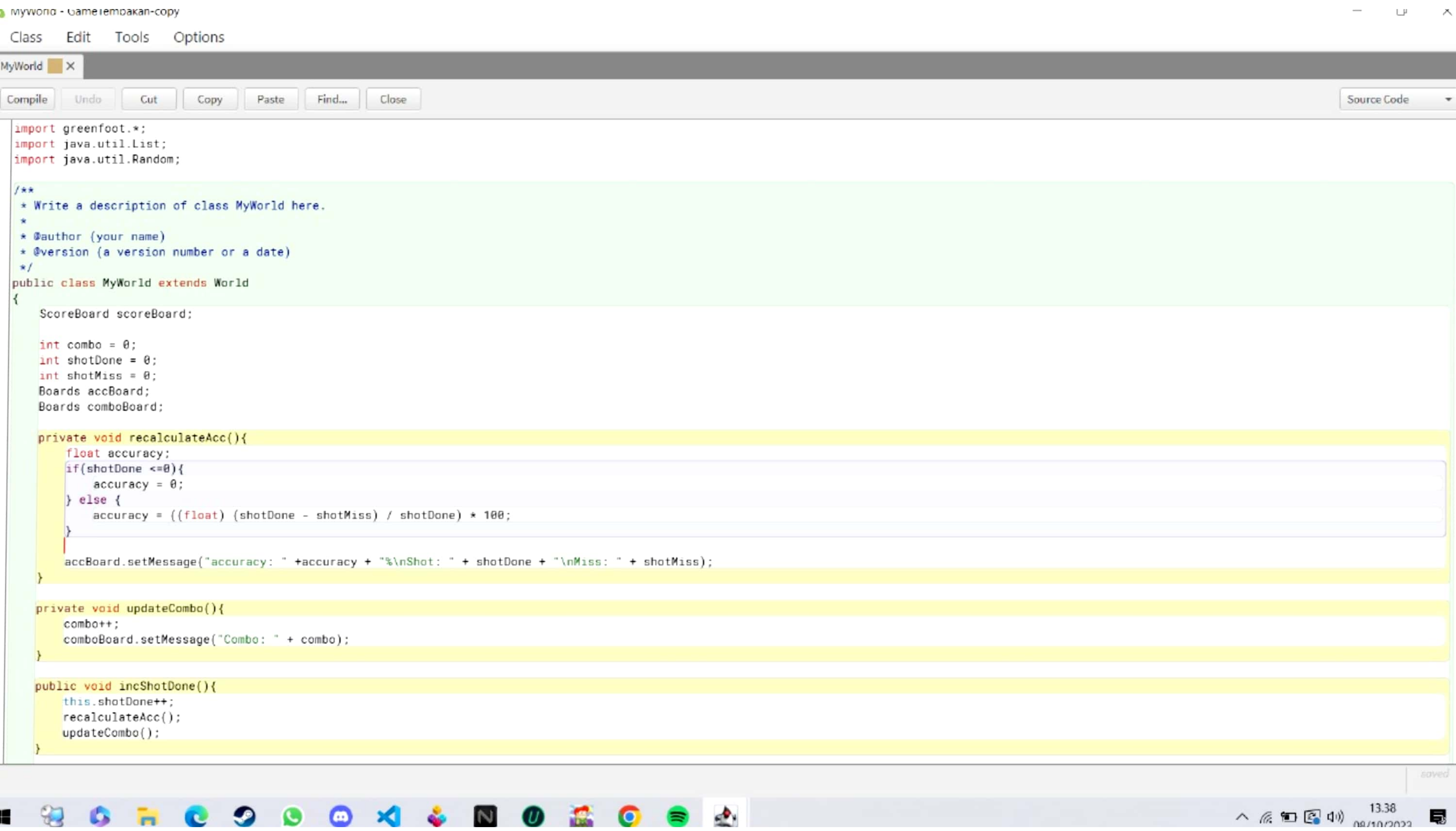
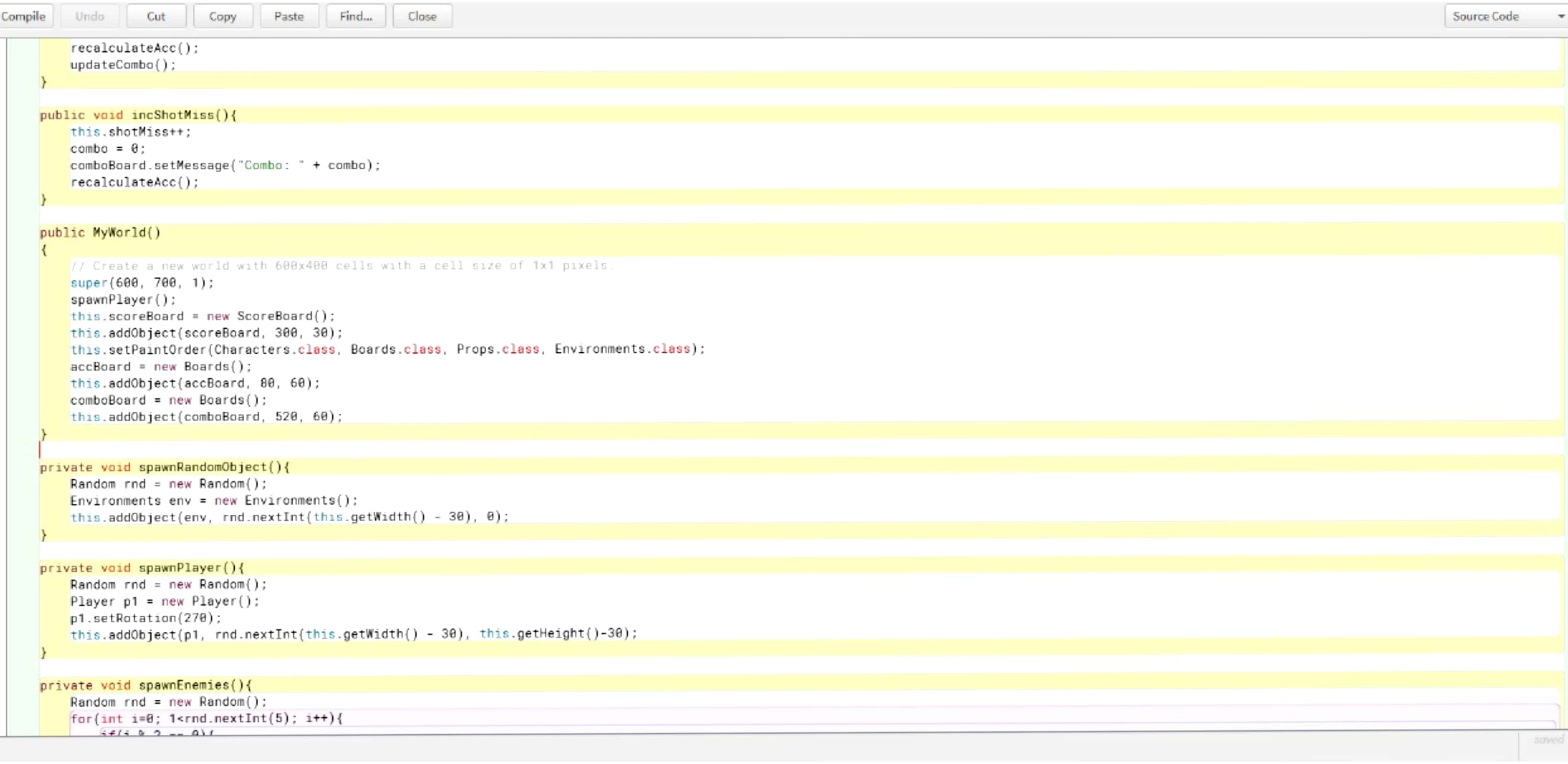
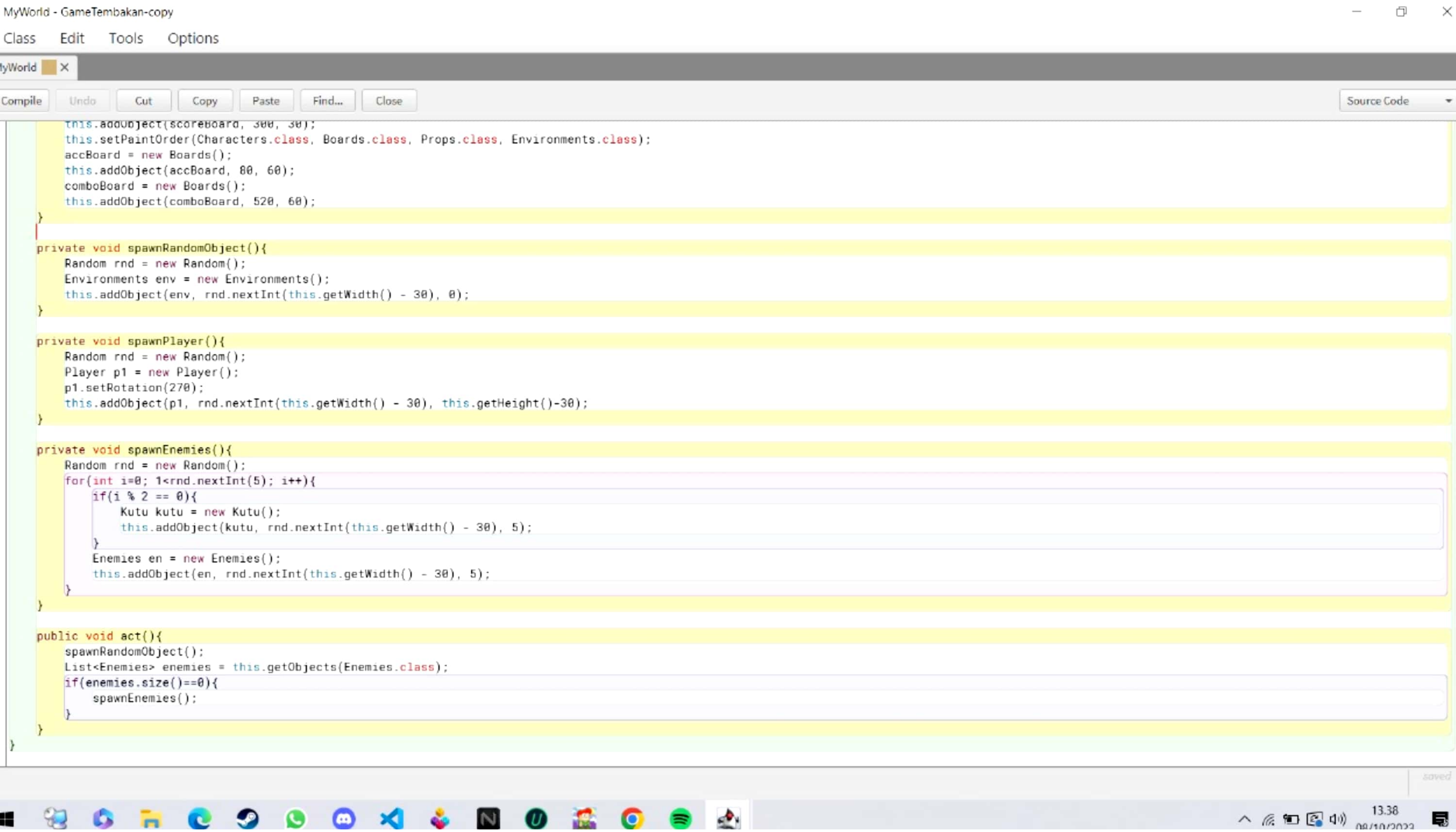


