Министерство образования и науки Российской Федерации

Калужский филиал федерального государственного бюджетного образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ "Фундаментальные науки"

КАФЕДРА "Программное обеспечение ЭВМ, информационные технологии и прикладная математика"

О Т Ч Е Т

Лабораторная работа №2

ДИСЦИПЛИНА: " Мобильные операционные системы "

ТЕМА: " ИСПОЛЬЗОВАНИЕ МЕНЮ И ВИДЖЕТОВ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ДЛЯ РАЗРАБОТКИ ПРОСТЫХ ANDROID ПРИЛОЖЕНИЙ "

Выполнил: студент гр. ИТД-81 Турченко С.А._____

Проверил: Гришунов С.С. _____

Дата сдачи (защиты) лабораторной работы:

Результаты сдачи (защиты):

Количество рейтинговых баллов

Оценка

Калуга, 2018 г.

Цель:

1. Научиться использовать графический и текстовый режимы для создания интерфейса приложения
2. Изучить особенности реализации обработчиков событий
3. Научиться реализовывать логику работы приложения с учетом специфики платформы Android
4. Понять логику работы приложения с меню с учетом специфики платформы Android
5. Разработать и подключить разные типы меню с учетом аппаратных ограничений мобильных устройств
6. Понять особенности реализации обработчиков пунктов меню

Задача:

В качестве результата работы приложения необходимо создать не менее шести логов. Также необходимо добавить не менее четырех уведомлений. В меню, при имеющейся возможности, необходимо добавить иконки.

Вариант №2

Используя меню Options (подменю) разработать приложение для выбора цвета (4 цвета), фигуры (3 фигуры) и выбора области Activity для вывода фигуры (по центру, слева, справа, сверху и внизу). В меню предусмотреть пункт очистки Activity т.е. заливка layout белым цветом. Изначально экран пустой с белым цветом фона. Если пользователь выбирает цвет и область вывода, то они запоминаются, но фигура не отображается. Если пользователь выбирает фигуру, то она отображается цветом по умолчанию и в области по умолчанию, например, в центре. В дальнейшем цвет, область вывода и фигура меняются в зависимости от выбора пунктов меню.

Код программы

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    DrawView draw;
    DrawOption option;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        this.setDraw(new DrawOption(true));
        this.option = new DrawOption(false);
    }

    public void setDraw(DrawOption option)
    {
        draw = new DrawView(this, option);
        setContentView(draw);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        Log.d("activity", "create menu");
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_selection, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Log.d("activity", "selected option");
        switch (item.getItemId()) {
            case R.id.menu_color_black:
                option.Color = Color.BLACK;
                Toast.makeText(this, R.string.menu_action_color_black_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_color_gray:
                option.Color = Color.GRAY;
                Toast.makeText(this, R.string.menu_action_color_gray_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_color_blue:
                option.Color = Color.BLUE;
                Toast.makeText(this, R.string.menu_action_color_blue_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_color_red:
                option.Color = Color.RED;
                Toast.makeText(this, R.string.menu_action_color_red_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_form_circle:
                option.Figure = DrawOption.eFigure.CIRCLE;
                Toast.makeText(this, R.string.menu_action_figure_circle_hint,
Toast.LENGTH_SHORT).show();
                drawFigure();
                return true;
            case R.id.menu_form_square:
                option.Figure = DrawOption.eFigure.SQUARE;
```

```
                Toast.makeText(this, R.string.menu_action_figure_square_hint,
Toast.LENGTH_SHORT).show();
                drawFigure();
                return true;
            case R.id.menu_form_rectangle:
                option.Figure = DrawOption.eFigure.RECTANGLE;
                Toast.makeText(this, R.string.menu_action_figure_rectangle_hint,
Toast.LENGTH_SHORT).show();
                drawFigure();
                return true;
            case R.id.menu_position_center:
                option.Position = DrawOption.ePosition.CENTER;
                Toast.makeText(this, R.string.menu_action_position_center_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_position_left:
                option.Position = DrawOption.ePosition.LEFT;
                Toast.makeText(this, R.string.menu_action_position_left_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_position_right:
                option.Position = DrawOption.ePosition.RIGHT;
                Toast.makeText(this, R.string.menu_action_position_right_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_position_bottom:
                option.Position = DrawOption.ePosition.BOTTOM;
                Toast.makeText(this, R.string.menu_action_position_bottom_hint,
Toast.LENGTH_SHORT).show();
                return true;
            case R.id.menu_position_top:
                option.Position = DrawOption.ePosition.TOP;
                Toast.makeText(this, R.string.menu_action_position_top_hint,
Toast.LENGTH_SHORT).show();
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }

