

Министерство Образования Республики Молдова
Технический университет Молдовы
Кафедра Автоматики и Информационные Технологии

Отчет

По лабораторной работе №4
РАМ – Разработка Мобильных Приложений
Тема: Векторная анимация в Android

Выполнил:

ст. группы TI-154 Топал Е.

Проверил:

Русу К.

Кишинёв 2016

Тема работы:

Векторная анимация в Android

Цель работы:

Разработка приложения для рендеринга векторной анимации в Android.

Реализация графического интерфейса

Поскольку главной задачей данной лабораторной работы был рендеринг анимации, графический интерфейс был организован достаточно просто. Все свободное пространство главной Activity занято единственным View (AnimationView), кастомным классом, реализованном в приложении и наследующим от SurfaceView.

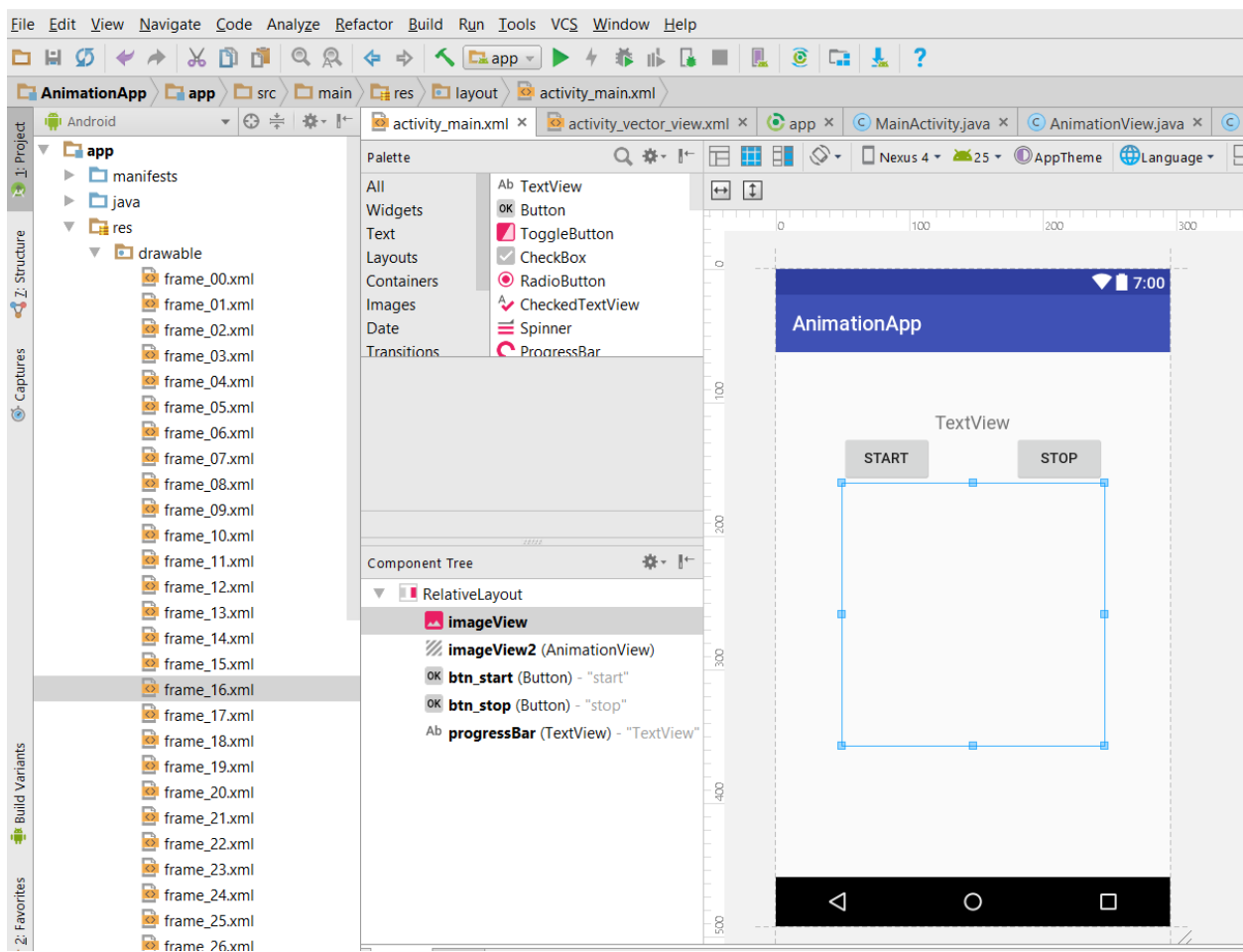


Рис. 1 – Интерфейс приложения

Подготовка анимации

Для выполнения работы была выбрана анимация в форме GIF-изображения, скачанная из интернета. После этого анимация была кадрирована с помощью бесплатной CLI-программы `magick`, в итоге вышло 46 кадров. Скрипт раскадровки представлен на ниже:

```
mkdir jpeg
mkdir jpg
mkdir png
mkdir bmp
magick convert Pacman.gif png/xx_%02d.png
magick convert Pacman.gif jpeg/xx_%02d.jpeg
magick convert Pacman.gif jpg/frame_%02d.jpg
magick convert Pacman.gif bmp/frame_%02d.bmp
```

После завершения кадрирования, была получена папка с 45 кадрами в разных форматах. С помощью программы `magick` и `potrace` была проведена пакетная обработка и конвертация изображений в формат `svg`.

png-to-svg+rename.sh

```
ext=.png
cd frames/png
rm -r ../../svg-converted/$ext
mkdir ../../svg-converted/$ext
for file in *$ext
do
    echo Processing file $file ...
    file_renamed=`basename $file $ext`
    file=`basename $file $ext`
    magick convert $file$ext $file_renamed.pnm
    potrace -s -o ../../svg-converted2/$ext/$file_renamed.svg
    $file_renamed.pnm
    rm $file_renamed.pnm
    #magick convert $file svg-converted/${file/_delay-0.05s$($ext)/.svg};
    #read -n1 -r -p "Press any key to continue..." key
done
```

Последним этапом была выполнена конвертация из формата `SVG` в `XML` с помощью открытого онлайн-сервиса `SVG to Vector Drawable`[4]. Процесс представлен на рис. 3:



Рис. 3 – Конвертация из SVG в Android Vector Drawable

Рендеринг анимации

VectorDrawableCompat drawable – это подготовка листа для рисования. При запуске Activity происходит загрузка и подготовка графических объектов:

```
public void loadDrawables() {
    Display display = getWindowManager().getDefaultDisplay();
    Point size = new Point();
    display.getSize(size);
    int width = size.x;
    int height = size.y;
    for (int i = 1; i < 45; i++) {
        VectorDrawableCompat drawable =
        VectorDrawableCompat.create(getResources(), getResourceId("frame" + i,
        "drawable"), null);
        int dheight = (int) (width / 1.5f);
        drawable.setBounds(0, (height - dheight) / 2, width, (height - dheight) /
        2 + dheight);
        vectorDrawables.add(drawable);
    }
}
```

Gif рисуется на Canvas в отдельном потоке, в классе который наследует от андроидовского класса SurfaceView – VectorView. Можно было поочередно вставлять в Image View, но производительность была бы очень низкой и frame per second, то есть количество кадров в секунду. При срабатывании события SurfaceCreated() создается новый поток, в теле которого и происходит рисование следующим образом[5]:

```

@Override
public void run() {
    while(running){
        Canvas canvas = null;
        try {
            if (System.currentTimeMillis()-prev<30) continue;
            canvas = surfaceHolder.lockCanvas(null);
            if (canvas == null)
                continue;
            canvas.drawColor(Color.parseColor("#FFFFFF"));
            //Log.d("CANVAS", "DRAWING "+index);
            Main.vectorDrawables.get(index).draw(canvas);
            if (index<43) index++;
            else
                index = 0;
            prev = System.currentTimeMillis();

        } finally {
            if (canvas != null) {
                surfaceHolder.unlockCanvasAndPost(canvas);
                postInvalidate();
            }
        }
    }
}

```

Поле index класса указывает на индекс элемента листа, который будет выведен на экран на этом кадре. Время между рендерингом кадров фиксировано и установлено в 30 мс (достаточно для плавного отображения анимации).

```

if (System.currentTimeMillis()-prev<30)
    continue;
canvas = surfaceHolder.lockCanvas(null);
if (canvas == null)
    continue;

```

Анимация будет рендериться циклично до тех пор, пока параметр running не примет значение false и поток не завершит свое выполнение. Это случится при срабатывании события SurfaceDestroyed для компонента VectorView.

```

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    drawThread.setRunning(false);
    Log.d("CANVAS", "DESTROYED");
}

