Технический университет Молдовы

Кафедра Автоматики и Информационных Технологий

ОТЧЕТ

ПО ЛАБАРАТОРНОЙ РАБОТЕ №5

По предмету «Инструменты и среды разработки»

Работу выполнил: студентка группы TI-155

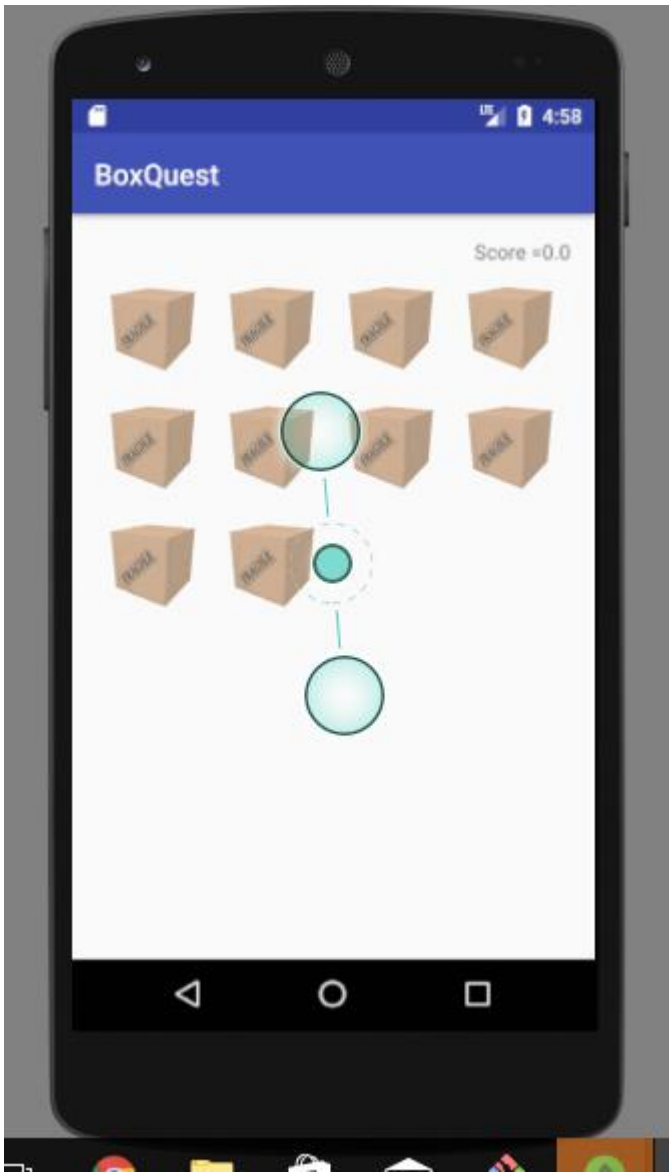
Голенцова Ольга

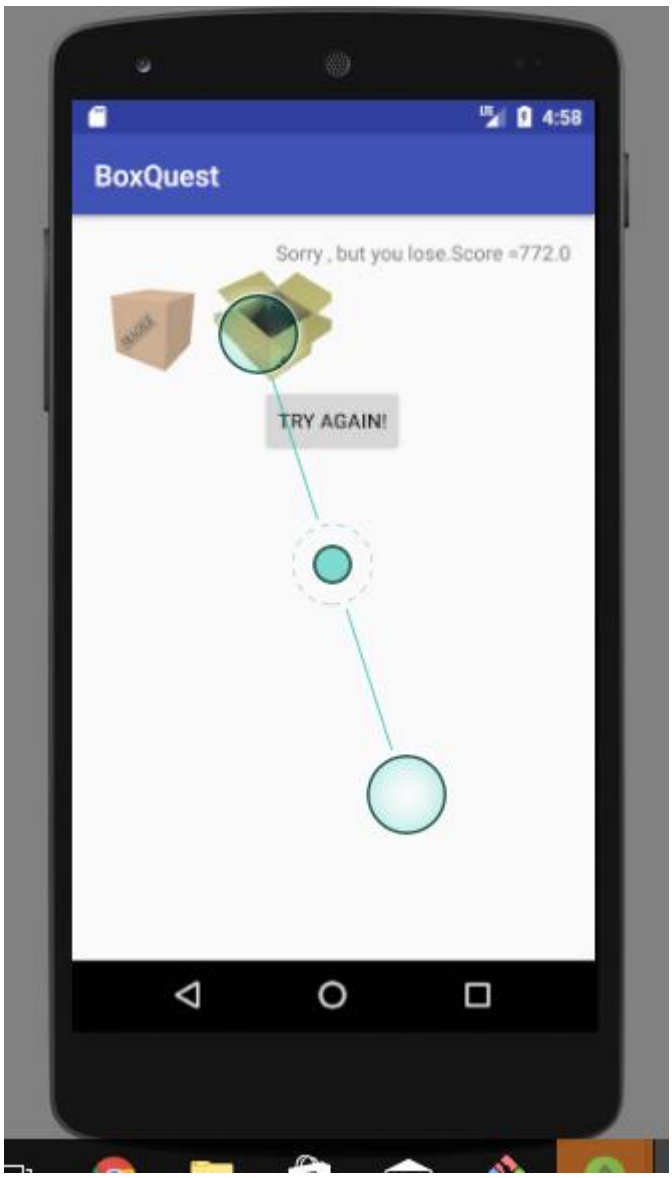
Проверил: *Гожин Виктор*

Кишинев 2017

Laboratory work Requirements:

- *Basic Level* (nota 5 || 6):




4:58
BoxQuest
Sorry , but you lose.Score =772.0
TRY AGAIN!

androiidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.olga.boxquest">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".view.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
```

```
        </activity>
    </application>

</manifest>
```

MainActivity.java

```
package com.olga.boxquest.view;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.olga.boxquest.controller.BoxPresenter;
import com.olga.boxquest.controller.Randomizer;
import com.olga.boxquest.model.Box;
import com.olga.boxquest.R;
import com.olga.boxquest.util.GridLayoutItemDecoration;
import com.olga.boxquest.controller.BoxRVAdapter;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity implements BoxPresenter {
    private RecyclerView mRecyclerView;
    private TextView mScoreTextView;
    private Button mNewGameButton;
    private ArrayList<Box> mBoxList = new ArrayList<>();
    private BoxRVAdapter mBoxRVAdapter;
