

UNIVERSITAS MUHAMMADIYAH
KALIMANTAN TIMMUR
(LAPORAN PEMOGRAMAN BERBASIS OBJECT)



KELOMPOK 6:

1. SENDHY RHYSA
2. DITO HARDIANSYAH
3. NOVARIYANI NUR R

PENDAHULUAN

Pada Laporan Kali ini Kami kelompok 6 membuat semua game yang bernama **DINO MASTER**, yang dimana dalam game ini menggunakan kecerdasan buatan.

LAPORAN KONTEN

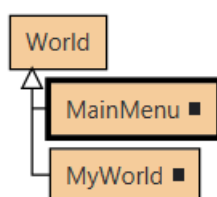
- Tampilan Awal



Keterangan:

Pada tampilan awal ini kami membuat desain dengan aplikasi canva dengan menyesuaikan background pada game.

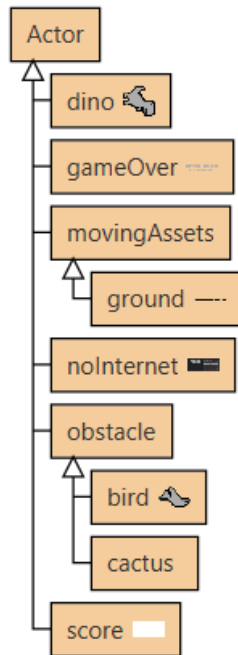
- Tampilan World & Background



Keterangan:

Pada tampilan world ini terdapat 2 subclass yang terdiri dari MainMenu dan MyWorld.

- Tampilan Actor



Keterangan:

Pada tampilan actor terdapat banyak subclass yang terdiri dari dino,gameOver, movingAssets, ground, noInternet, obstacle, bird, cactus, score dimana masing-masing subclass memiliki fungsinya tersendiri.

- Fungsi MyWorld & MainMenu

```

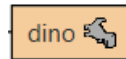
/**
 * Ini adalah Mainmenu class.
 * @author (Muhammad Ammar Nabil)
 */
public class MainMenu extends World
{
    public MainMenu()
    {
        super(900, 400, 1);
        setBackground("background.jpg");
        prepare();
    }

    private void prepare()
    {
        dino dino = new dino();
        addObject(dino,70,279);
        addObject(new noInternet(), 470, 220);
        dino.alive = false;
        setPaintOrder(noInternet.class, gameOver.class, dino.class, score.class, bird.class, cactus.class,
            ground.class);
        if (Greenfoot.mouseClicked(dino)) {
            dino.alive = true;
            noInternet.mainmenu = false;
            Greenfoot.setWorld(new MyWorld());
        }
    }
}
  
```

Keterangan:

MyWorld berfungsi sebagai background/latar belakang yang memiliki fungsi seperti gambar diatas dimulai dari latar,ukuran latar,dan tampilan object lainnya.

- Fungsi Actor Dino, Bird, cactus



```

Class Edit Tools Options
dino x
Compile Undo Cut Copy Paste Find... Close Source Code

/**
 * It's Dino class
 *
 * @Muhammad Ammar Nabil
 * @version beta
 */
public class dino extends Actor
{
    private int counter = 0;
    private int counterMin = 0;
    private int counterJump = 0;
    private int jumpSpeed;
    private boolean walk = true;
    private boolean down;
    private boolean jumpPress = false;
    private boolean onGround = true;
    private boolean pressed;

    static public boolean alive = true;

    public dino()
    {
        setImage("dino-jump.png");
        setLocation(70, 279);
        setRotation(-90);
    }

    public void act()
    {
        if(alive){
            if(onGround){
                counter++;
                counterJump = 0;
                counterMin = 0;
                if(counter>=5) {
                    setLocation(70, 279);
                    counter = 0;
                    running();
                }

                if (Greenfoot.isKeyDown("down")){
                    down = true;
                    crouch();
                }else {
                    down = false;
                    if (jumpState() && !jumpPress) {
                        setImage("dino-jump.png");
                        setLocation(70, 274);
                        move(5);
                        jumpSpeed = 10;
                        onGround = false;
                        jumpPress = true;
                    } else {
                        if(!jumpState()){
                            jumpPress = false;
                        } else {
                            running();
                        }
                    }
                }
            }
            else {
                counterMin++;
                if((jumpPress && jumpState() && counterJump<=12) ||
                    counterMin < 8){
                    counterJump++;
                    move(10);
                } else {
                    counterJump = 20;
                    jumpSpeed--;
                    move(jumpSpeed);
                    if(getY() >= 274){
                        setLocation(70, 279);
                        onGround = true;
                    }
                }
            }
        }

        if (isTouching(obstacle.class)){
            alive = false;
        }

        }else{
            getWorld().addObject(new gameOver(), 450, 200);
        }
    }
}

