2. Actor subclass Player

```
public Player()
{
    getImage().setColor(Color.RED);
    getImage().fillRect(0, 0, 100, 20);
}
```

- Player()

Disini kami menggunakan getImage untuk mengatur warna objek**Player** menjadi merah.

```
public void act()
{
    if(Greenfoot.isKeyDown("right"))
        setLocation(getX() + 5, getY());
    if(Greenfoot.isKeyDown("left"))
        setLocation(getX() - 5, getY());
    // Add your action code here.
}
```

- act()

Method ini digunakan untuk mengatur pergerakan dan kontrol objek player.

### 3. Actor subclass Ball

```
int dy = 5;
int dx = 0;
int count = 0;
public Player player;
public World myWorld;
public Ball(Player player, World world)
{
    myWorld = world;
    this.player = player;
    getImage().scale(getImage().getWidth()/2, getImage().getHeight()/2);
}
```

Dalam subclass Ball yang pertama di atur adalah kecepatannya pada sumbu y yaitu 5.

- Ball(Player player, World world)

Karena asset bola nya dianggap terlalu besar jadi kami mengecilkannya.

```
public void act()
{
    moveAround();
    bounce();
    bounceOffEdge();
    hitBlock();
    youLose();
    youWin();
}
```

- act()

Pada method act di dalamnya dimasukan method-method logika yang akan mencakup pergerakan, pantulan, mengenai **Block** dan reaksi ketika menabrak dinding dunia.

```
public void moveAround()
{
    setLocation(getX() + dx, getY() + dy);
}
```

- moveAround()

Methode moveAround berfungsi agar actor atau objek **Ball** bergerak kearah yang telah diatur.

```
public void bounce()
{
    if(isTouching(Player.class) && this.player.getX() > getX() + 50)
    {
        dy = -dy;
        dx-= 6;
    }
    else if(isTouching(Player.class) && this.player.getX() < getX() - 50)
    {
        dy = -dy;
        dx+= 10;
    }
    else if(isTouching(Player.class) && this.player.getX() > getX() + 40)
    {
        dy = -dy;
        dx-= 5;
    }
    else if(isTouching(Player.class) && this.player.getX() < getX() - 40)
    {
        dy = -dy;
        dx+= 5;
    }
    else if(isTouching(Player.class) && this.player.getX() > getX() + 30)
    {
        dy = -dy;
        dx-= 4;
    }
    else if(isTouching(Player.class) && this.player.getX() < getX() - 30)
    {
        dy = -dy;
        dx+= 4;
    }
    else if(isTouching(Player.class) && this.player.getX() > getX() + 20)
    {
        dy = -dy;
        dx-= 3;
    }
    else if(isTouching(Player.class) && this.player.getX() < getX() - 20)
    {
        dy = -dy;
        dx+= 3;
    }
}
```

```
else if(isTouching(Player.class) && this.player.getX() < getX() - 20)
{
    dy = -dy;
    dx+= 3;
}
else if(isTouching(Player.class) && this.player.getX() > getX() + 10)
{
    dy = -dy;
    dx-= 2;
}
else if(isTouching(Player.class) && this.player.getX() < getX() - 10)
{
    dy = -dy;
    dx+= 2;
}
else if((isTouching(Player.class) || isTouching(Block.class)))
{
    dy = -dy;
}
}
```

- Bounce()

Method ini berfungsi agar objek **Ball** dapat kemampuan agar dapat memantul.

```
public void bounceOffEdge()
{
    if(getX() >= getWorld().getWidth() -1)
    {
        dx = -dx;
    }
    if(getX() <= 0)
    {
        dx = -dx;
    }
    if(getY() <= 0)
    {
        dy = -dy;
    }
    if(getY() >= getWorld().getHeight() -1)
    {
        dy = -dy;
    }
}
```

- bounceOffEdge()

Method ini berfungsi agar objek **Ball** dapat memantul di dinding dunia selain dinding dibawah **Player**.

```

public void hitBlock()
{
    Actor block = getOneIntersectingObject(Block.class);
    if(block != null && block.getX() > getX() + 30)
    {
        getWorld().removeObject(block);
        dx = -dx;
        count++;
    }
    else if(block != null && block.getX() < getX() - 30)
    {
        getWorld().removeObject(block);
        dx = -dx;
        count++;
    }
    else if (block != null)
    {
        getWorld().removeObject(block);
        count++;
    }
}

```

- hitBlock()

Method ini memunculkan reaksi **Ball** ketika mengenai objek **Block**, yang dimana akan menghapus keadaan **Block** menjadi tidak ada.

```

public void youLose()
{
    if(getY() >= getWorld().getHeight() - 1)
    {
        getWorld().showText("YOU LOSS!!!!", getWorld().getWidth()/2, getWorld().getHeight()/2);
        Greenfoot.stop();
    }
}

public void youWin()
{
    if(count == 32)
    {
        getWorld().showText("YOU WIN!!!!", getWorld().getWidth()/2, getWorld().getHeight()/2);
        Greenfoot.stop();
    }
}

```

- youLose()

Ketika objek **Ball** menyentuh dinding dibawah **Player**, game akan di hentikan dan akan muncul text "YOU LOSS!!!!".

- youWin()

Ketika Objek **Block** dapat dihancurkan keseluruhannya, game akan berhenti dan akan muncul text "YOU WIN!!!!".

