Emmanuel Adu Sarfo

Java Game

```
// Here is an example of a simple game loop in Java that updates the
//game state and redraws the screen at a regular interval:

public class GameLoop {

    private final int TICKS_PER_SECOND = 60;
    private final int SKIP_TICKS = 1000 / TICKS_PER_SECOND;
    private final int MAX_FRAMESKIP = 5;

    public void runGameLoop() {

        long nextGameTick = System.currentTimeMillis();
        int loops;

        while (true) {

            loops = 0;
            while (System.currentTimeMillis() > nextGameTick && loops < MAX_FRAMESKIP) {
                updateGameState(); // update the game state
                nextGameTick += SKIP_TICKS;
                loops++;
            }

            renderGame(); // redraw the screen
        }
    }

    private void updateGameState() {
        // update game state logic here
    }

    private void renderGame() {
        // redraw screen logic here
    }
}



//Here is an example of a simple Car class in Java that represents
//the player-controlled car and has methods to move left, right, up
//and down on the screen:

import java.awt.Rectangle;

public class Car {
    private int x;
    private int y;
    private int width;
    private int height;
    private int speed;
    private Rectangle collisionBox;

    public Car(int x, int y, int width, int height, int speed) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
        this.speed = speed;
        collisionBox = new Rectangle(x, y, width, height);
```

```java
    }

    public void moveLeft() {
        x -= speed;
        updateCollisionBox();
    }

    public void moveRight() {
        x += speed;
        updateCollisionBox();
    }

    public void moveUp() {
        y -= speed;
        updateCollisionBox();
    }

    public void moveDown() {
        y += speed;
        updateCollisionBox();
    }

    private void updateCollisionBox() {
        collisionBox.setBounds(x, y, width, height);
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }
    public int getWidth() {
        return width;
    }

    public int getHeight() {
        return height;
    }

    public Rectangle getCollisionBox() {
        return collisionBox;
    }

}


//Here is an example of a simple ObstacleCar class in Java that
//represents randomly placed obstacle cars that move around on the
//screen that the player's car must avoid:

import java.util.Random;

public class ObstacleCar {
    private int x;
    private int y;
    private int width;
    private int height;
    private int speed;
    private int direction;
    private Random rand;
    private Rectangle collisionBox;

    public ObstacleCar(int x, int y, int width, int height, int speed) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
        this.speed = speed;
        rand = new Random();
        direction = rand.nextInt(4);
        collisionBox = new Rectangle(x, y, width, height);
    }

    public void move() {
        if (direction == 0) {
            x -= speed;
        } else if (direction == 1) {
            x += speed;
        } else if (direction == 2) {
            y -= speed;
        } else {
            y += speed;
        }
        updateCollisionBox();
    }

    public void changeDirection() {
        direction = rand.nextInt(4);
    }

    private void updateCollisionBox() {
        collisionBox.setBounds(x, y, width, height);
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }
    public int getWidth() {
        return width;
    }

    public int getHeight() {
        return height;
    }

    public Rectangle getCollisionBox() {
        return collisionBox;
    }
}

//Here's an example of how to implement collision detection in the game loop:

import java.awt.Rectangle;

public class Game {
    private Car playerCar;
    private List<ObstacleCar> obstacleCars;
    private boolean gameOver;
    public void update() {
        // Move the player's car
        playerCar.move();
        // Move the obstacle cars
        for (ObstacleCar obstacleCar : obstacleCars) {
            obstacleCar.move();
        }
        // Check for collision
        for (ObstacleCar obstacleCar : obstacleCars) {
            if (playerCar.getCollisionBox().intersects(obstacleCar.getCollisionBox())) {
                gameOver = true;
                return;
            }
        }
    }
}


//Here's an example of a simple Scoreboard class in Java:

import java.awt.Graphics;
import java.awt.Font;

public class Scoreboard {
    private int score;
    private int level;
    private Font font;

    public Scoreboard() {
        score = 0;
        level = 1;
        font = new Font("Arial", Font.BOLD, 20);
    }

    public void incrementScore() {
        score += level;
    }

    public void draw(Graphics g) {
        g.setFont(font);
        g.drawString("Score: " + score, 10, 20);
        g.drawString("Level: " + level, 10, 40);
    }
}


//Here's an example of how to implement this feature in the game class:

public class Game {
    private int gameSpeed;
    private long lastSpeedIncrement;
    private final int SPEED_INCREMENT_INTERVAL = 30_000; // Increase speed every 30 seconds

    public void update() {
        // Move the player's car
        playerCar.move();
        // Move the obstacle cars
        for (ObstacleCar obstacleCar : obstacleCars) {
            obstacleCar.move(gameSpeed);
        }
        // Check for collision
        for (ObstacleCar obstacleCar : obstacleCars) {
            if (playerCar.getCollisionBox().intersects(obstacleCar.getCollisionBox())) {
                gameOver = true;
                return;
            }
        }
        // Increase game speed
        long currentTime = System.currentTimeMillis();
        if (currentTime - lastSpeedIncrement > SPEED_INCREMENT_INTERVAL) {
            gameSpeed++;
            lastSpeedIncrement = currentTime;
        }
    }
}



//Here's an example of how you might use these libraries to draw
//the player's car, the obstacle cars, and the road:

import java.awt.Color;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class Game {
    private PlayerCar playerCar;
    private ObstacleCar[] obstacleCars;
    private BufferedImage roadImage;

    public Game() {
        // Initialize player car
        playerCar = new PlayerCar(50, 50);
        // Initialize obstacle cars
        obstacleCars = new ObstacleCar[5];
        for (int i = 0; i < obstacleCars.length; i++) {
            obstacleCars[i] = new ObstacleCar(150 * i, 100);
        }
        try {
            roadImage = ImageIO.read(new File("road.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void draw(Graphics g) {
        // Draw the road
        g.drawImage(roadImage, 0, 0, null);
        // Draw the player's car
        playerCar.draw(g);
        // Draw the obstacle cars
        for (ObstacleCar obstacleCar : obstacleCars) {
            obstacleCar.draw(g);
        }
    }
}
```