Atividade Framework 01

Nome: Nicolas Luís Crusco Rocha de Moraes

Objetivo

- Seguir o tutorial de criação do projeto base do LibGDX;
- Visitar o website do LibGDX;
- Começar a ler a documentação do LibGDX.
- Comentar essa vivência.

Código

```
package interno.mygdx.game;

import java.util.Iterator;

public class MyGdxGame extends ApplicationAdapter {
     private Texture dropImage;
     private Texture bucketImage;
     private Sound dropSound;
     private Music rainMusic;
     private SpriteBatch batch;
     private OrthographicCamera camera;
     private Rectangle bucket;
     private Array<Rectangle> raindrops;
     private long lastDropTime;

     @Override
     public void create() {
         // carrega as imagens do balde e da gota
         dropImage = new Texture(Gdx.files.internal("drop.png"));
         bucketImage = new Texture(Gdx.files.internal("bucket.png"));

         // carrega o audio de gota e o som da chuva
         dropSound = Gdx.audio.newSound(Gdx.files.internal("drop.wav"));
         rainMusic = Gdx.audio.newMusic(Gdx.files.internal("rain.mp3"));

         // faz um loop do som da chuva e o toca
         rainMusic.setLooping(true);
         rainMusic.play();

         // cria a camera e um variavel chamada de SpriteBatch
         camera = new OrthographicCamera();
         camera.setToOrtho(false, 800, 480);
         batch = new SpriteBatch();

         // cria um retangulo para o balde
         bucket = new Rectangle();
         bucket.x = 800 / 2 - 64 / 2; // centro do balde horizontal
         bucket.y = 20; // parte inferior
         bucket.width = 64;
         bucket.height = 64;

         // desenvolve os sistema das gotas caindo
         raindrops = new Array<Rectangle>();
         spawnRaindrop();
         }
```

```
        private void spawnRaindrop() {
            Rectangle raindrop = new Rectangle();
            raindrop.x = MathUtils.random(0, 800-64);
            raindrop.y = 480;
            raindrop.width = 64;
            raindrop.height = 64;
            raindrops.add(raindrop);
            lastDropTime = TimeUtils.nanoTime();
            }

        @Override
        public void render() {
            // aplicando a ideia do RGB do fundo
            ScreenUtils.clear(0, 0, 0.2f, 1);

            // faz um update da camera
            camera.update();

            batch.setProjectionMatrix(camera.combined);

            batch.begin();
            batch.draw(bucketImage, bucket.x, bucket.y);
            for(Rectangle raindrop: raindrops) {
               batch.draw(dropImage, raindrop.x, raindrop.y);
               }
            batch.end();

            if(Gdx.input.isTouched()) {
               Vector3 touchPos = new Vector3();
               touchPos.set(Gdx.input.getX(), Gdx.input.getY(), 0);
               camera.unproject(touchPos);
               bucket.x = touchPos.x - 64 / 2;
               }
            if(Gdx.input.isKeyPressed(Keys.LEFT)) bucket.x -= 200 * Gdx.graphics.getDeltaTime();
            if(Gdx.input.isKeyPressed(Keys.RIGHT)) bucket.x += 200 * Gdx.graphics.getDeltaTime();

            if(bucket.x < 0) bucket.x = 0;
            if(bucket.x > 800 - 64) bucket.x = 800 - 64;

            if(TimeUtils.nanoTime() - lastDropTime > 1000000000) spawnRaindrop();

             for (Iterator<Rectangle> iter = raindrops.iterator(); iter.hasNext(); ) {
                Rectangle raindrop = iter.next();
                raindrop.y -= 200 * Gdx.graphics.getDeltaTime();
                if(raindrop.y + 64 < 0) iter.remove();
                if(raindrop.overlaps(bucket)) {
                   dropSound.play();
                   iter.remove();
                   }
             }
}

        @Override
        public void dispose() {

            dropImage.dispose();
            bucketImage.dispose();
            dropSound.dispose();
            rainMusic.dispose();
            batch.dispose();
            }
        }
```

Conclusão

O conceito do LibGDX, é um conceito bem interresante para desenvolvimento de jogos, contendo um conceito que quase a assemelha a biblioteca BGI do C, porém com conceitos mais complexos para desenvolver projetos mais complexos, como o jogo que foi desenvolvido no codigo acima, além da questão da orientação ao objeto, auxilia no desenvolvimento e na organização do jogo.