#### 4. Actor subclass Block

```

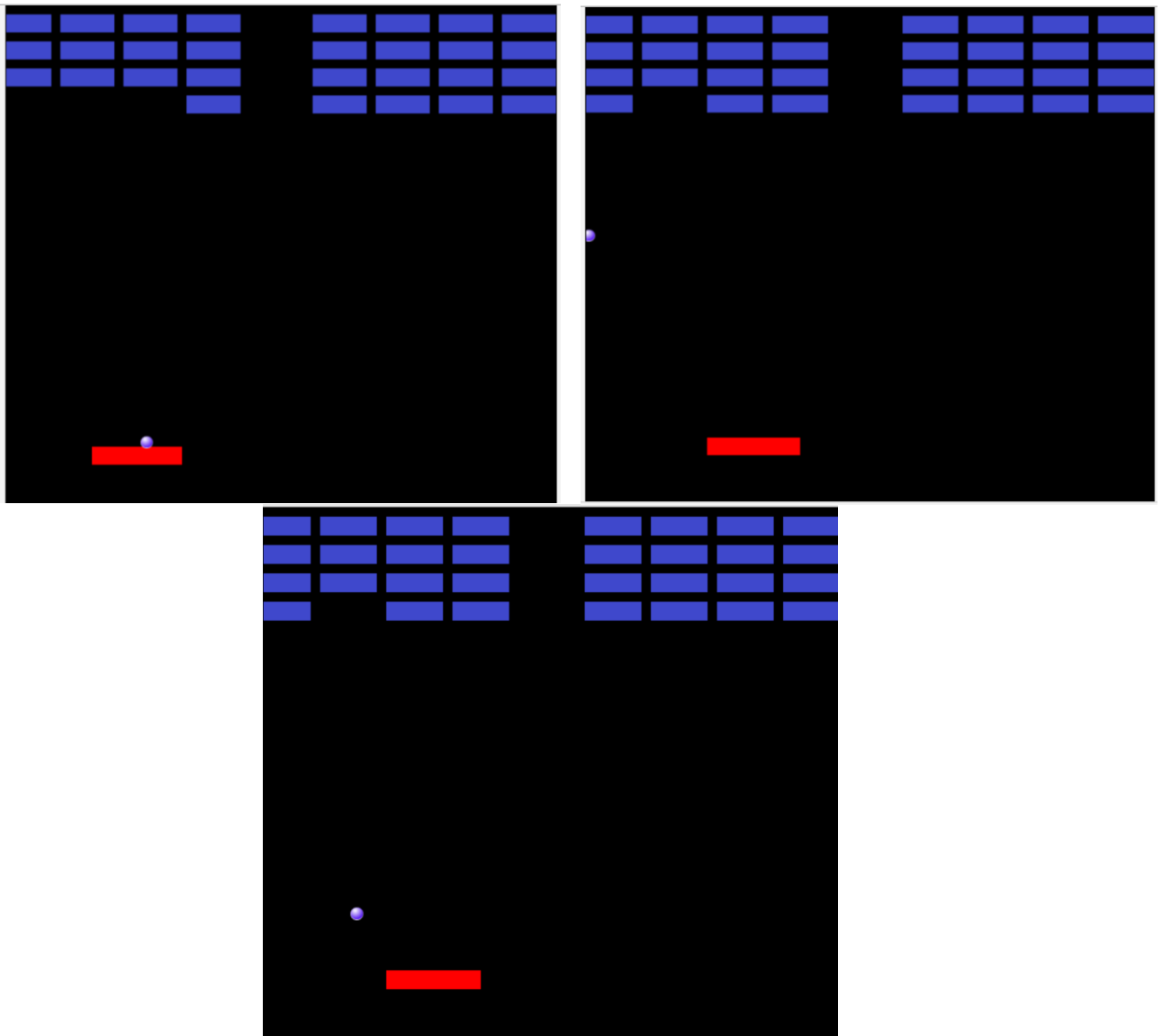
public Block()
{
    getImage().fillRect(60,0, 50, 20);
}

```

- Block()

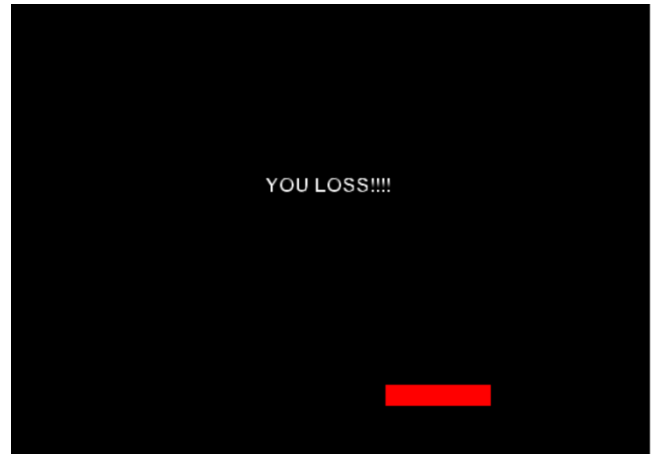
Menggunakan getImage untuk membantu dan mewarnai subclass **Block**.

## Hasil



Terlihat pada diatas bahwa objek **Ball** jika mengenai **Block** maka akan langsung menghapus **Block** dari dunianya. **Ball** juga dapat memantul didinding dan objek lain.





Game akan berhenti dan akan muncul tulisan “YOU WIN!!!” jika pemain dapat mempertahankan **Ball** memantul menggunakan **Player** sehingga menghabiskan **Block** yang ada.

Sebaliknya game akan berhenti dengan tulisan “YOU LOSE!!!” jika pemain gagal memantulkan Kembali **Ball** ke atas dan jatuh ke bawah **Player**.

Demikian laporan project kami yaitu game Bloke Breaker yang dibuat menggunakan software greenfoot. Laporan ini dibuat untuk memenuhi tugas akhir semester ganjil pada mata kuliah pemrograman berorientasikan objek. Mohon maaf jika ada kekurangan atau kekeliruan dalam pembuatan project ini. terima kasih.