# *Національний технічний університет України*

# *«Київський політехнічний інститут»*

#### ***Факультет інформатики та обчислювальної техніки***

## Лабораторна робота №4

*з курсу "****Комп’ютерна графіка****"*

***Виконав:***

*Бедь А.М.*

***Група*** *ІО-12,*

***Номер варіанту*** *1202*

***Номер підгрупи 15***

***Київ - 2013р.***

**Завдання.**

Відтворити рух довільно обраного об’єкту за заданою траєкторією по варіанту. По верхній частині квадрата проти годинникової стрілки.

**Теоретичні відомості.**

Для описання квадрата використовувався клас Rect, який описує прямокутник.

**Лістинг коду:**

**package** com.animation.graphicslab;

**import** android.app.Activity;

**import** android.content.pm.ActivityInfo;

**import** android.os.Bundle;

**import** android.view.Menu;

**import** android.view.Window;

**import** android.view.WindowManager;

**import** android.widget.LinearLayout;

**import** android.widget.SeekBar;

**import** android.widget.SeekBar.OnSeekBarChangeListener;

**public** **class** MainActivity **extends** Activity **implements** OnSeekBarChangeListener {

LinearLayout llAnimation;

SeekBar sbSize;

Animation animation;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);

requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);

getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*,

WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);

setContentView(R.layout.*main*);

sbSize = (SeekBar) findViewById(R.id.*sbSize*);

sbSize.setOnSeekBarChangeListener(**this**);

llAnimation = (LinearLayout) findViewById(R.id.*llAnimation*);

animation = **new** Animation(**this**);

llAnimation.addView(animation);

}

@Override

**public** **boolean** onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.*main*, menu);

**return** **true**;

}

@Override

**public** **void** onProgressChanged(SeekBar seekBar, **int** progress,

**boolean** fromUser) {

**switch** (seekBar.getId()) {

**case** R.id.*sbSize*:

animation.setSize(progress \* 10);

**break**;

}

}

@Override

**public** **void** onStartTrackingTouch(SeekBar seekBar) {

}

@Override

**public** **void** onStopTrackingTouch(SeekBar seekBar) {

}

**Animation.java**

**package** com.animation.graphicslab;

**import** android.content.Context;

**import** android.graphics.Bitmap;

**import** android.graphics.BitmapFactory;

**import** android.graphics.Canvas;

**import** android.graphics.Color;

**import** android.graphics.Paint;

**import** android.graphics.Paint.Style;

**import** android.graphics.Rect;

**import** android.view.SurfaceHolder;

**import** android.view.SurfaceHolder.Callback;

**import** android.view.SurfaceView;

**public** **class** Animation **extends** SurfaceView **implements** Callback {

**protected** **int** centerX;

**protected** **int** centerY;

**protected** **int** height;

**protected** **int** width;

**protected** Manager manager;

**protected** Monster monster;

**protected** Bitmap feld;

**protected** **int** size;

**private** Paint paint;

**protected** Rect rect;

**public** Animation(Context context) {

**super**(context);

feld = BitmapFactory.*decodeResource*(getResources(), R.drawable.*feld*);

monster = **new** Monster(getResources());

getHolder().addCallback(**this**);

paint = **new** Paint();

paint.setStyle(Style.*STROKE*);

paint.setColor(Color.*RED*);

paint.setTextSize(50);

}

@Override

**public** **void** surfaceCreated(SurfaceHolder holder) {

height = getHeight();

width = getWidth();

centerX = width / 2;

centerY = height / 2;

monster.scale = (**float**) (height / 6.0 / monster.bitmapMonster

.getHeight());

monster.change();

manager = **new** Manager(**this**);

manager.setRunning(**true**);

manager.fps = 100;

manager.start();

rect = **new** Rect(centerX - size, centerY - size, centerX + size, centerY

+ size);

monster.x = rect.right;

monster.y = rect.bottom / 2;

}

@Override

**public** **void** surfaceDestroyed(SurfaceHolder holder) {

**boolean** flag = **true**;

manager.setRunning(**false**);

**while** (flag) {

**try** {

manager.join();

flag = **false**;