    }

    public void onScreenClear(MenuItem item) {
        Log.d("activity", "clicked on clear button");
        this.setDraw(new DrawOption(true));
    }

    public  void drawFigure() {
        this.setDraw(this.option);
    }

    public void onDrawFigure(MenuItem item) {
        Log.d("activity", "clicked on draw button");
        this.drawFigure();
    }
}
```

DrawOption.java

```
public class DrawOption {


    enum eFigure {
        SQUARE,
        CIRCLE,
        RECTANGLE
```

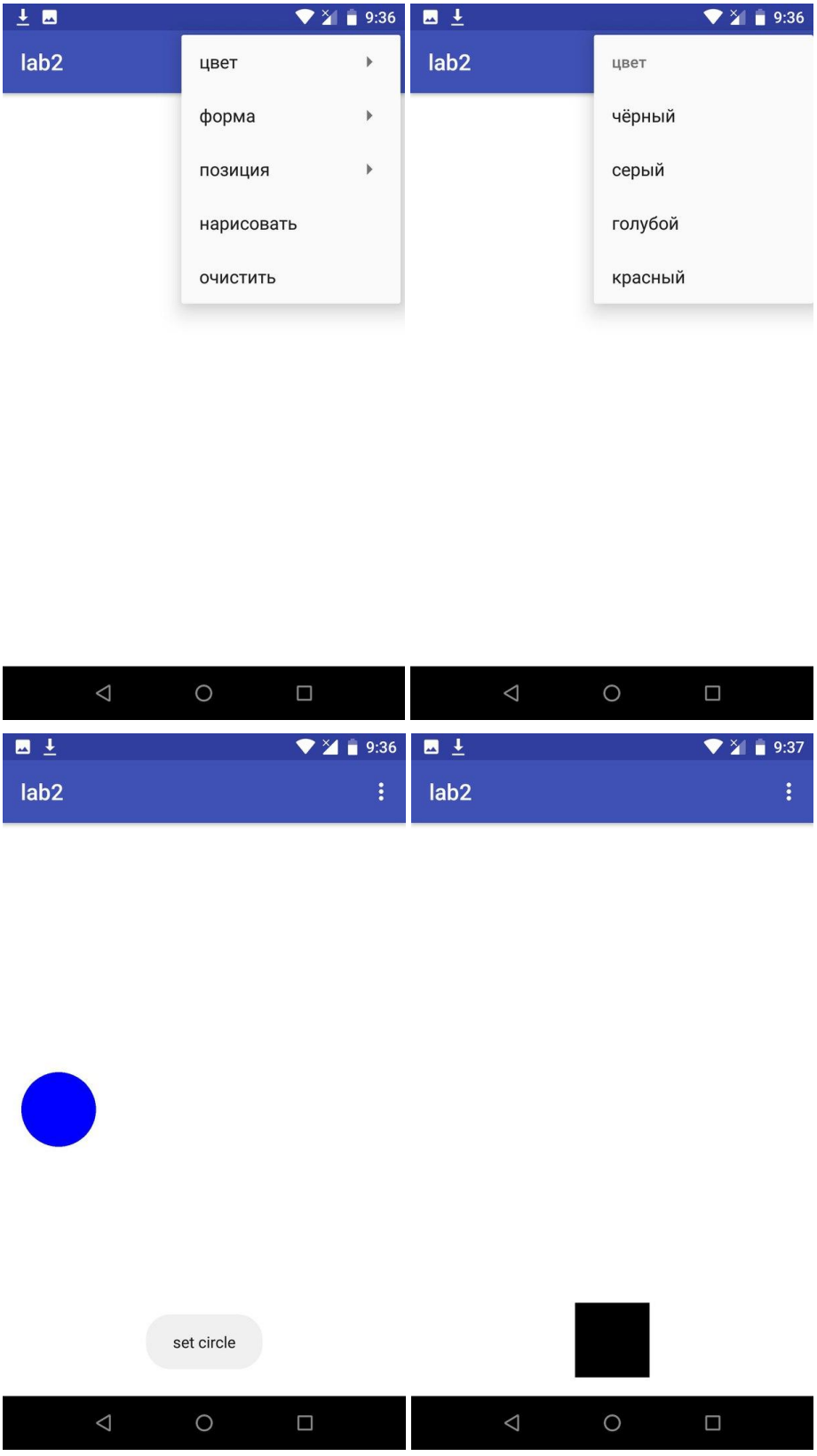
```
    }