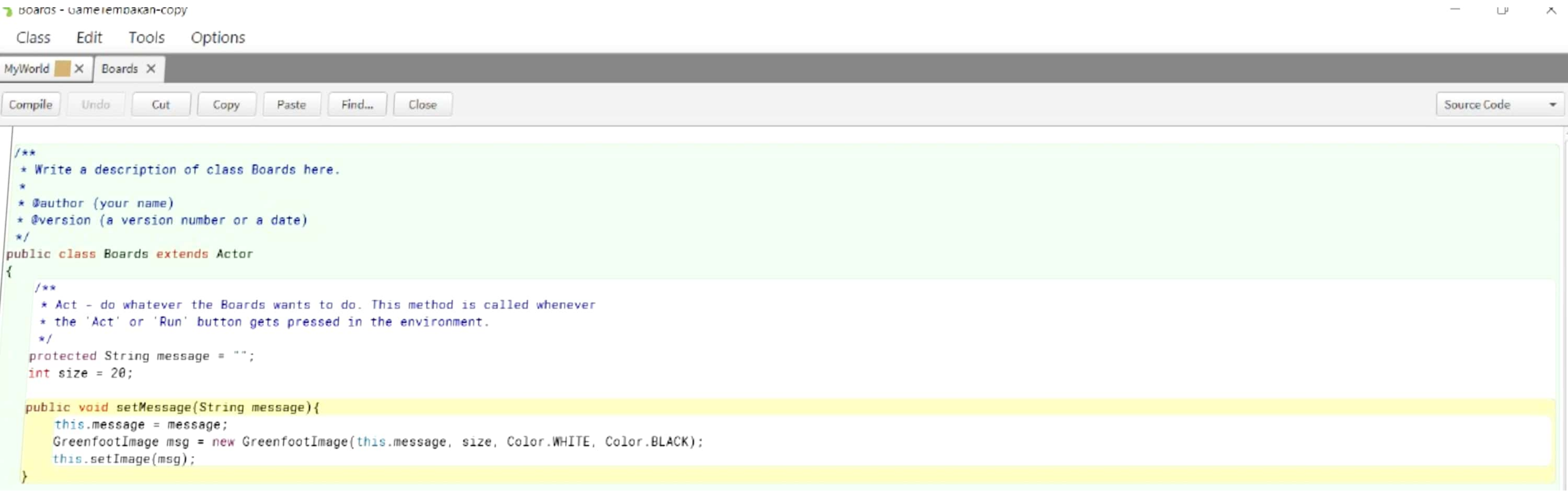
```java
import greenfoot.*;
import java.util.List;
import java.util.Random;

/**
 * Write a description of class MyWorld here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MyWorld extends World
{
    ScoreBoard scoreBoard;

    int combo = 0;
    int shotDone = 0;
    int shotMiss = 0;
    Boards accBoard;
    Boards comboBoard;

    private void recalculateAcc(){
        float accuracy;
        if(shotDone <=0){
            accuracy = 0;
        } else {
            accuracy = ((float) (shotDone - shotMiss) / shotDone) * 100;
        }

        accBoard.setMessage("accuracy: " +accuracy + "%\nShot: " + shotDone + "\nMiss: " + shotMiss);
    }

    private void updateCombo(){
        combo++;
        comboBoard.setMessage("Combo: " + combo);
    }

    public void incShotDone(){
        this.shotDone++;
        recalculateAcc();
        updateCombo();
    }
```



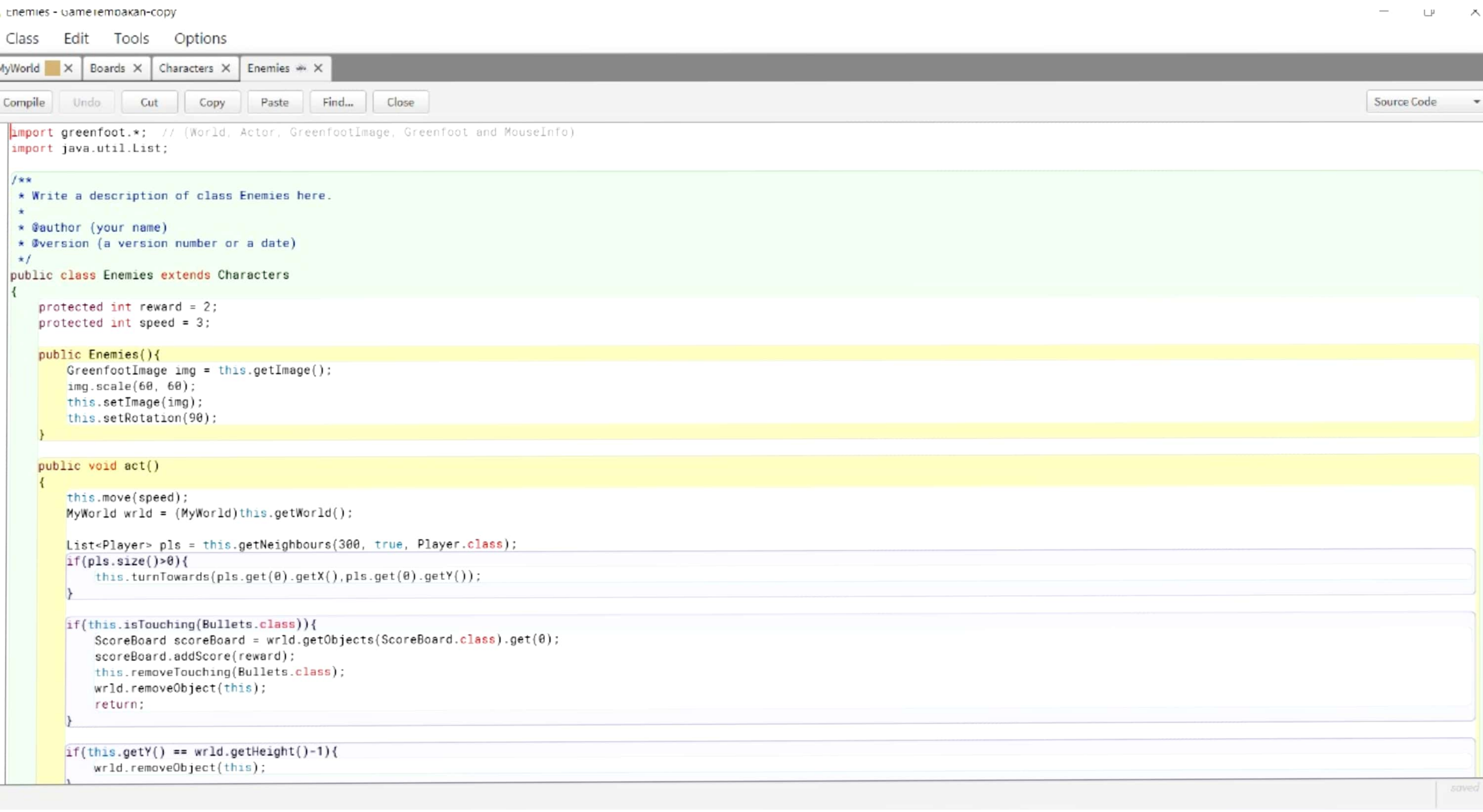
```
        recalculateAcc();
        updateCombo();
    }

    public void incShotMiss(){
        this.shotMiss++;
        combo = 0;
        comboBoard.setMessage("Combo: " + combo);
        recalculateAcc();
    }

    public MyWorld()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(600, 700, 1);
        spawnPlayer();
        this.scoreBoard = new ScoreBoard();
        this.addObject(scoreBoard, 300, 30);
        this.setPaintOrder(Characters.class, Boards.class, Props.class, Environments.class);
        accBoard = new Boards();
        this.addObject(accBoard, 80, 60);
        comboBoard = new Boards();
        this.addObject(comboBoard, 520, 60);
    }

    private void spawnRandomObject(){
        Random rnd = new Random();
        Environments env = new Environments();
        this.addObject(env, rnd.nextInt(this.getWidth() - 30), 0);
    }

    private void spawnPlayer(){
        Random rnd = new Random();
        Player p1 = new Player();
        p1.setRotation(270);
        this.addObject(p1, rnd.nextInt(this.getWidth() - 30), this.getHeight()-30);
    }

    private void spawnEnemies(){
        Random rnd = new Random();
        for(int i=0; 1<rnd.nextInt(5); i++){
```