    private double mScore = 0;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        buttonSetup();
        rvSetup();


    }

    private void rvSetup() {
        mRecyclerView = (RecyclerView) findViewById(R.id.game_rv);
//        mBoxRVAdapter = new BoxRVAdapter(mBoxList,this);
        //grid layout manager with 4 places for boxes
        mRecyclerView.setLayoutManager(new GridLayoutManager(this, 4));
        //decoration with space 2px
        mRecyclerView.addItemDecoration(new GridLayoutItemDecoration(2));
        //adapter setup
        mBoxRVAdapter = new BoxRVAdapter(mBoxList, MainActivity.this);
        mRecyclerView.setAdapter(new BoxRVAdapter(mBoxList, this));
    }

    private void buttonSetup() {
        mNewGameButton = (Button) findViewById(R.id.start_game_button);
        mNewGameButton.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View view) {
                //set score to 0 for every new game
                mScore = 0;
                String scoreText = getString(R.string.default_score_text) + mScore;
                mScoreTextView.setText(scoreText);
                //hide button
                view.setVisibility(View.GONE);
                mNewGameButton.setText(R.string.try_again);
                //game init
                mBoxList = Randomizer.getStartingBoxList();
                mBoxRVAdapter = new BoxRVAdapter(mBoxList, MainActivity.this);
                mRecyclerView.setAdapter(mBoxRVAdapter);