```

```

private void running(){
    if(walk) {
        setImage("dino-0.png");
        walk = false;
    }else {
        setImage("dino-1.png");
        walk = true;
    }
}

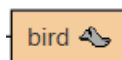
private void crouch(){
    setLocation(86,296);
    if(walk) {
        setImage("dino-crouch-0.png");
    }else {
        setImage("dino-crouch-1.png");
    }
}
}

private boolean jumpState() {
    if (Greenfoot.isKeyDown("up") || Greenfoot.isKeyDown("space")){
        return true;
    }
    if(pressed){return true;}
    return false;
}
}

```

Keterangan:

Dari hasil input diatas kita dapat melihat dimana actor utama yaitu dino yang memiliki peran utama, sebagai peran yang banyak berkontribusi pada game tersebut.



```

Class Edit Tools Options
bird X
Compile Undo Cut Copy Paste Find... Close Source Code
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

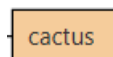
public class bird extends obstacle
{
    private int counter = 0;
    private boolean fly = true;
    private boolean bird = true;

    public bird(){
        setImage("bird-0.png");
    }

    public void act()
    {
        if(dino.alive){
            counter++;
            if(bird){
                fly();
                move(-6 - MyWorld.difficulty);
                if(isAtEdge()){
                    getWorld().removeObject(this);
                }
            }
        }
    }

    private void fly(){
        if(counter >= 5){
            counter = 0;
            if(fly){
                setLocation(getX(), getY()+12);
                setImage("bird-1.png");
                fly = false;
            } else {
                setLocation(getX(), getY()-12);
                setImage("bird-0.png");
                fly = true;
            }
        }
    }
}

```



```

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

public class cactus extends obstacle
{
    private int counter = 0;
    private boolean cactus = true;

    public cactus(){
        setImage("cactus-small-1.png");
    }

    public void act()
    {
        if(dino.alive){
            counter++;
            if(cactus){
                randomization();
                cactus = false;
            }
            move(-6 - MyWorld.difficulty);
            if(isAtEdge()){
                cactus = true;
                getWorld().removeObject(this);
            }
        }
    }

    private void randomization(){
        if (Greenfoot.getRandomNumber(4) == 0){
            setImage("cactus-big-1.png");
        } else {
            setImage("cactus-small-2.png");
        }
    }
}

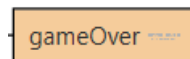
```

Class compiled - no syntax errors

Keterangan:

Pada hasil Input diatas kita dapat melihat fungsi dari actor bird & cactus yang dimana berfungsi sebagai musuh dari actor dino sebagai dari peran utama yang dimana fungsi actor bird & cactus sebagai elemeator.

- Fungsi Actor gameOver, noInternet, score



```

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

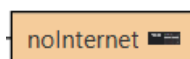
public class gameOver extends Actor
{
    public void act()
    {
        dino dino = new dino();
        if (Greenfoot.isKeyDown("space") || Greenfoot.isKeyDown("up")){
            dino.alive = true;
            Greenfoot.setWorld(new MyWorld());
        }
    }

    public gameOver(){
        setLocation(450, 200);
        setImage("game-over.png");
    }
}

```

Keterangan:

Pada tampilan gameOver ini tampilan inputny berfungsi sebagi background selesai dari permainan tersebut yang dimana akhir dari sebuah permainan.