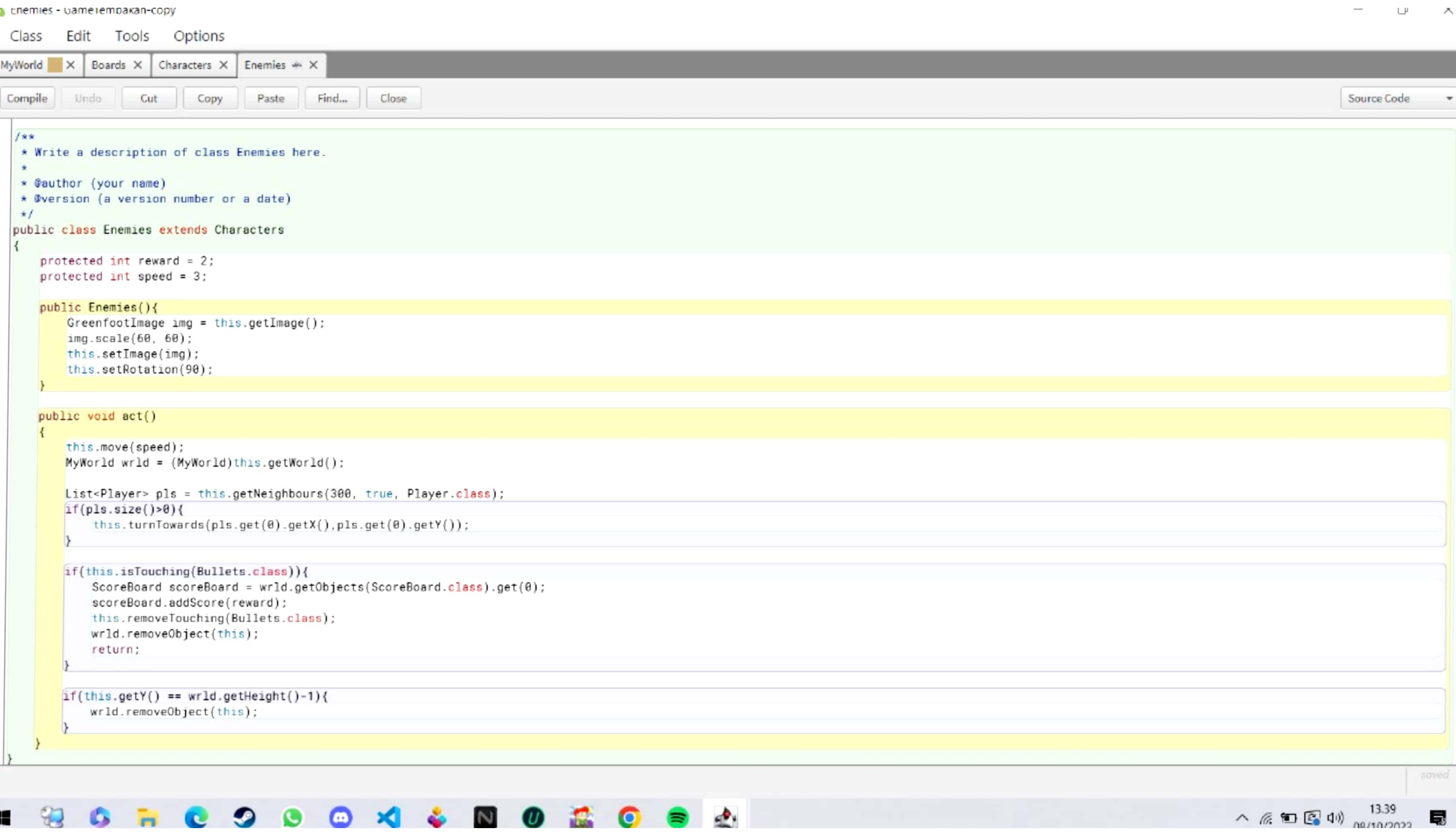
```
        this.addObject(scoreBoard, 300, 30);
        this.setPaintOrder(Characters.class, Boards.class, Props.class, Environments.class);
        accBoard = new Boards();
        this.addObject(accBoard, 80, 60);
        comboBoard = new Boards();
        this.addObject(comboBoard, 520, 60);
    }

    private void spawnRandomObject(){
        Random rnd = new Random();
        Environments env = new Environments();
        this.addObject(env, rnd.nextInt(this.getWidth() - 30), 0);
    }

    private void spawnPlayer(){
        Random rnd = new Random();
        Player p1 = new Player();
        p1.setRotation(270);
        this.addObject(p1, rnd.nextInt(this.getWidth() - 30), this.getHeight()-30);
    }

    private void spawnEnemies(){
        Random rnd = new Random();
        for(int i=0; 1<rnd.nextInt(5); i++){
            if(i % 2 == 0){
                Kutu kutu = new Kutu();
                this.addObject(kutu, rnd.nextInt(this.getWidth() - 30), 5);
            }
            Enemies en = new Enemies();
            this.addObject(en, rnd.nextInt(this.getWidth() - 30), 5);
        }
    }

    public void act(){
        spawnRandomObject();
        List<Enemies> enemies = this.getObjects(Enemies.class);
        if(enemies.size()==0){
            spawnEnemies();
        }
    }
}
```



