МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

Лабораторная работа №7

по дисциплине: «Современные технологии разработки программного обеспечения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Чибисов И.А.* | Преподаватель | *Эстрайх И.В.*  *ст. преподаватель* | |
| Факультет | *АВТФ* |  |  | |
| Направление (специальность) подготовки | *09.04.03 – Прикладная информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *АПМ-22* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр | *010594214* |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| подпись | | подпись | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г | |

Новосибирск

2023

Цель работы: научиться использовать графику и анимацию в мобильных приложениях.

Задание на лабораторную работу

1. Создайте приложение, в котором используется графика. Поэкспериментируйте с цветами, расположением и размерами объектов, чтобы получилась красивая картинка. Проверьте, чтобы рисунок правильно отображался при смене ориентации устройства.

2. Создайте приложение, в котором используется анимация. На экран кроме объектов, задаваемых с помощью xml-файлов, поместите какое-либо изображение (как это делали с изображением собаки в первой части работы).

3. Включите в отчет цель работы, коды программ, скриншоты с результатами и вывод.

Ход работы:

Java

Создадим первый проект, в нем создадим статичную картинку

Картинку собаки добавим в drawable, остальное рисуем в методе onDraw. Код представлен ниже:

@Override  
protected void onDraw(Canvas canvas) {  
 super.onDraw(canvas);  
 int width = canvas.getWidth();  
 int height = canvas.getHeight();  
 // стиль Заливка  
 mPaint.setStyle(Paint.Style.*FILL*);  
 // закрашиваем холст белым цветом  
 mPaint.setColor(Color.*WHITE*);  
 canvas.drawPaint(mPaint);  
 // Рисуем желтый круг  
 mPaint.setAntiAlias(true);  
 mPaint.setColor(Color.*YELLOW*);  
 // canvas.drawCircle(950, 30, 25, mPaint);  
 canvas.drawCircle(width - 30, 30, 25, mPaint);  
 // Рисуем зеленый прямоугольник  
 mPaint.setColor(Color.*GREEN*);  
 // canvas.drawRect(20, 650, 950, 680, mPaint);  
 canvas.drawRect(0, canvas.getHeight() - 30, width, height, mPaint);  
 // Рисуем текст  
 mPaint.setColor(Color.*BLUE*);  
 mPaint.setStyle(Paint.Style.*FILL*);  
 mPaint.setAntiAlias(true);  
 mPaint.setTextSize(32);  
 // canvas.drawText("Собака на лужайке ", 30, 648, mPaint);  
 canvas.drawText("Собака на лужайке ", 30, height - 32, mPaint);  
 // Текст под углом  
 // int x = 810;  
 int x = width - 170;  
 int y = 190;  
 mPaint.setColor(Color.*GRAY*);  
 mPaint.setTextSize(27);  
 String beam = "Лучик солнца!";  
 canvas.save();  
 // Создаем ограничивающий прямоугольник для наклонного текста  
 // поворачиваем холст по центру текста  
 canvas.rotate(-45, x + mRect.exactCenterX(), y +  
 mRect.exactCenterY());  
 // Рисуем текст  
 mPaint.setStyle(Paint.Style.*FILL*);  
 canvas.drawText(beam, x, y, mPaint);  
 // восстанавливаем холст  
 canvas.restore();  
 // Выводим изображение  
 // canvas.drawBitmap(mBitmap, 450, 530, mPaint);  
 canvas.drawBitmap(mBitmap, width - mBitmap.getWidth(), height -  
 mBitmap.getHeight() - 30, mPaint);  
}

На рисунках 1-2 показано выполнение программы, под разными углами поворота экрана.

