Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Automatica și Tehnologii Informaționale

RAPORT

Lucrare de laborator Nr.5
la M.I.D.P.S.

A efectuat: st. gr. TI-144
I. Stratan

A verificat: lector.univ.
I.Cojanu
lector.sup.
S.Cojocaru

Chişinău 2016

LUCRAREA DE LABORATOR 5

Dezvoltarea unei aplicatii mobile

- **Visual Studio**
- **Xcode**
- **Android Studio**
- Eclipse
- NetBeans

Prerequisites:

- IDEs: Visual Studio, Xcode, Android Studio, Eclipse, NetBeans
- Limbaje de programare: C#, JavaScript, Objective C, Java, Swift
- Tehnologii si Frameworks: Windows Mobile, iOS, Android

Obiective:

- Cunostinte de baza privina arhitectura unei aplicatii mobile
- Cunostinte de baza ale platformei SDK

Conditii Generale:

Se considera ca ai trecut cu succes laboratorul daca ai urmat toti pasii din:

1. Submission Process
2. Trebuie sa elaborezi un program prototip care il vei arata in timpul laboratorului
3. Ai respectat DL (data limita)

Technical Prerequisites:

- Your application must be developed and tested in SDK included Emulator.
- You probably would like to run your application on real device.
- Your application must support multiple screen resolutions.

Laboratory Requirements:

- *Basic Level* (nota 5 || 6) :
    - Realizeaza o aplicatie simpla "Hello world" care va contine 2 butoane care vor afisa 2 pagini diferite, folosing 2 elemente diferite de interactiune

- *Normal Level* (nota 7 || 8):
    - Implimenteaza un simplu ceas sau stopwatch
- *Advanced Level* (nota 9 || 10):
    - Realizeaza o aplicatie care va implimenta tehnica *Pomodoro* **SAU**
    - O alta aplicatie sofisticata la alegere
        - Game
- *Bonus Point*
    - Foloseste libraria cross platform pentru a realiza o apliacatie cross platform (aplicatia poate fi compilata atit pe Android, cit si pe iOS)
    - Folosirea Facebook/Twitter/Google Maps API

Note: Alege si implimenteaza un singur nivel.

Analiza lucrari de laborator:

1. Am realizat o joaca simpla in Android Studio pe platforma android.
2. Am inclus 2 texturi si 2 music
3. Am creat functii pentru diferite evenimente
4. Am realizat controlul jocului cu touch screen
5. Am portat joaca si pe versiune desktop
6. Am realizat 3 clase , care fiecare clasa are rolul sau un joaca
7. Am folosit tehnologia LibGDX