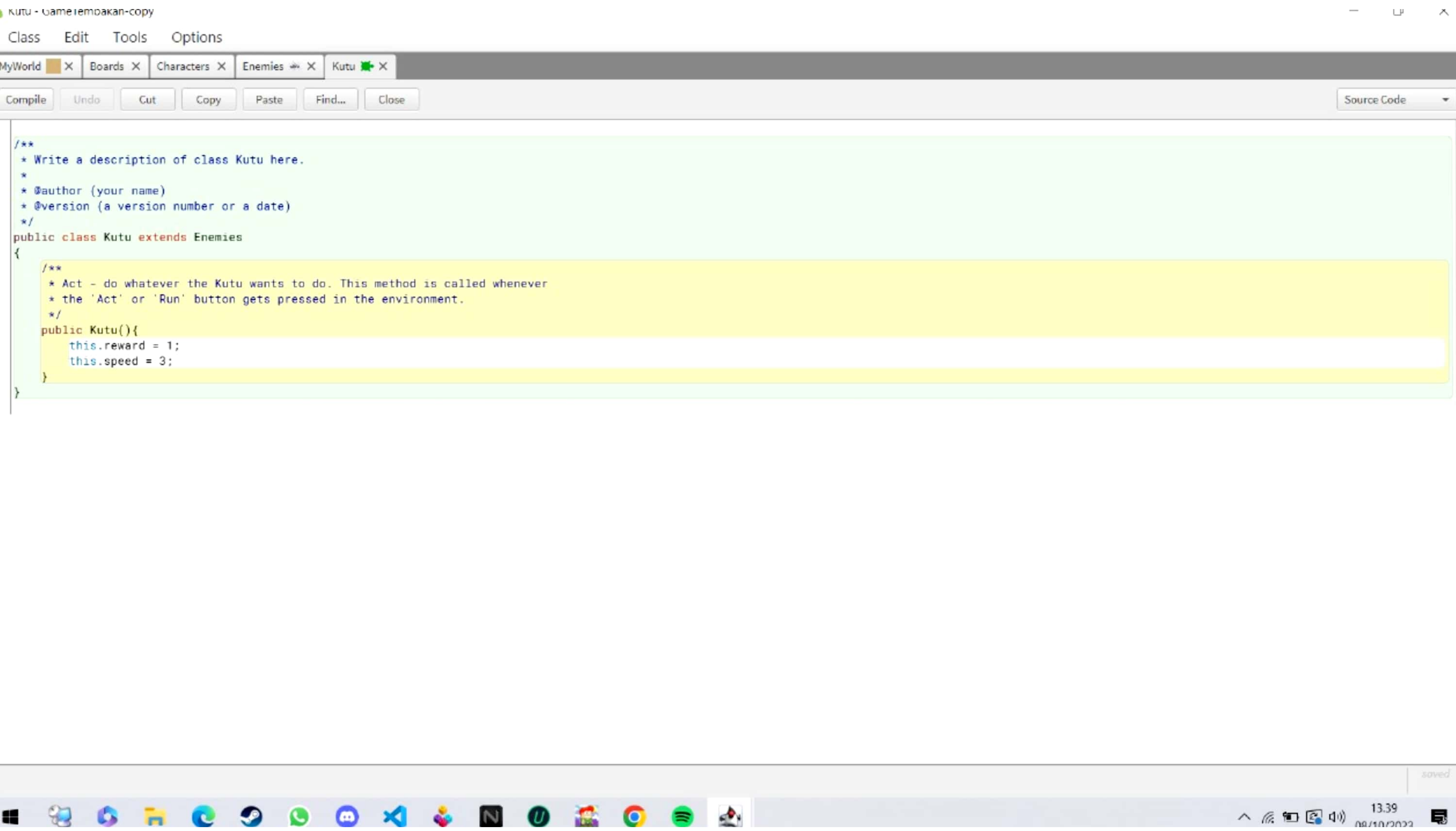
```java
/**
 * Write a description of class Boards here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Boards extends Actor
{
    /**
     * Act - do whatever the Boards wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    protected String message = "";
    int size = 20;

    public void setMessage(String message){
        this.message = message;
        GreenfootImage msg = new GreenfootImage(this.message, size, Color.WHITE, Color.BLACK);
        this.setImage(msg);
    }
```



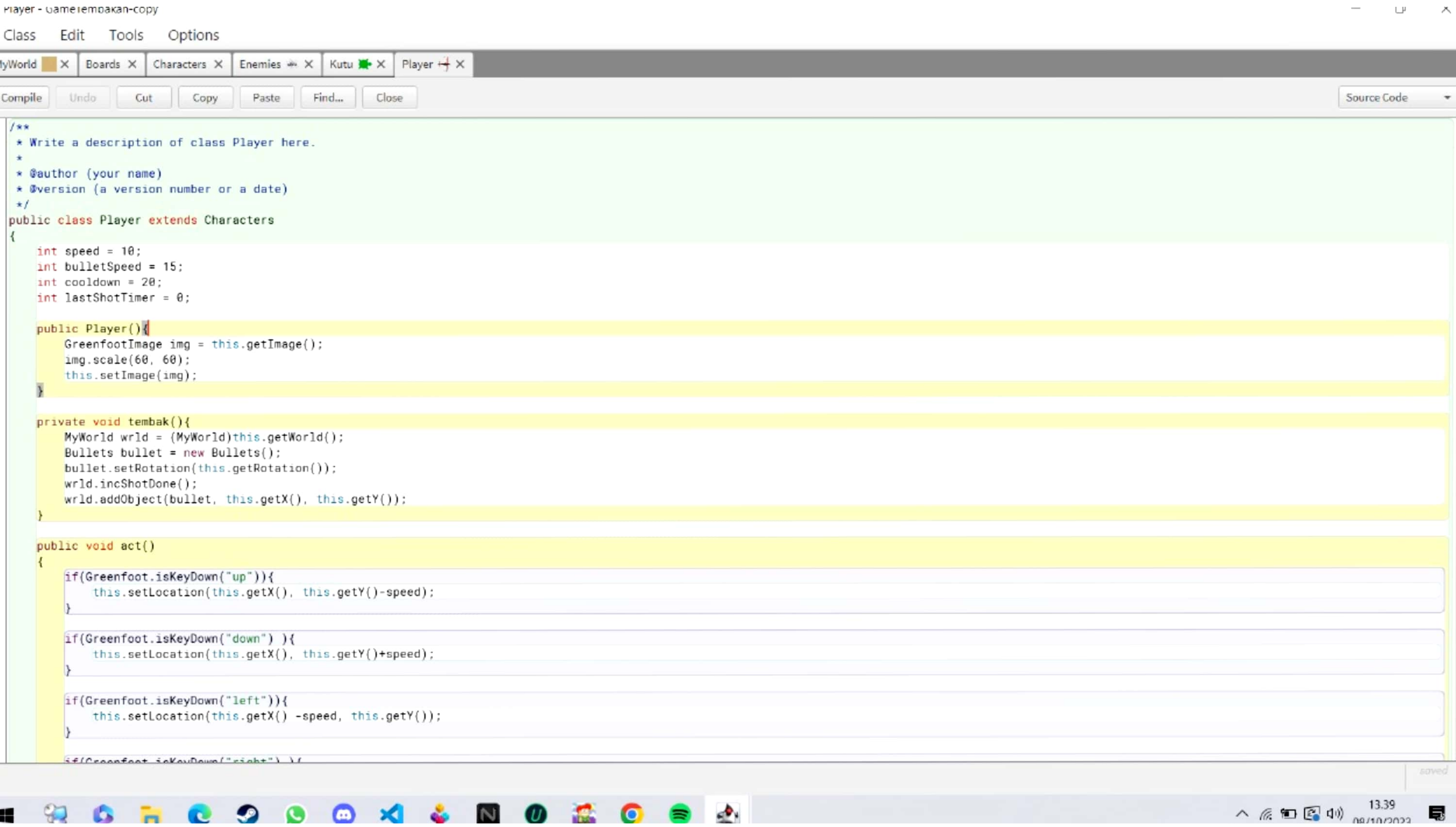
```java
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
import java.util.List;

/**
 * Write a description of class Enemies here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Enemies extends Characters
{
    protected int reward = 2;
    protected int speed = 3;

    public Enemies(){
        GreenfootImage img = this.getImage();
        img.scale(60, 60);
        this.setImage(img);
        this.setRotation(90);
    }

    public void act()
    {
        this.move(speed);
        MyWorld wrld = (MyWorld)this.getWorld();

        List<Player> pls = this.getNeighbours(300, true, Player.class);
        if(pls.size()>0){
            this.turnTowards(pls.get(0).getX(),pls.get(0).getY());
        }

        if(this.isTouching(Bullets.class)){
            ScoreBoard scoreBoard = wrld.getObjects(ScoreBoard.class).get(0);
            scoreBoard.addScore(reward);
            this.removeTouching(Bullets.class);
            wrld.removeObject(this);
            return;
        }

        if(this.getY() == wrld.getHeight()-1){
            wrld.removeObject(this);
```