```

Результаты работы

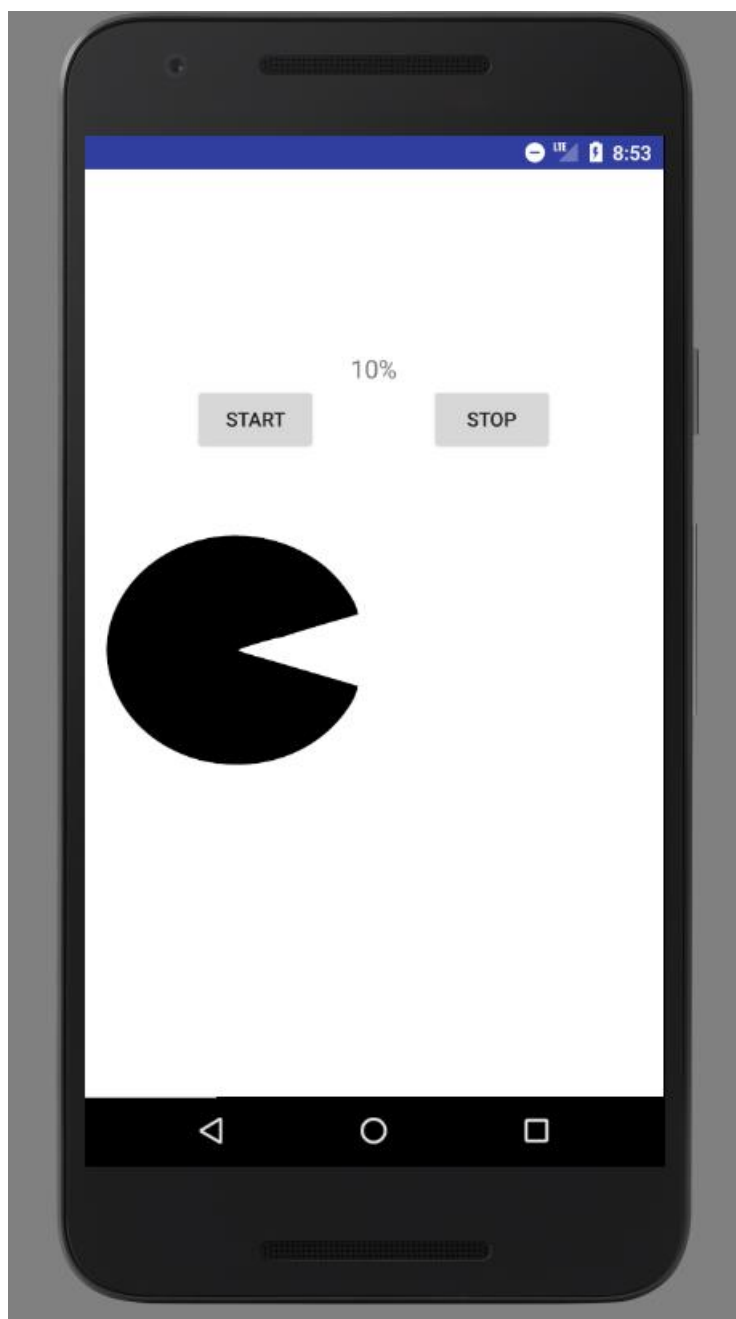


Рис. 4 – Рендеринг анимации в окне эмулятора

Выводы

В ходе данной лабораторной работы было реализовано приложение, которое осуществляет рендеринг векторной анимации стандартными средствами Android. Для выполнения этого задания, были пройдены этапы, описанные выше, начиная с кадрирования GIF-анимации и преобразования всех полученных изображений в векторную форму, заканчивая конвертацией из стандарта SVG в специальный XML формат Android Vector Drawable через специальный интернет - ресурс. Полученные знания и умения будут полезными при выполнении дальнейших лабораторных работ и при реализации более крупных проектов.

Библиография

1. Abstract GIF Animation - <https://static1.squarespace.com/static/551950f3e4b0ad84142e81c2/5730c3473c44d8b6196241a6/5730c34840261d2ef98ffadf/1462821734020/RainbowCircle.gif>
2. Exploder GIF Online - <http://ezgif.com/>
3. Vector Magic - <http://vectormagic.com/home>
4. SVG to Android Vector Drawable - <http://inloop.github.io/svg2android/>
5. Android Developers – Vector Drawable - <https://developer.android.com/reference/android/graphics/drawable/VectorDrawable.html>