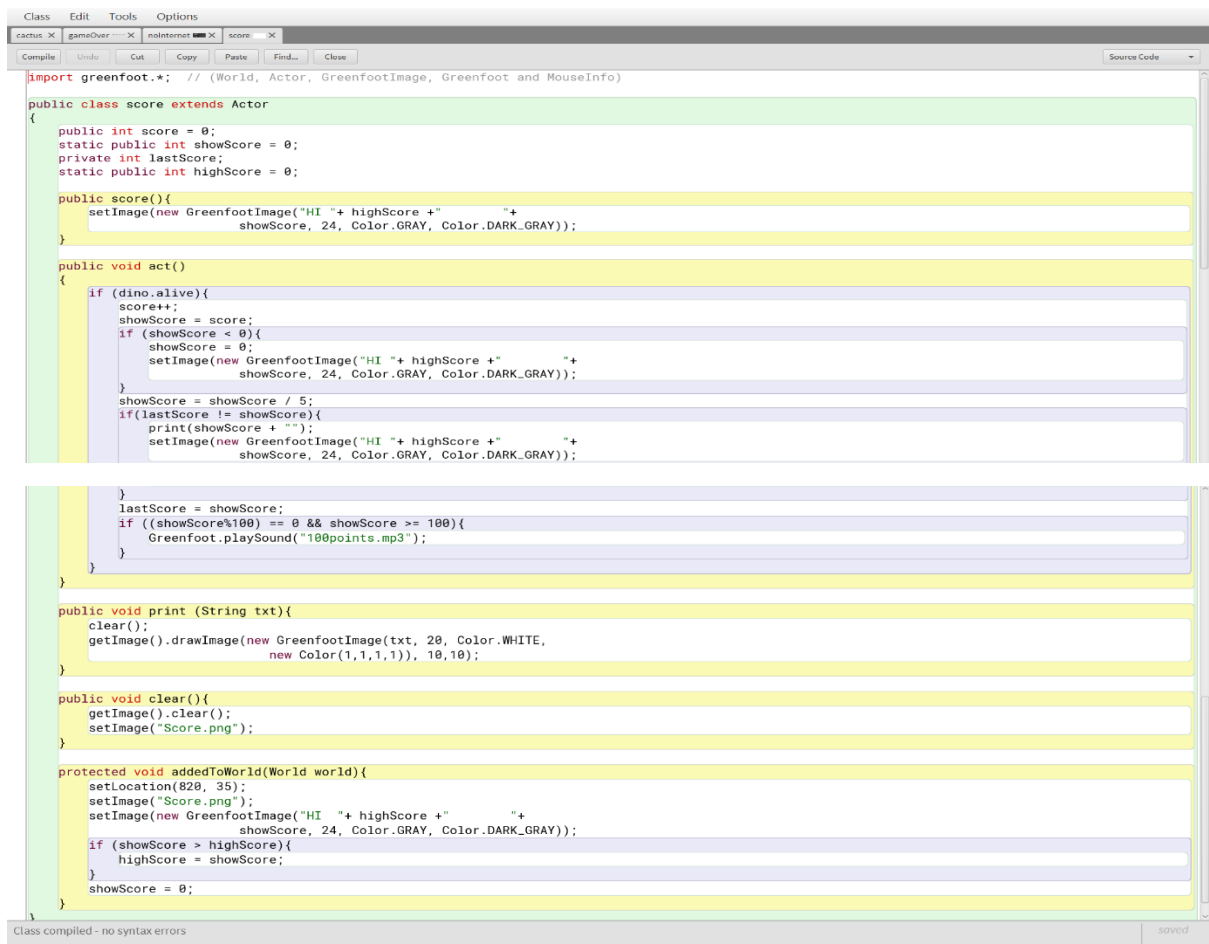
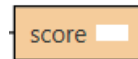
```
public class noInternet extends Actor
{
    static public boolean mainmenu = true;

    public noInternet(){
        setImage("no-internet.png");
    }

    public void act()
    {
        if(mainmenu){
            setImage("no-internet.png");
            dino dino = new dino();
            if(Greenfoot.isKeyDown("space") || Greenfoot.isKeyDown("up")){
                dino.alive = true;
                Greenfoot.setWorld(new MyWorld());
            }
        }
    }
}
```

Keterangan:

Pada tampilan input noInternet ini akan berada diawal permainan yang dimana hanya berfungsi sebagai tampilan awal mulai permainan.



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

public class score extends Actor
{
    public int score = 0;
    static public int showScore = 0;
    private int lastScore;
    static public int highScore = 0;

    public score(){
        setImage(new GreenfootImage("HI " + highScore + " " +
            showScore, 24, Color.GRAY, Color.DARK_GRAY));
    }

    public void act()
    {
        if (dino.alive){
            score++;
            showScore = score;
            if (showScore < 0){
                showScore = 0;
                setImage(new GreenfootImage("HI " + highScore + " " +
                    showScore, 24, Color.GRAY, Color.DARK_GRAY));
            }
            showScore = showScore / 5;
            if(lastScore != showScore){
                print(showScore + " ");
                setImage(new GreenfootImage("HI " + highScore + " " +
                    showScore, 24, Color.GRAY, Color.DARK_GRAY));
            }
        }
        lastScore = showScore;
        if ((showScore%100) == 0 && showScore >= 100){
            Greenfoot.playSound("100points.mp3");
        }
    }

    public void print (String txt){
        clear();
        getImage().drawImage(new GreenfootImage(txt, 20, Color.WHITE,
            new Color(1,1,1,1)), 10,10);
    }

    public void clear(){
        getImage().clear();
        setImage("Score.png");
    }

    protected void addedToWorld(World world){
        setLocation(820, 35);
        setImage("Score.png");
        setImage(new GreenfootImage("HI " + highScore + " " +
            showScore, 24, Color.GRAY, Color.DARK_GRAY));
        if (showScore > highScore){
            highScore = showScore;
        }
        showScore = 0;
    }
}
```

Keterangan:

Pada tampilan input score ini berfungsi sebagai pengatur score pada game yang dimana setiap actor utama (Dino) bertahan pada waktu yang lama maka score pun akan ikut berjalan sesuai dengan durasi waktu yang dimainkan.

Kesimpulan:

Dari laporan diatas kami dapat menyimpulkan bahwa game yang kami buat ini adalah game yang membutuhkan ketepatan, kelincahan, dan kecermatan untuk bisa mencari score sebanyak-banyaknya dengan melewati rintangan-rintangan yang ada.