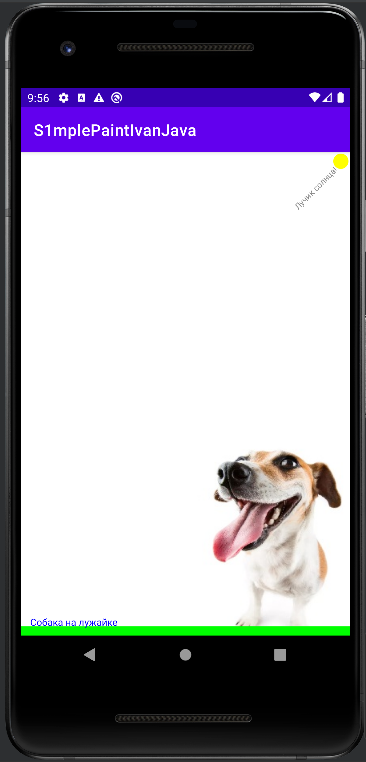


Рисунок 1 – Вертикальная ориентация

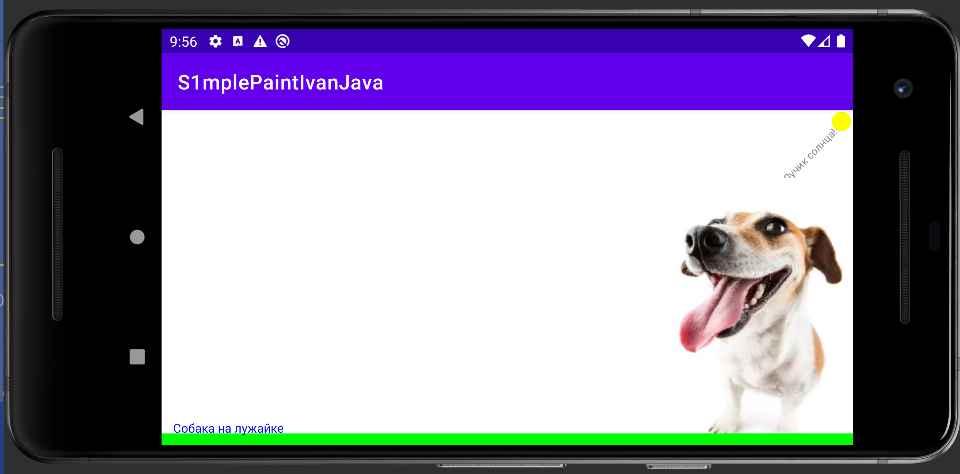


Рисунок 2 – Горизонтальная ориентация

Как видно на скриншотов в обеих ориентациях приложения выглядят великолепно.

Создадим второй проект, в нем будет анимация.  
 Здесь анимацию мы создаем в xml файлайх в папке res/anim. Так же как и сами рисунки, созданы в xml и лежат в res/drawable. Подробный код представлен в проекте.

Картинку собаки поместим в drawable.

Класс для создания анимации и добавления картинки представлен ниже.

public class MainActivity extends Activity {  
 private Paint mPaint = new Paint();  
 private Bitmap mBitmap;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
// Получим ссылку на солнце  
 ImageView sunImageView = (ImageView)  
 findViewById(R.id.*sun*);  
// Анимация для восхода солнца  
 Animation sunRiseAnimation =  
 AnimationUtils.*loadAnimation*(this, R.anim.*sun\_rise*);  
// Подключаем анимацию к нужному View  
 sunImageView.startAnimation(sunRiseAnimation);  
// Получим ссылку на часы  
 ImageView clockImageView = (ImageView) findViewById(R.id.*clock*);  
// анимация для вращения часов  
 Animation clockTurnAnimation = AnimationUtils.*loadAnimation*(this,  
 R.anim.*clock\_turn*);  
 clockImageView.startAnimation(clockTurnAnimation);  
// получим ссылку на часовую стрелку  
 ImageView hourImageView = (ImageView)  
 findViewById(R.id.*hour\_hand*);  
// анимация для стрелки  
 Animation hourTurnAnimation = AnimationUtils.*loadAnimation*(this,  
 R.anim.*hour\_turn*);  
// присоединяем анимацию  
 hourImageView.startAnimation(hourTurnAnimation);  
  
 ImageView myImageView = (ImageView) findViewById(R.id.*imageView*);  
 myImageView.setImageResource(R.drawable.*dog*);  
  
 }  
}

На рисунках 3-4 показано выполнение программы, под разными углами поворота экрана.



Рисунок 3 – Вертикальная ориентация



Рисунок 4 – Горизонтальная ориентация

Как видно на скриншотов в обеих ориентациях приложения выглядят великолепно.

Вывод: Мы научились использовать графику и анимацию в мобильных приложениях.