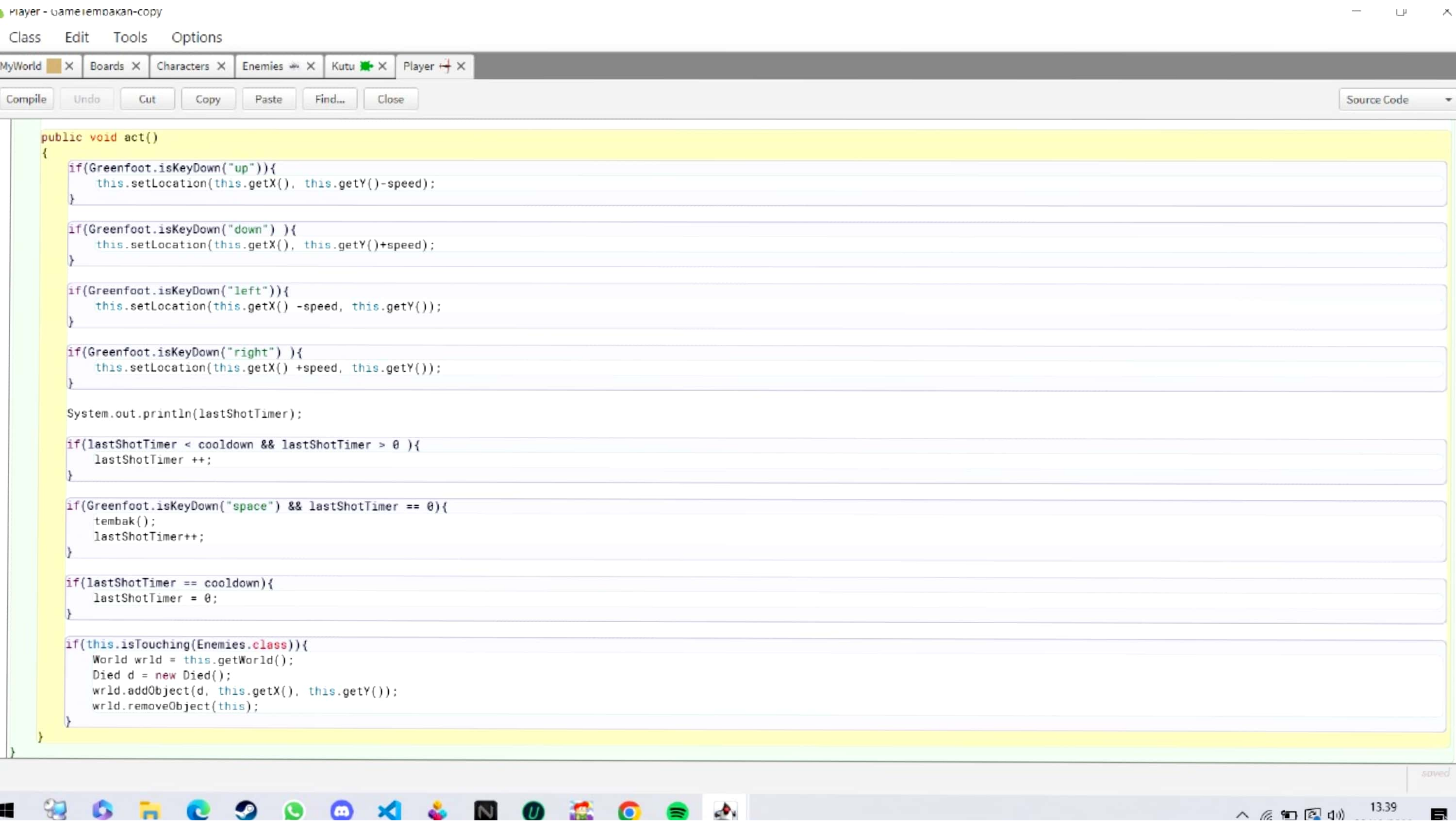
```java
/**
 * Write a description of class Enemies here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Enemies extends Characters
{
    protected int reward = 2;
    protected int speed = 3;

    public Enemies(){
        GreenfootImage img = this.getImage();
        img.scale(60, 60);
        this.setImage(img);
        this.setRotation(90);
    }

    public void act()
    {
        this.move(speed);
        MyWorld wrld = (MyWorld)this.getWorld();

        List<Player> pls = this.getNeighbours(300, true, Player.class);
        if(pls.size()>0){
            this.turnTowards(pls.get(0).getX(),pls.get(0).getY());
        }

        if(this.isTouching(Bullets.class)){
            ScoreBoard scoreBoard = wrld.getObjects(ScoreBoard.class).get(0);
            scoreBoard.addScore(reward);
            this.removeTouching(Bullets.class);
            wrld.removeObject(this);
            return;
        }

        if(this.getY() == wrld.getHeight()-1){
            wrld.removeObject(this);
        }
    }
}
```



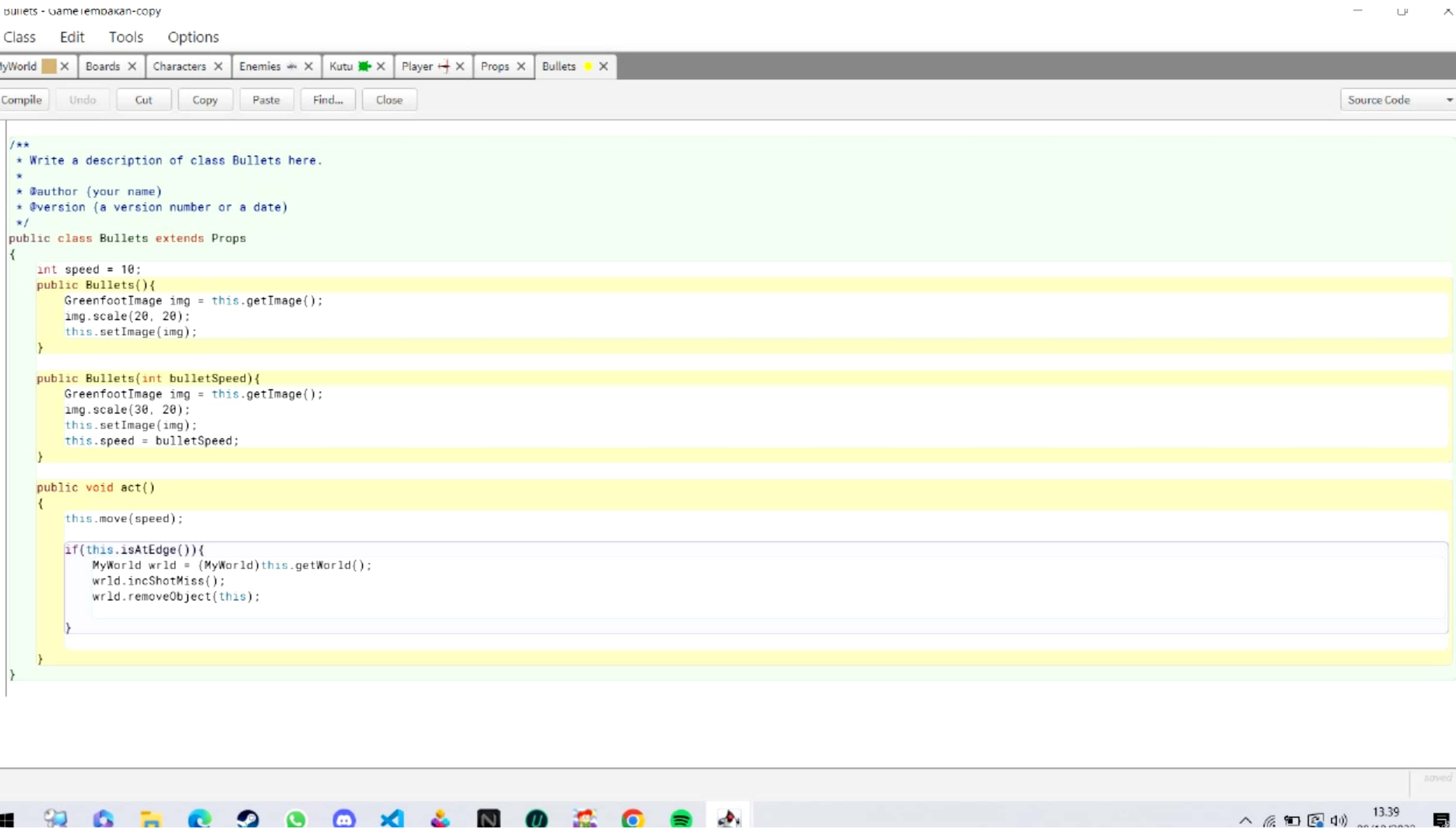
```java
/**
 * Write a description of class Kutu here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Kutu extends Enemies
{
    /**
     * Act - do whatever the Kutu wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public Kutu(){
        this.reward = 1;
        this.speed = 3;
    }
}
```