} **catch** (InterruptedException e) {

}

}

}

**public** **void** move() {

**if** (Math.*abs*(monster.x + monster.bitmapMonster.getWidth()

\* monster.scale / 2 - rect.left) <= 5) {

monster.dx = 0;

monster.dy = 5;

**if** (monster.y + monster.bitmapMonster.getHeight() \* monster.scale

/ 2 >= centerY) {

monster.x = (**int**) (rect.right - monster.bitmapMonster

.getWidth() \* monster.scale / 2);

monster.y = (**int**) (centerY - monster.bitmapMonster.getWidth()

\* monster.scale / 2);

}

} **else** {

**if** (Math.*abs*(monster.y + monster.bitmapMonster.getHeight()

\* monster.scale / 2 - rect.top) <= 5) {

monster.dx = -5;

monster.dy = 0;

} **else** {

**if** (Math.*abs*(monster.x + monster.bitmapMonster.getWidth()

\* monster.scale / 2 - rect.right) <= 5) {

monster.dx = 0;

monster.dy = -5;

}

}

}

monster.change();

}

**public** **void** onDraw(Canvas canvas) {

canvas.drawBitmap(feld, 0, 0, **null**);

canvas.drawRect(rect, paint);

monster.onDraw(canvas);

canvas.drawText(String.*valueOf*(size \* 2), 50, 50, paint);

}

**public** **void** setSize(**int** size) {

rect.set(centerX - size, centerY - size, centerX + size, centerY + size);

monster.x = (**int**) (rect.right - monster.bitmapMonster.getWidth()

\* monster.scale / 2);

monster.y = (**int**) (centerY - monster.bitmapMonster.getWidth()

\* monster.scale / 2);

**this**.size = size;

}

}

Manager.java

**package** com.animation.graphicslab;

**import** android.graphics.Canvas;

**public** **class** Manager **extends** Thread {

**private** **boolean** isRunning;

**private** Animation animation;

**protected** **int** fps;

**public** Manager(Animation animation) {

**this**.animation = animation;

}

**public** **void** setRunning(**boolean** isRunning) {

**this**.isRunning = isRunning;

}

@Override

**public** **void** run() {

**long** ticksPS = 1000 / fps;

**long** startTime;

**long** sleepTime;

**while** (isRunning) {

Canvas canvas = **null**;

startTime = System.*currentTimeMillis*();

**try** {

canvas = animation.getHolder().lockCanvas();

**synchronized** (animation.getHolder()) {

**if** (canvas != **null**) {

animation.onDraw(canvas);

animation.move();

}

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**if** (canvas != **null**) {

animation.getHolder().unlockCanvasAndPost(canvas);

}

}

sleepTime = ticksPS - (System.*currentTimeMillis*() - startTime);

**try** {

**if** (sleepTime > 0)

*sleep*(sleepTime);

**else**

*sleep*(10);

} **catch** (Exception e) {

}

}

}

}

**package** com.animation.graphicslab;

**import** android.content.res.Resources;

**import** android.graphics.Bitmap;

**import** android.graphics.BitmapFactory;

**import** android.graphics.Canvas;

**import** android.graphics.Matrix;

**public** **class** Monster {

**protected** **int** x;

**protected** **int** y;

**protected** **int** dx;

**protected** **int** dy;

**protected** Bitmap bitmapMonster;

**protected** Matrix matrix;

**protected** **float** scale;

**public** Monster(Resources resources){

bitmapMonster = BitmapFactory.*decodeResource*(resources, R.drawable.*monster*);

matrix = **new** Matrix();

}

**public** **void** change(){

x = x + dx;

y = y + dy;

matrix.setScale(scale, scale);

matrix.postTranslate(x, y);

}

**public** **void** onDraw(Canvas canvas){

canvas.drawBitmap(bitmapMonster, matrix, **null**);

}

}

**Висновок.**

Дана лабораторна робота була виконана під операційну систему Android. Використовувалося стандартне API, яке дозволило реалізувати поставлену задачу. Координати об’єкта, який рухається контролювалися за допомогою стандартного класу в API Android Rect. Також було реалізовано функціональність, яка дозволяє змінювати розміри квадрата. Для наглядності на екрані також промальовується сам квадрат.