Clasa GameScreen:

```
package info.fandroid.drop;

import com.badlogic.gdx.ApplicationAdapter;
import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.Input;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.audio.Music;
import com.badlogic.gdx.audio.Sound;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.graphics.Texture;
import com.badlogic.gdx.graphics.g2d.SpriteBatch;
import com.badlogic.gdx.math.MathUtils;
import com.badlogic.gdx.math.Vector3;
```

```
import com.badlogic.gdx.utils.Array;
import com.badlogic.gdx.utils.TimeUtils;
import com.badlogic.gdx.math.Rectangle;
import java.util.Iterator;

public class GameScreen implements Screen {
    final  Drop game;
    OrthographicCamera camera;
    SpriteBatch batch;
    Texture dropImage;
    Texture bucketImage;
    Sound dropSound;
    Music rainMusic;
    Rectangle bucket;
    Vector3 touchPos;
    Array<Rectangle> raindrops;
    long lastDropTime;
    int dropsGatchered;


    public  GameScreen (final Drop gam) {
        this.game =gam;

        camera = new OrthographicCamera();
        camera.setToOrtho(false, 800, 480);

        batch = new SpriteBatch();

        touchPos = new Vector3();

        dropImage = new Texture("droplet.png");
        bucketImage = new Texture("bucket.png");

        dropSound = Gdx.audio.newSound(Gdx.files.internal("waterdrop.wav"));
        rainMusic =
Gdx.audio.newMusic(Gdx.files.internal("undertreeinrain.mp3"));

        rainMusic.setLooping(true);
        rainMusic.play();

        bucket = new Rectangle();
        bucket.x = 800 / 2 - 64 / 2;
        bucket.y = 20;
        bucket.width = 64;
        bucket.height = 64;

        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 (float delta) {
        Gdx.gl.glClearColor(0, 0, 0.2f, 1);
```

```
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();

        game.batch.setProjectionMatrix(camera.combined);
        game.batch.begin();
        game.font.draw(game.batch,"Picaturi stranse:"+dropsGatchered,0,480);
        game.batch.draw(bucketImage, bucket.x, bucket.y);
        for (Rectangle raindrop: raindrops){
            game.batch.draw(dropImage, raindrop.x, raindrop.y);
        }
        game.batch.end();

        if(Gdx.input.isTouched()){
            touchPos.set(Gdx.input.getX(), Gdx.input.getY(), 0);
            camera.unproject(touchPos);
            bucket.x = (int) (touchPos.x -64 / 2);
        }

        if(Gdx.input.isKeyPressed(Input.Keys.LEFT)) bucket.x -= 200 *
Gdx.graphics.getDeltaTime();
        if(Gdx.input.isKeyPressed(Input.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();

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

    @Override
    public void resize(int width, int height) {

    }

    @Override
    public void pause() {

    }

    @Override
    public void resume() {

    }

    @Override
    public void hide() {

    }

    @Override
    public void dispose() {
```

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

    }

    @Override
    public void show() {
        rainMusic.play();
    }
}
```

Clasa Drop:

```
package info.fandroid.drop;


import com.badlogic.gdx.Game;

import com.badlogic.gdx.graphics.g2d.BitmapFont;

import com.badlogic.gdx.graphics.g2d.SpriteBatch;


/**

 * Created by ION on 10.04.2016.

 */

public class Drop  extends Game {


    SpriteBatch batch;

    BitmapFont font;


    @Override

    public void create() {

        batch= new SpriteBatch();

        font =new BitmapFont();

        this.setScreen(new MainMenuScreen(this));

