

Ministerul Educației și Tineretului al Republicii Moldova
Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronică
Catedra Automatica și Tehnologii Informaționale

Raport

MIDPS

Lucrarea de laborator Nr. 5

Tema: Dezvoltarea unei aplicatii mobile

A efectuat:

studentul grupei TI-141: **Benea Leonid**

A verificat:

lector superior: **Svetlana Cojocaru**

Interactive Development Environments

Laboratory Work #5

Tool-urile recomandate sunt marcate cu bold

Foloseste un nou IDE pentru fiecare lucrare de laborator 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

Objective:

- 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:

- Submission Process
- Trebuie sa elaborezi un program prototip care il vei arata in timpul laboratorului
- 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, folosind 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 aplicatie cross platform (aplicatia poate fi compilata atat pe Android, cit si pe iOS)
- Folosirea Facebook/Twitter/Google Maps API

Android Studio



A trecut mult timp de când am aflat primele detalii despre Android Studio. Anunțat pentru prima oară la începutul lui 2015, acest proiect trebuia să funcționeze pe sistemul WYSIWYG – What You See is What You Get. Cu alte cuvinte, trebuia să se manifeste printr-un mod simplu, intuitiv și eficient de a crea aplicații de Android fără cunoștințe avansate de programare.

În plus, amatorii platformei online open source GitHub vor fi fericiți să afle că pot cu ușurință să insereze bucăți de cod de acolo în propriile aplicații pentru Android. Nu mai rămâne decât să aveți o idee revoluționară de aplicație pentru ca în scurt timp aceasta să devină profitabilă în Google Play.

Programul : **mygame**

Codul sursa :

Drop.java :

```
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 Leonid on 29.05.2016.
 */
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();
    }
}
```

GameScreen.java :

```
package info.fandroid.drop;

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.math.MathUtils;
import com.badlogic.gdx.math.Rectangle;
import com.badlogic.gdx.math.Vector3;
import com.badlogic.gdx.utils.Array;
import com.badlogic.gdx.utils.TimeUtils;

import java.util.Iterator;

public class GameScreen implements Screen {

    final Drop game;
    OrthographicCamera camera;
    Texture dropImage;
    Texture bucketImage;
    Sound dropSound;
    Music rainMusic;
    Rectangle bucket;
    Vector3 touchPos;
    Array<Rectangle> raindrops;
    long lastDropTime;
    int dropsGathered;
    String dropString;

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

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

        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 strinse " + dropsGathered, 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 > 700000000 - (dropsGathered *
1000000)) 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)){
                dropsGathered++;
                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();
}
}

```

MainMenuScreen.java :

```

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;

/**
 * Created by Leonid on 29.05.2016.
 */
public class MainMenuScreen implements Screen {

    final Drop game;
    OrthographicCamera camera;
}

```

```

public MainMenuScreen(final Drop gam) {
    game = gam;

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

@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, "Laboratorul 5 ", 100, 150);
    game.font.draw(game.batch, "Atinge ecranul pentru a incepe ", 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() {
}
}

```




Concluzie:

Pentru prima dată am elaborat o aplicație pentru Android . Am utilizat pe deplin mediu de dezvoltare Android Studio .