```java
package hevs.fragil.patapon.units;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.graphics.g2d.Animation.PlayMode;
import com.badlogic.gdx.math.Vector2;

import ch.hevs.gdx2d.lib.GdxGraphics;
import hevs.fragil.patapon.drawables.SpriteSheet;
import hevs.fragil.patapon.mechanics.CurrentLevel;

public class UnitRender {
    private Look look = Look.DEFAULT;

    private Gesture gesture = Gesture.WALK;

    private State state = State.WALK;

    private float opacity = 1f;
    protected int nAttacks;

    private int bodyIndex;

    private SpriteSheet body, eye, arms, legs;

    protected float counter = -1;
    protected float cooldownCounter;

    private boolean gestureRunning = false;

    private Stabilizer pos = new Stabilizer();

    public boolean attack = false;

    /**
     * Constructor for a new UnitRender
     * @param bodyIndex : the body sprite index
     * @param preDelay : the delay for the attack animation
     */
    public UnitRender (int bodyIndex){
        this.bodyIndex = bodyIndex;
    }


    public Look getLook() {
        return look;
    }
    public void setLook(Look expression) {
        this.look = expression;
    }
    public Gesture getGesture() {
        return gesture;
    }
    public void setGesture(Gesture gesture) {
        this.gesture = gesture;
    }

    public void setState(State s) {
        state = s;
    }
    public State getState() {
        return state;
    }

    public Vector2 getPos(){
        return pos.getStabilizedPos();
    }
```

```java

    public void draw(GdxGraphics g, float x, float y, float angle) {
        if (state == State.DYING)
            drawDead(g, pos.stabilized(x,y), angle);
        else
            drawAlive(g, pos.stabilized(x,y));
    }
    private void drawAlive(GdxGraphics g, Vector2 position) {
        gestureSwitch();

        float stateTime = CurrentLevel.getLevel().getStateTime();
        int legsIndex = legs.drawAllFrames(stateTime, position);
        body.drawWalkAnimation(legsIndex, bodyIndex, position.x, position.y-5);
        eye.drawWalkAnimation(legsIndex, look.ordinal(), position.x, position.y-5);
        arms.drawFrames(stateTime, gesture.ordinal() * 4, 4, position.x, position.y-5);
    }

    private void drawDead(GdxGraphics g, Vector2 position, float angle) {
        gestureSwitch();
        legs.drawRotatedFrame(0, angle, position.x, (float) (position.y-Math.cos(angle)*32));
        body.drawRotatedFrame(bodyIndex, angle, position.x, (float) (position.y-Math.cos(angle)*32));
        eye.drawRotatedFrame(Look.DYING.ordinal(), angle, position.x, (float) (position.y-Math.cos(angle)*32));
        arms.drawRotatedFrame(0, angle, position.x, (float) (position.y-Math.cos(angle)*32));
    }
    private void gestureSwitch() {
        float dt = Gdx.graphics.getDeltaTime();

        if(counter >= 0){
            counter += dt;
        }
        if(counter >= 4 * arms.getFrameDuration()){
            gesture = Gesture.WALK;
            counter = -1;
        }
    }
    protected void launch(Gesture a) {
        if(gesture != a){
            setGesture(a);
            if(counter == -1)
                counter = 0;
        }
    }
    public boolean die() {
        opacity -= 0.005f;
        if (opacity <= 0) {
            return true;
        }
        return false;
    }

    /** This is only to load files in the PortableApplication onInit method */
    public void setLegsSprite(String url, int cols, int rows, boolean isEnnemi) {
        legs = new SpriteSheet(url, cols, rows, 0.2f, isEnnemi, PlayMode.LOOP);
    }

    /** This is only to load files in the PortableApplication onInit method */
    public void setBodySprite(String url, int cols, int rows) {
        body = new SpriteSheet(url, cols, rows, 1f, false, PlayMode.LOOP);
    }

    /** This is only to load files in the PortableApplication onInit method */
    public void setEyeSprite(String url, int cols, int rows) {
        eye = new SpriteSheet(url, cols, rows, 0.2f, false, PlayMode.LOOP);
    }

    /** This is only to load files in the PortableApplication onInit method */
    public void setArmsSprite(String url, int cols, int rows, boolean isEnnemi) {
        arms = new SpriteSheet(url, cols, rows, 0.2f, isEnnemi, PlayMode.NORMAL);
    }
    public boolean gestureRunning() {
        return gestureRunning;
    }
}
```