    }


    @Override
```

```
    public void render() {
        super.render();
    }

    @Override
    public void dispose() {
        super.dispose();
        batch.dispose();
        font.dispose();
    }
}
```

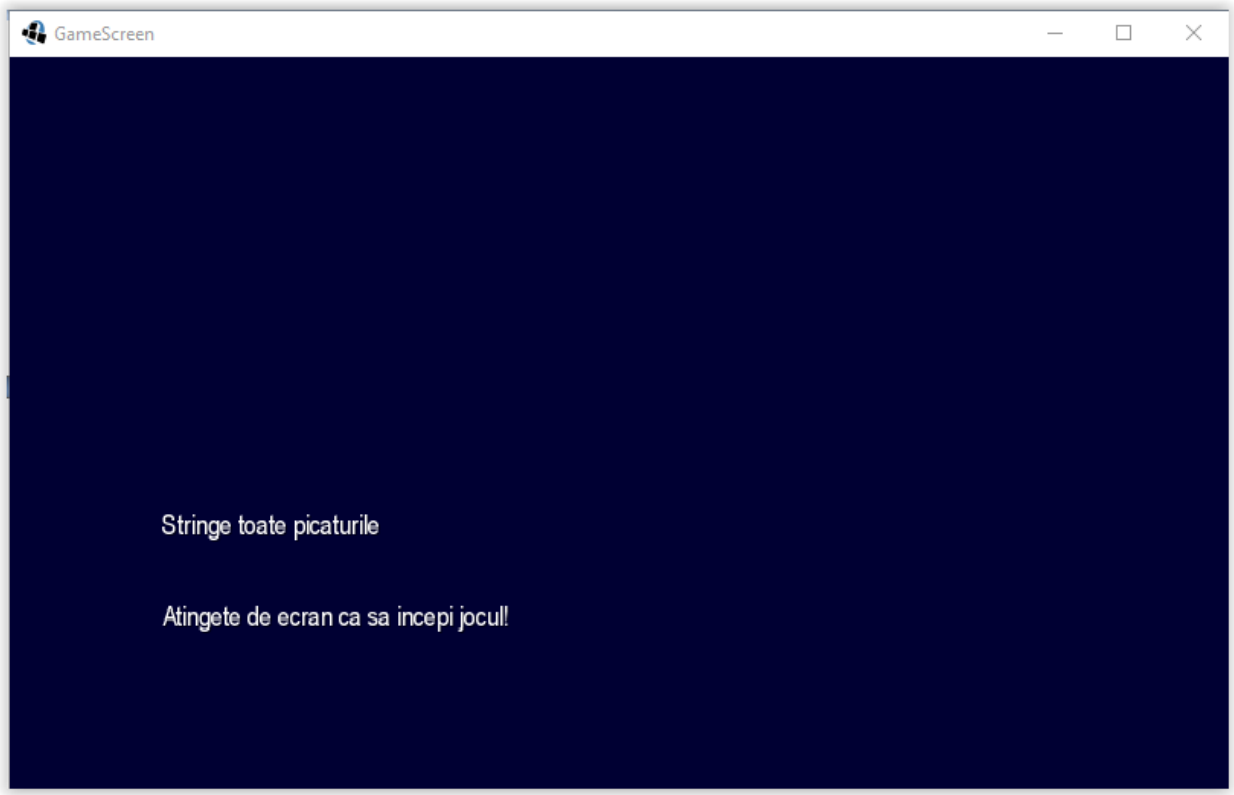
Clasa MainMenuScreen:

```
package info.fandroid.drop;

import com.badlogic.gdx.Gdx;
import com.badlogic.gdx.Screen;
import com.badlogic.gdx.graphics.GL20;
import com.badlogic.gdx.graphics.OrthographicCamera;
import com.badlogic.gdx.math.Rectangle;

/**
 * Created by ION on 10.04.2016.
 */
public class MainMenuScreen implements Screen {

   final  Drop game;
    OrthographicCamera camera;

    public MainMenuScreen(Drop gam) {
        game = gam;
        camera = new OrthographicCamera();
```

```
        camera.setToOrtho(false,800,400);
    }

    @Override
    public void show() {

    }

    @Override
    public void render(float delta) {
        Gdx.gl.glClearColor(0, 0, 0.2f, 1);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);

        camera.update();

        game.batch.setProjectionMatrix(camera.combined);
        game.batch.begin();
        game.font.draw(game.batch, "Stringe toate picaturile", 100,150);
        game.font.draw(game.batch, "Atingete de ecran ca sa incepi jocul!",
100,100);
        game.batch.end();

        if(Gdx.input.isTouched())
        {
            game.setScreen(new GameScreen(game));
            dispose();
        }
    }

    @Override
    public void resize(int width, int height) {
```

```
    }

    @Override
    public void pause() {

    }
    @Override
    public void resume() {

    }

    @Override
    public void hide() {

    }

    @Override
    public void dispose() {

    }
}
```

Imagini Run pe versiune Desktop



Imagini Run pe versiune Android

Concluzie

În urma efectuării acestui laborator, am făcut cunoștiință cu softul Android Studio, am înțeles cum se poate creea aplicații pe android, ios,desktop,html. Am folosit LibGDX. LibGdx este un puternic framework pentru vizualizare și crearea jocurilor care folosește limbajul java. El ne piermite să creăm jocuri pe platforma Windows, Linux, Mac OS, Ios, Html 5 și Android. Și noi putem să scrim odata codul pentru o aplicație și putem să o portăm pe oricare platformă fără modificație.