    public enum ePosition {
        CENTER,
        LEFT,
        RIGHT,
        TOP,
        BOTTOM
    }

    public Boolean Clear;
    public int Color;
    public eFigure Figure;
    public ePosition Position;

    public DrawOption(Boolean isClear) {
        this.Clear = isClear;
        this.Figure = eFigure.CIRCLE;
        this.Position = ePosition.CENTER;
        this.Color = android.graphics.Color.BLACK;
    }

}
```

DrawView.java

```
class DrawView extends View {

    Paint p;
    Canvas canvas;
    DrawOption option;


    public DrawView(Context context, DrawOption option) {
        super(context);
        p = new Paint();
        p.setColor(option.Clear ? Color.WHITE : option.Color);
        this.option = option;
    }


    class Position {
        public int X;
        public int Y;
    }

    private Position calculatePosition(int width, int height) {
        Position result = new Position();
        DrawOption.ePosition position = this.option.Position;

        int shiftX = 100;
        int shiftY = 100;
        int padding = 50;

        if (position == DrawOption.ePosition.CENTER) {
            result.X = width / 2 - shiftX;
            result.Y = height / 2 - shiftY;
        }

        if (position == DrawOption.ePosition.TOP) {
            result.X = width / 2 - shiftX;
            result.Y = padding;
        }

        if (position == DrawOption.ePosition.BOTTOM) {
```

```
            result.X = width / 2 - shiftX;
            result.Y = height - shiftY * 2 - padding;
        }

        if (position == DrawOption.ePosition.LEFT) {
            result.X = padding;
            result.Y = height / 2 - shiftY;
        }

        if (position == DrawOption.ePosition.RIGHT) {
            result.X = width - shiftX * 2 - padding;
            result.Y = height / 2 - shiftY;
        }

        Log.d("figure", "calculated position as X:" + Integer.toString(result.X) +
            " and Y:" + Integer.toString(result.Y));
        return result;
    }

    @Override
    protected void onDraw(Canvas canvas) {
        this.canvas = canvas;

        // clear screen
        this.canvas.drawARGB(255, 255, 255, 255);
        Log.d("canvas", "canvas cleared");

        if (this.option.Clear)
            return;

        Position position = calculatePosition(canvas.getWidth(),
canvas.getHeight());

        Log.d("figure", "calculating figure and output it");
        switch (this.option.Figure) {
            case CIRCLE:
                Log.d("canvas", "draw circle");
                canvas.drawCircle(position.X + 100, position.Y + 100, 100, p);
                break;
            case SQUARE:
                Log.d("canvas", "draw rectangle");
                canvas.drawRect(position.X, position.Y, position.X + 200,
position.Y + 200, p);
                break;
            case RECTANGLE:
                Log.d("canvas", "draw square");
                canvas.drawRect(position.X, position.Y, position.X + 150,
position.Y + 200, p);
                break;
        }
    }

}
```

Результат



Вывод

Приобретены практические навыки создания графических интерфейсов на основе различных виджетов и меню.