            }
        });

        mScoreTextView = (TextView) findViewById(R.id.score_text_view);
    }

    @Override
    public void onBoxClick(int size) {

        //score = chance to lose * 100 * lvl number
        double level = 11 - size;
        double loseChance = 1d / (double) size * 100;
        mScore += Math.round(level * loseChance);
        mScoreTextView.setText(getScoreText());
    }

    @Override
    public void onLose() {
        mNewGameButton.setVisibility(View.VISIBLE);
        String loseText = "Sorry , but you lose."+getScoreText();
        mScoreTextView.setText(loseText);
    }

    @Override
    public void onWin() {

        mNewGameButton.setVisibility(View.VISIBLE);
        String winText = "Congratulations , you win!"+getScoreText();
        mScoreTextView.setText(winText);
    }



    private String getScoreText() {
        return getString(R.string.default_score_text) + mScore;

    }
}
```

GridLayoutDecoration.java

```
package com.olga.boxquest.util;


import android.graphics.Rect;
import android.support.v7.widget.RecyclerView;
import android.view.View;

public class GridLayoutItemDecoration extends RecyclerView.ItemDecoration {
    private int space;

    public GridLayoutItemDecoration(int space) {
```

```
        this.space = space;
    }

    @Override
    public void getItemOffsets(Rect outRect, View view, RecyclerView parent,
RecyclerView.State state) {
        outRect.left = space;
        outRect.right = space;
        outRect.bottom = space;

        // Add top margin only for the first item to avoid double space between
items
        if (parent.getChildAdapterPosition(view) == 0)
            outRect.top = space;
    }
}
```

SquareRelativeLayout.java

```
package com.olga.boxquest.util;

import android.annotation.TargetApi;
import android.content.Context;
import android.os.Build;
import android.util.AttributeSet;
import android.widget.RelativeLayout;

public class SquareRelativeLayout extends RelativeLayout {

    public SquareRelativeLayout(Context context) {
        super(context);
    }

    public SquareRelativeLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public SquareRelativeLayout(Context context, AttributeSet attrs, int
defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    @TargetApi(Build.VERSION_CODES.LOLLIPOP)
    public SquareRelativeLayout(Context context, AttributeSet attrs,         int
defStyleAttr, int defStyleRes) {
        super(context, attrs, defStyleAttr, defStyleRes);
    }

    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        // Set a square layout.
        super.onMeasure(widthMeasureSpec, widthMeasureSpec);
    }

}
```

Box.java

```
package com.olga.boxquest.model;

public class Box {
    /**
     * if true - contains bomb , else not
     */
    private boolean hasBomb;
```

```
    public Box(boolean hasBomb) {
        this.hasBomb = hasBomb;
    }

    public boolean hasBomb() {
        return hasBomb;
    }

    @Override
    public String toString() {
        final StringBuffer sb = new StringBuffer("Box{");
        sb.append("hasBomb=").append(hasBomb);
        sb.append('}');
        return sb.toString();
    }
}
```

BoxPresenter.java

```
package com.olga.boxquest.controller;


public interface BoxPresenter {
    /**
     * Callback for list size for presenter
     * @param size size of list
     */
    void onBoxClick(int size);

    /**
     * Callback for lose state.
     */
    void onLose();

    /**
     * Callback for win state.
     */
    void onWin();
}
```

BoxRVAdapter.java

```
package com.olga.boxquest.controller;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import com.olga.boxquest.model.Box;
import com.olga.boxquest.R;

import java.util.List;

public class BoxRVAdapter extends RecyclerView.Adapter<BoxRVAdapter.BoxViewHolder>
{
    private List<Box> mBoxList;
    private BoxPresenter mPresenter;
    private  boolean mIsGameOver;



    public BoxRVAdapter(List<Box> boxList, BoxPresenter presenter) {
        mBoxList = boxList;
        mPresenter = presenter;
    }
```

```
    @Override
    public BoxViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        return new
BoxViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.box_layout,
parent, false));
    }

    @Override
    public void onBindViewHolder(BoxViewHolder holder, int position) {
    }