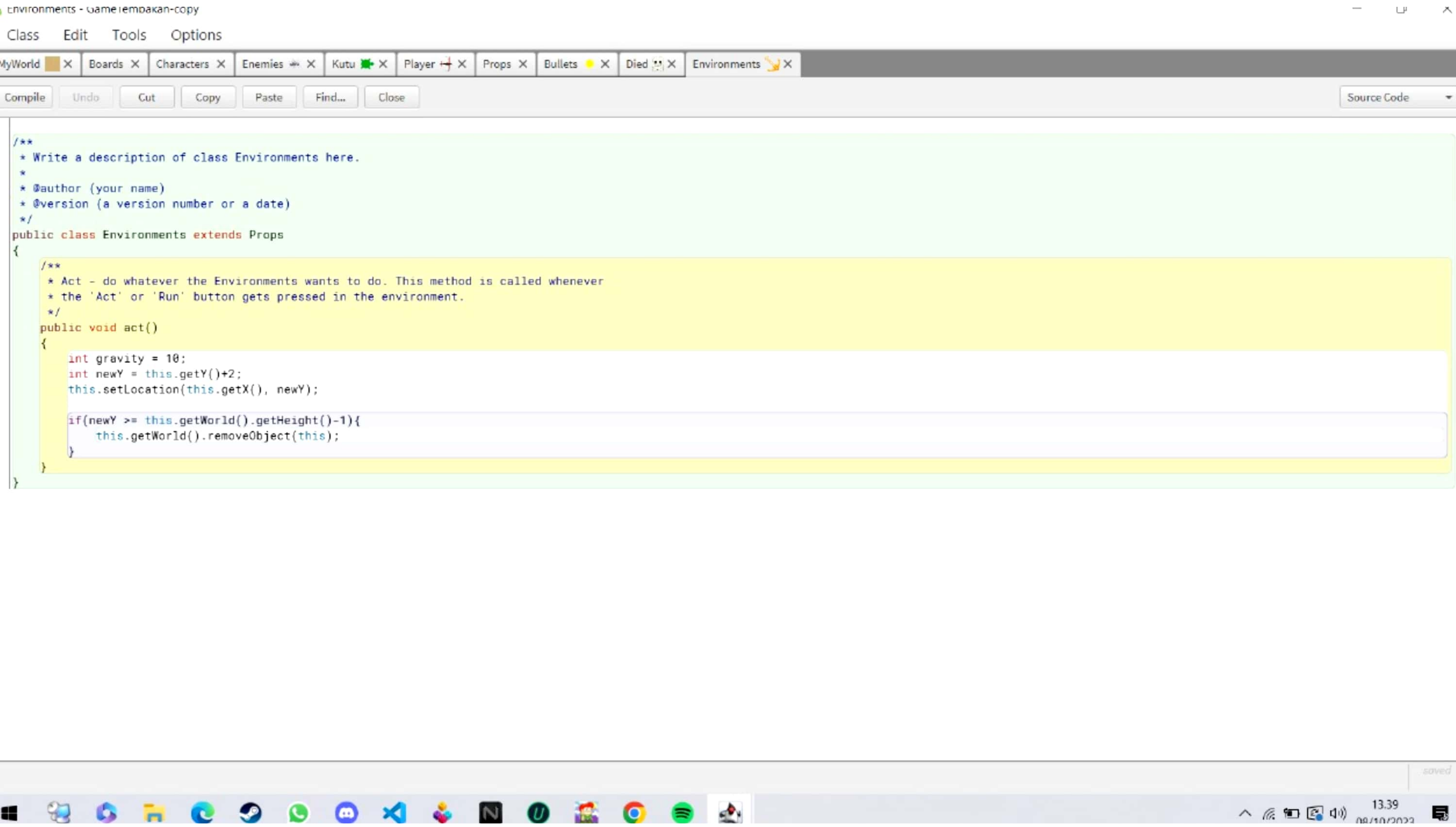
```java
/**
 * Write a description of class Player here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Player extends Characters
{
    int speed = 10;
    int bulletSpeed = 15;
    int cooldown = 20;
    int lastShotTimer = 0;

    public Player(){
        GreenfootImage img = this.getImage();
        img.scale(60, 60);
        this.setImage(img);
    }

    private void tembak(){
        MyWorld wrld = (MyWorld)this.getWorld();
        Bullets bullet = new Bullets();
        bullet.setRotation(this.getRotation());
        wrld.incShotDone();
        wrld.addObject(bullet, this.getX(), this.getY());
    }

    public void act()
    {
        if(Greenfoot.isKeyDown("up")){
            this.setLocation(this.getX(), this.getY()-speed);
        }

        if(Greenfoot.isKeyDown("down") ){
            this.setLocation(this.getX(), this.getY()+speed);
        }

        if(Greenfoot.isKeyDown("left")){
            this.setLocation(this.getX() -speed, this.getY());
        }

        if(Greenfoot.isKeyDown("right") ){
```



```
    public void act()
    {
        if(Greenfoot.isKeyDown("up")){
            this.setLocation(this.getX(), this.getY()-speed);
        }

        if(Greenfoot.isKeyDown("down") ){
            this.setLocation(this.getX(), this.getY()+speed);
        }

        if(Greenfoot.isKeyDown("left")){
            this.setLocation(this.getX() -speed, this.getY());
        }

        if(Greenfoot.isKeyDown("right") ){
            this.setLocation(this.getX() +speed, this.getY());
        }

        System.out.println(lastShotTimer);

        if(lastShotTimer < cooldown && lastShotTimer > 0 ){
            lastShotTimer ++;
        }

        if(Greenfoot.isKeyDown("space") && lastShotTimer == 0){
            tembak();
            lastShotTimer++;
        }

        if(lastShotTimer == cooldown){
            lastShotTimer = 0;
        }

        if(this.isTouching(Enemies.class)){
            World wrld = this.getWorld();
            Died d = new Died();
            wrld.addObject(d, this.getX(), this.getY());
            wrld.removeObject(this);
        }
    }
}
```



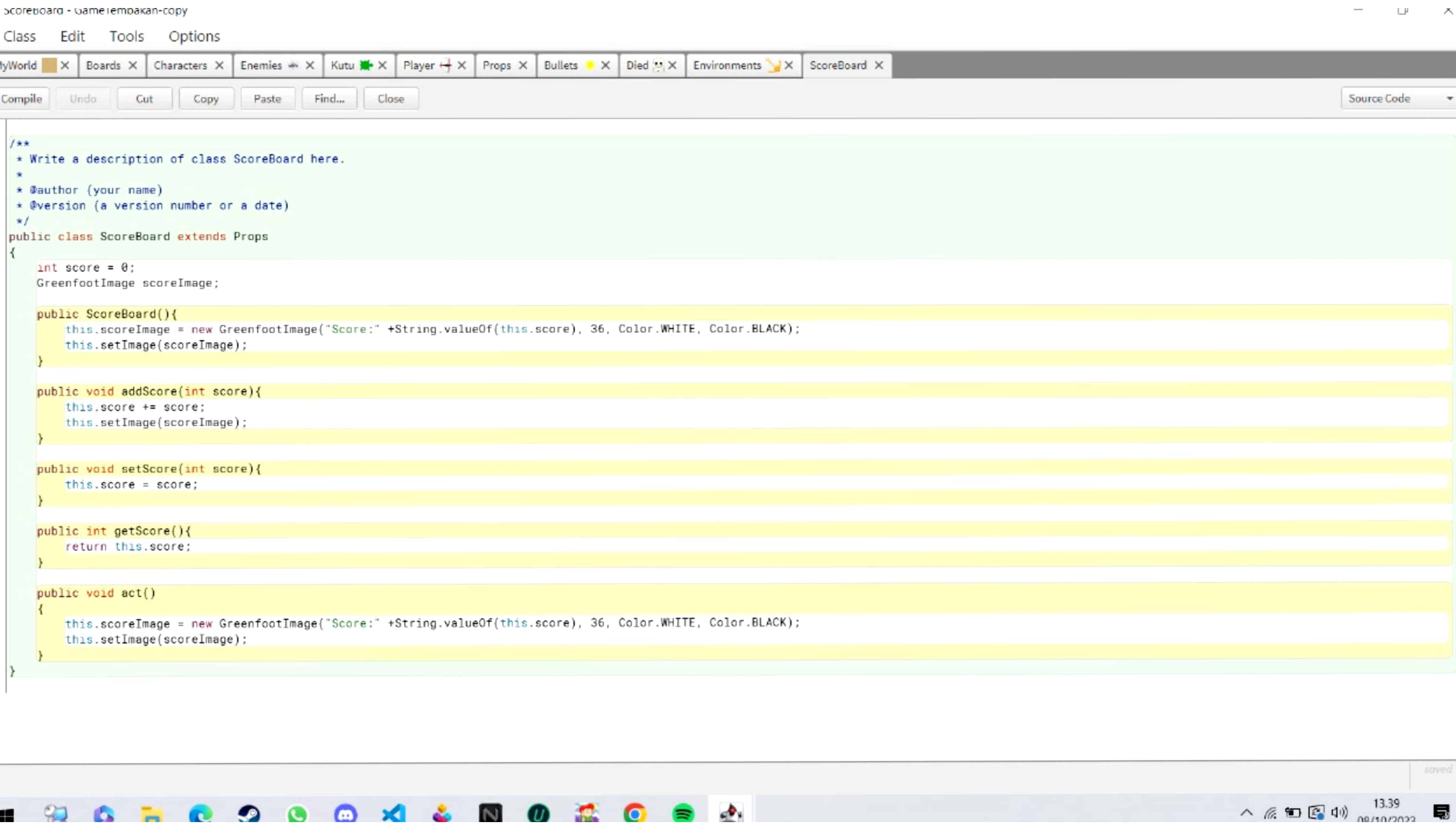
```java
/**
 * Write a description of class Bullets here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Bullets extends Props
{
    int speed = 10;
    public Bullets(){
        GreenfootImage img = this.getImage();
        img.scale(20, 20);
        this.setImage(img);
    }

    public Bullets(int bulletSpeed){
        GreenfootImage img = this.getImage();
        img.scale(30, 20);
        this.setImage(img);
        this.speed = bulletSpeed;
    }

    public void act()
    {
        this.move(speed);

        if(this.isAtEdge()){
            MyWorld wrld = (MyWorld)this.getWorld();
            wrld.incShotMiss();
            wrld.removeObject(this);

        }

    }
}
```