    @Override
    public int getItemCount() {
        return mBoxList.size();
    }

    public class BoxViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {
        private ImageView mImageView;


        public BoxViewHolder(View itemView) {
            super(itemView);
            mImageView = (ImageView) itemView.findViewById(R.id.bomb_image);
            mImageView.setOnClickListener(this);
        }

        @Override
        public void onClick(View view) {
            //if game over , no callback , else if bomb , change image to bomb , if
nor decrease size of list and mix items.
            //if user clicks too fast need to catch array of bounds exception
            try {
                if (!mIsGameOver) {
                    if (mBoxList.get(getAdapterPosition()).hasBomb()) {
                        //change image to bomb and end game
                        mIsGameOver = true;
                        mImageView.setImageResource(R.drawable.bomb_box);
                        mPresenter.onLose();
                    } else {
                        //delete item and mix items one more time
                            mPresenter.onBoxClick(mBoxList.size());
                            mBoxList.remove(getAdapterPosition());
                            notifyItemRemoved(getAdapterPosition());
                            notifyItemRangeChanged(getAdapterPosition(),
mBoxList.size());
                            Randomizer.mixList(mBoxList);
                            //if there is one item means no items remain , and user
has won the game.
                            if (getItemCount() == 1) {
                                mPresenter.onWin();
                                mIsGameOver = true;
                            }


                    }


                }
            } catch (ArrayIndexOutOfBoundsException e) {
                e.printStackTrace();
            }
```

```
        }
    }
}
```

Randomizer.java

```
package com.olga.boxquest.controller;

import com.olga.boxquest.model.Box;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;


public class Randomizer {
    private static Random mRandom = new Random();

    /**
     * Mix list by getting item from random index and inserting it in random
position.
     *
     * @param list list to mix
     */
    public static <T> void mixList(List<T> list) {
        for (int i = 0; i < list.size(); i++) {
            //random index where to insert
            int randomInsertIndex = mRandom.nextInt(list.size());
            //random index where to get item
            int randomItemIndex = mRandom.nextInt(list.size());
            T item = list.get(randomItemIndex);
            //removing it
            list.remove(randomItemIndex);
            //adding to new position
            list.add(randomInsertIndex, item);
        }

    }

    /**
     * Create new list with boxes , already mixed.
     *
     * @return list of mixed boxes.
     */
    public static ArrayList<Box> getStartingBoxList() {
        ArrayList<Box> boxList = new ArrayList<>();
        boxList.add(new Box(true));
        for (int i = 0; i < 9; i++) {
            boxList.add(new Box(false));
            mixList(boxList);
        }
        return boxList;
    }


}
```

Bomb

Empty Box

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".view.MainActivity">

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_centerInParent="true"
       android:id="@+id/start_game_button"
       android:layout_below="@+id/game_rv"
       android:text="@string/new_game"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/score_text_view"
        android:layout_alignParentRight="true"/>

    <android.support.v7.widget.RecyclerView
        android:layout_below="@id/score_text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@id/game_rv">
```

```
    </android.support.v7.widget.RecyclerView>
</RelativeLayout>
```

Box_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<com.olga.boxquest.util.SquareRelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
                                                android:layout_width="wrap_content"

android:layout_height="wrap_content">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="centerCrop"
        android:id="@+id/bomb_image"
        android:src="@drawable/empty_box"
        />

</com.olga.boxquest.util.SquareRelativeLayout>
```

Вывод: В ходе лабораторной работы я детальнее изучила работу в Android Studio. Разработана простейшая игра. Пользователь видит ряд боксов. При нажатии на бокс – открытии его – есть два сценария: либо бокс пустой, тогда он исчезает и все оставшиеся боксы перемешиваются, пользователю засчитываются очки(в зависимости от увеличение вероятности его проигрыша) либо в боксе оказывается бомба (картинка меняется) это означает проигрышь. В процессе разработки использована модель MVC – model-view-controller