Class Edit Tools Options

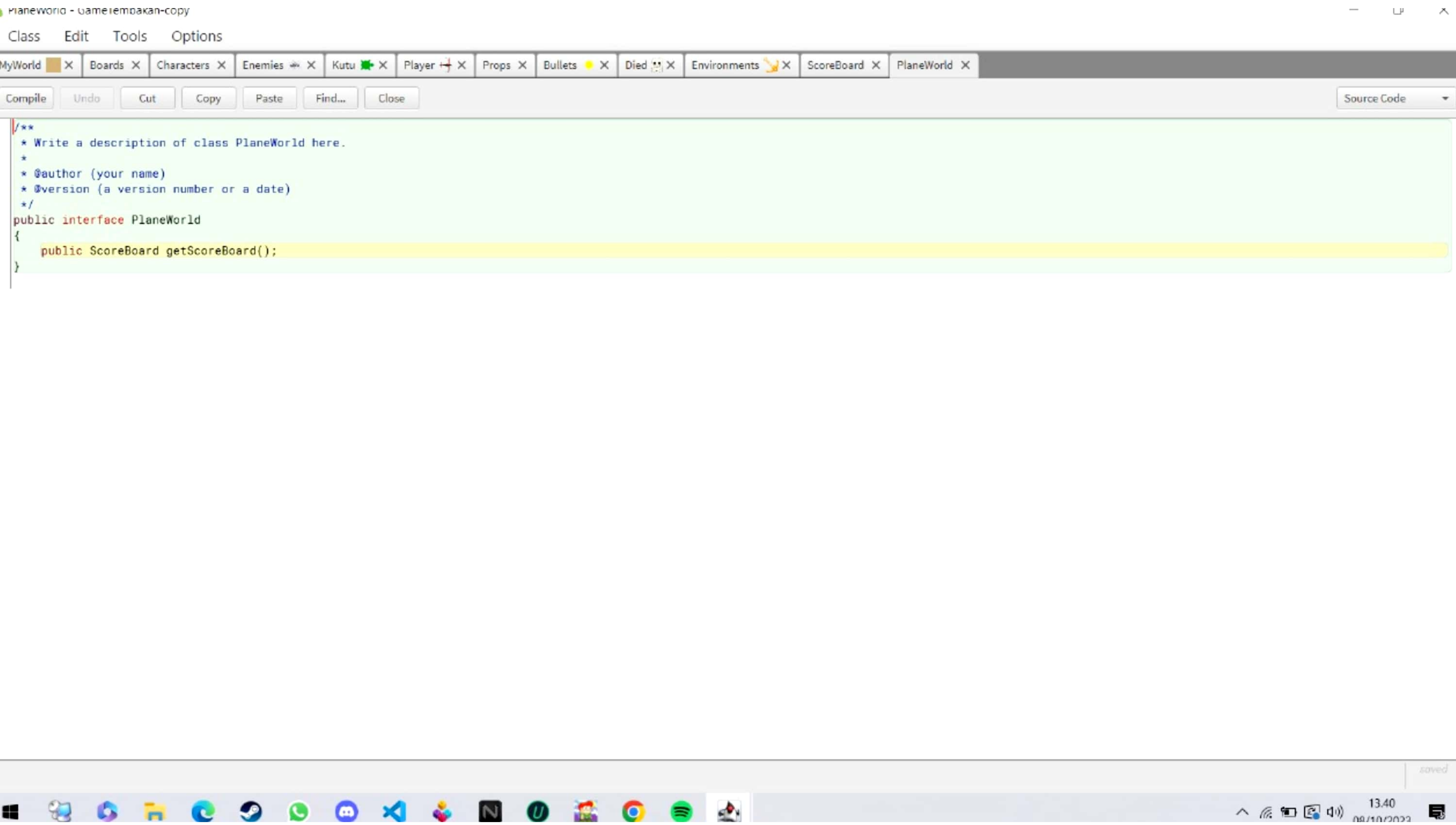
MyWorld | Boards | Characters | Enemies | Kutu | Player | Props | Bullets | Died | Environments

Compile | Undo | Cut | Copy | Paste | Find... | Close | Source Code

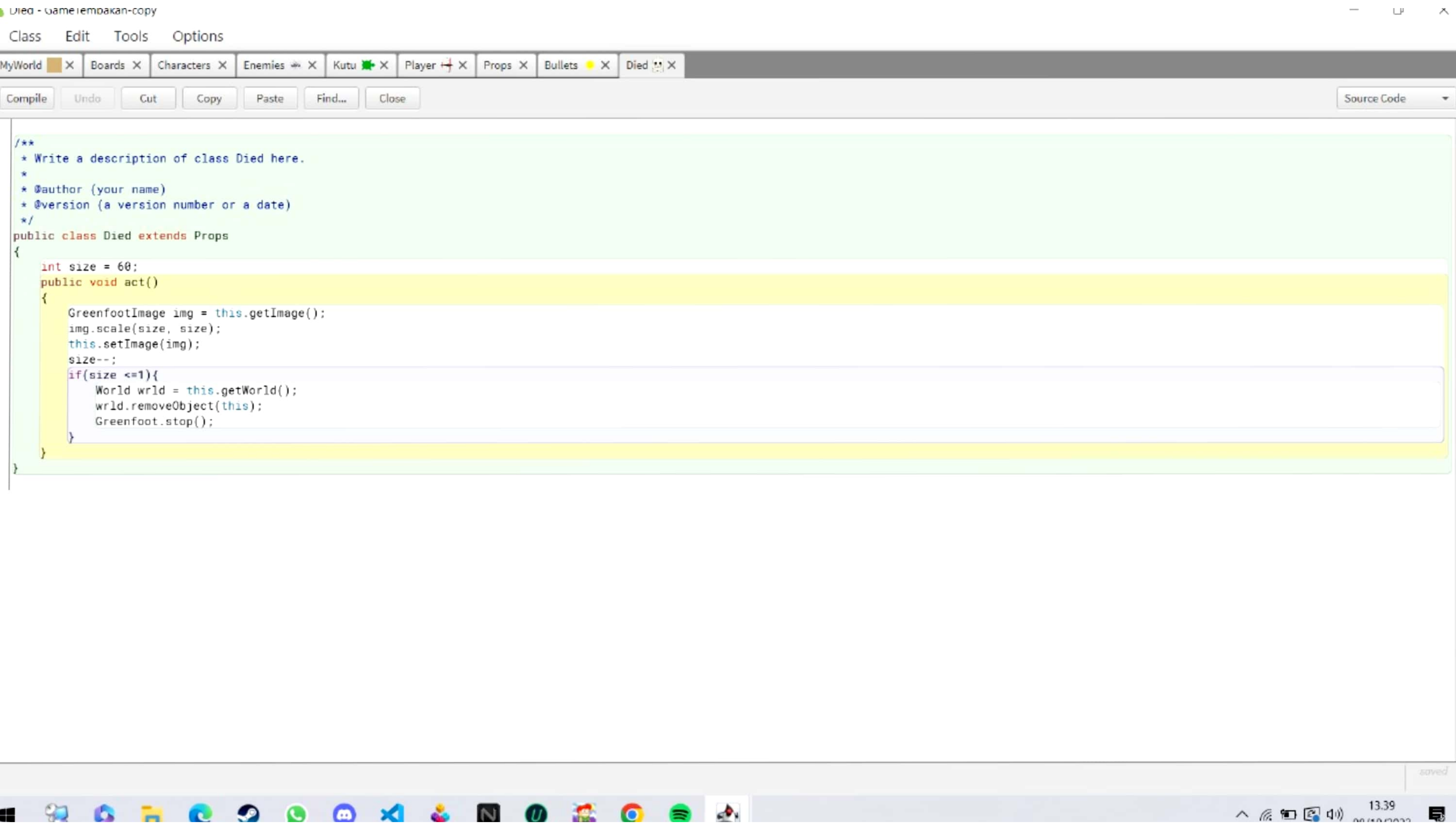
```java
/**
 * Write a description of class Environments here.
 * 
 * @author (your name) 
 * @version (a version number or a date)
 */
public class Environments extends Props
{
    /**
     * Act - do whatever the Environments wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        int gravity = 10;
        int newY = this.getY()+2;
        this.setLocation(this.getX(), newY);
        
        if(newY >= this.getWorld().getHeight()-1){
            this.getWorld().removeObject(this);
        }
    }
}
```



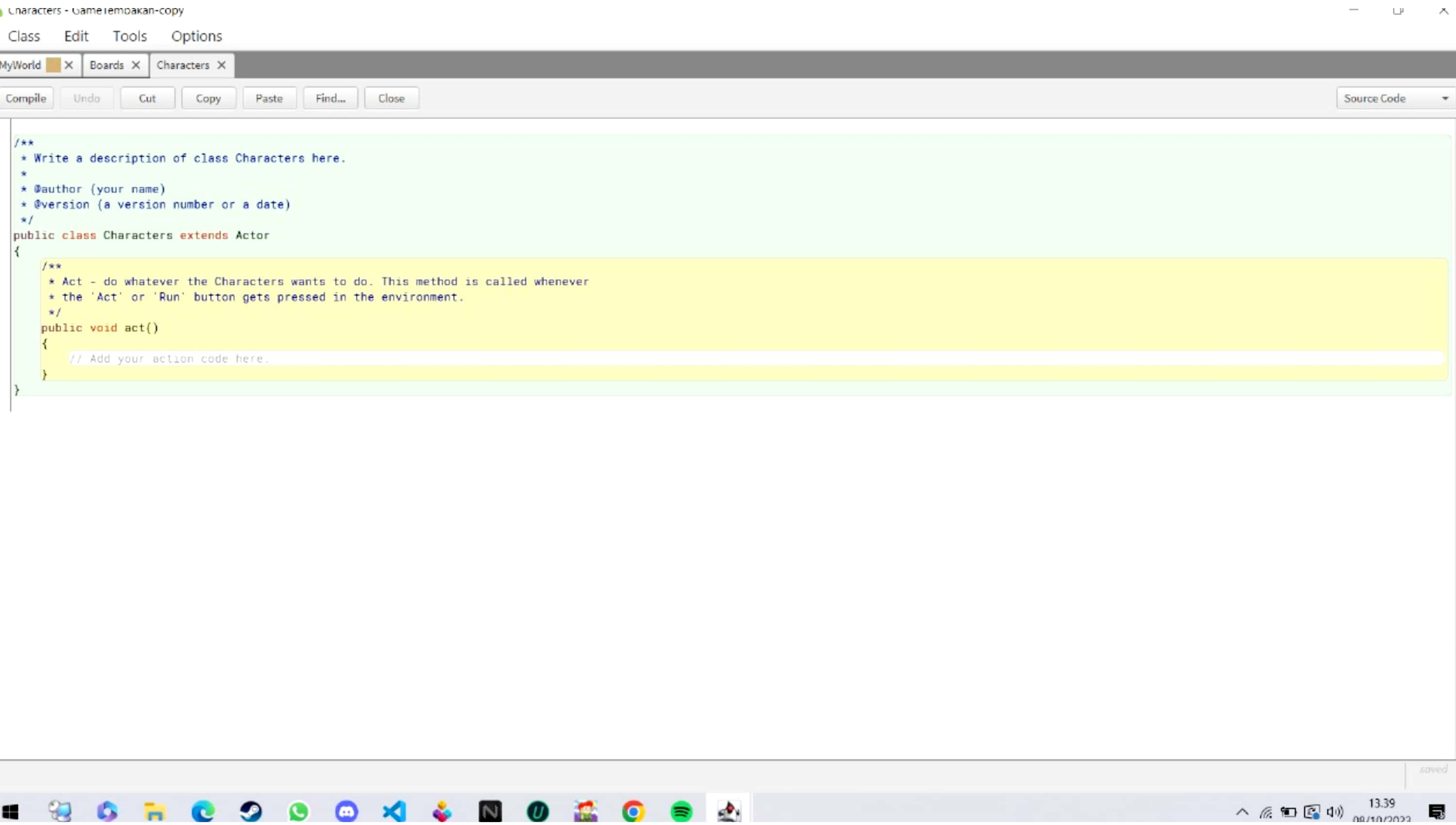
```java
/**
 * Write a description of class ScoreBoard here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class ScoreBoard extends Props
{
    int score = 0;
    GreenfootImage scoreImage;

    public ScoreBoard(){
        this.scoreImage = new GreenfootImage("Score:" +String.valueOf(this.score), 36, Color.WHITE, Color.BLACK);
        this.setImage(scoreImage);
    }

    public void addScore(int score){
        this.score += score;
        this.setImage(scoreImage);
    }

    public void setScore(int score){
        this.score = score;
    }

    public int getScore(){
        return this.score;
    }

    public void act()
    {
        this.scoreImage = new GreenfootImage("Score:" +String.valueOf(this.score), 36, Color.WHITE, Color.BLACK);
        this.setImage(scoreImage);
    }
}
```



```
/**
 * Write a description of class PlaneWorld here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public interface PlaneWorld
{
    public ScoreBoard getScoreBoard();
}
```



```java
/**
 * Write a description of class Died here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Died extends Props
{
    int size = 60;
    public void act()
    {
        GreenfootImage img = this.getImage();
        img.scale(size, size);
        this.setImage(img);
        size--;
        if(size <=1){
            World wrld = this.getWorld();
            wrld.removeObject(this);
            Greenfoot.stop();
        }
    }
}
```


Characters - GameTembakan-copy

Class Edit Tools Options

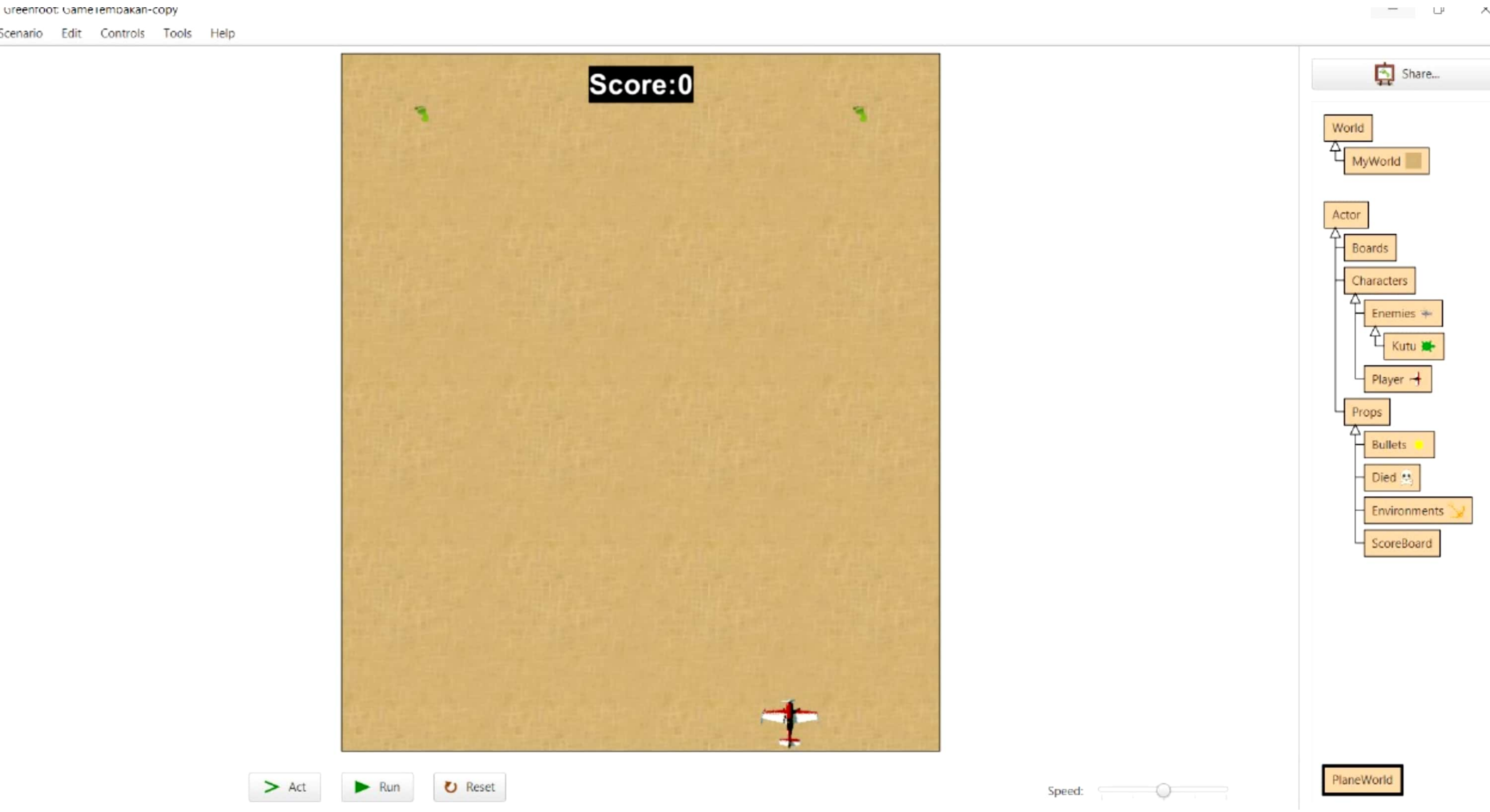
MyWorld Boards Characters

Compile Undo Cut Copy Paste Find... Close Source Code

```java
/**
 * Write a description of class Characters here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Characters extends Actor
{
    /**
     * Act - do whatever the Characters wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        // Add your action code here.
    }
}
```

saved

13.39
08/10/2023


Greenfoot: GameTembakan-copy
Scenario
Edit
Controls
Tools
Help
Score:0
Share...
World
MyWorld
Actor
Boards
Characters
Enemies
Kutu
Player
Props
Bullets
Died
Environments
ScoreBoard
PlaneWorld
Act
Run
Reset
Speed: