



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №2

по дисциплине «Разработка мобильных приложений»

Выполнили:

Студенты группы ИКБО-12-22

Солобай А.П.

Проверил:

Преподаватель

Степанов П.В.

Москва 2024 г.

Выполнение практической работы

Ссылка на GitHub со всеми работами: <https://github.com/Eckorezze/Mobile-dev/tree/master>

1. Обеспечение поддержки различных языков. Создание директорий для и файлов strings.xml в них.

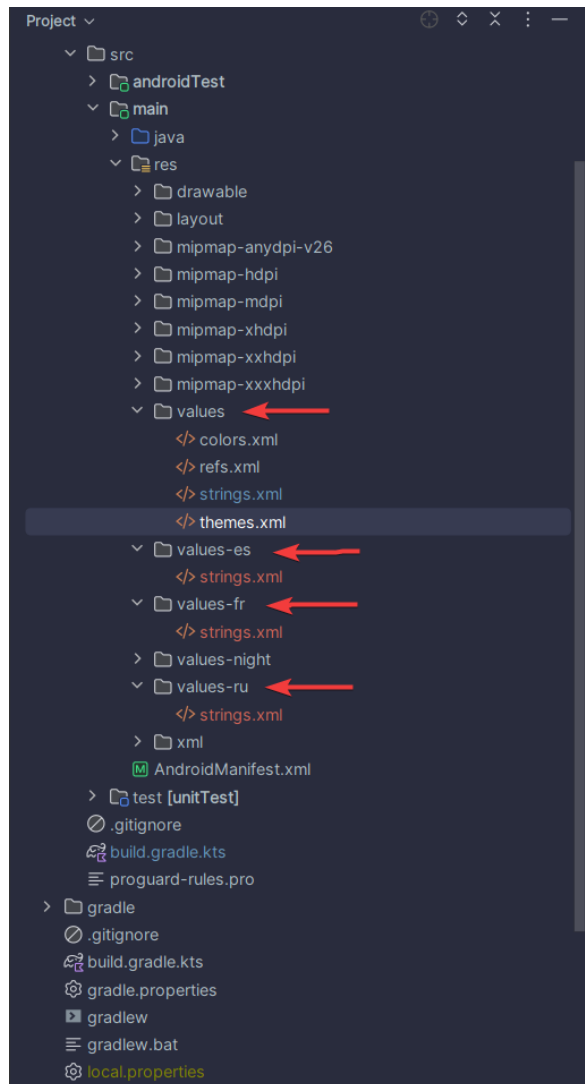


Рисунок 1 – Директории для поддержки английского (по умолчанию), русского, испанского и французского языков.

Листинг 1 – содержимое файла /res/values/strings.xml (поддержка английского языка)

```
<resources>
    <string name="app_name">SolobayAP</string>
    <string name="message">1 practice\n</string>
    <string name="myimage">My image</string>
    <string name="hello_hi">Hello, hi</string>
    <string name="togglebutton">ToggleButton</string>
    <string name="toggle_button">Off</string>
    <string name="toggle_buttonon">On</string>
    <string name="Hint">Input text</string>
    <string name="Test">Test</string>
    <string name="TestMessage">Test message</string>
</resources>
```

Листинг 2 – содержимое файла /res/values-ru/strings.xml (поддержка русского языка)

```
<resources>
    <string name="app_name"><span lang="ru">СолобайАП</span></string>
    <string name="message">1 <span
lang="ru">практика</span>\n</string>
    <string name="myimage"><span lang="ru">Мое
изображение</span></string>
    <string name="hello_hi"><span lang="ru">Привет,
привет</span></string>
    <string name="togglebutton"><span
lang="ru">Переключатель</span></string>
    <string name="toggle_button"><span lang="ru">Выкл</span></string>
    <string name="toggle_buttonon"><span
lang="ru">Вкл</span></string>
    <string name="Hint"><span lang="ru">Введите текст</span></string>
    <string name="Test"><span lang="ru">Тест</span></string>
    <string name="TestMessage"><span lang="ru">Тестовое
сообщение</span></string>
</resources>
```

Листинг 3 – содержимое файла /res/values-es/strings.xml (поддержка испанского языка)

```
<resources>
    <string name="app_name">SolobayAP</string>
    <string name="message">1 <span
lang="es">práctica</span>\n</string>
    <string name="myimage"><span lang="es">Mi imagen</span></string>
    <string name="hello_hi"><span lang="es">Hola,
hola</span></string>
    <string name="togglebutton"><span lang="es">Botón de
alternancia</span></string>
    <string name="toggle_button"><span
lang="es">Apagado</span></string>
    <string name="toggle_buttonon"><span
lang="es">Encendido</span></string>
    <string name="Hint"><span lang="es">Introduzca el
texto</span></string>
    <string name="Test"><span lang="es">Test</span></string>
    <string name="TestMessage"><span lang="es">Mensaje de
prueba</span></string>
</resources>
```

Листинг 4 – содержимое файла /res/values-fr/strings.xml (поддержка французского языка)

```
<resources>
    <string name="app_name"><span lang="fr">SolobaiAP</span></string>
    <string name="message">1 <span
lang="fr">pratique</span>\n</string>
    <string name="myimage"><span lang="fr">Mon image</span></string>
    <string name="hello_hi"><span lang="fr">Salut,
salut</span></string>
    <string name="togglebutton"><span lang="fr">Bouton
bascule</span></string>
    <string name="toggle_button"><span
lang="fr">Éteint</span></string>
    <string name="toggle_buttonon"><span
lang="fr">Allumé</span></string>
    <string name="Hint"><span lang="fr">Entrez le
texte</span></string>
    <string name="Test"><span lang="fr">Test</span></string>
    <string name="TestMessage"><span lang="fr">Message de
test</span></string>
</resources>
```

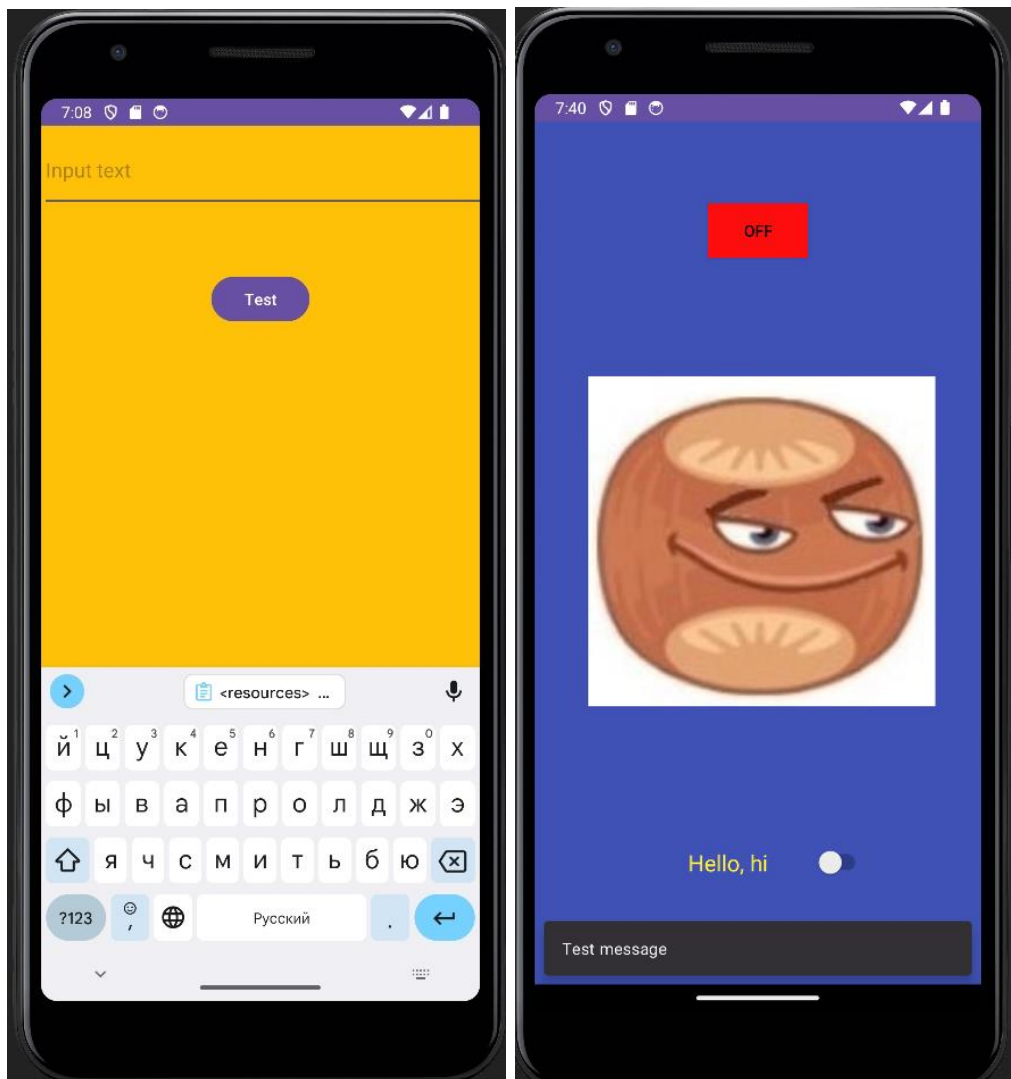


Рисунок 2,3 – Проверка локализации для английского языка.

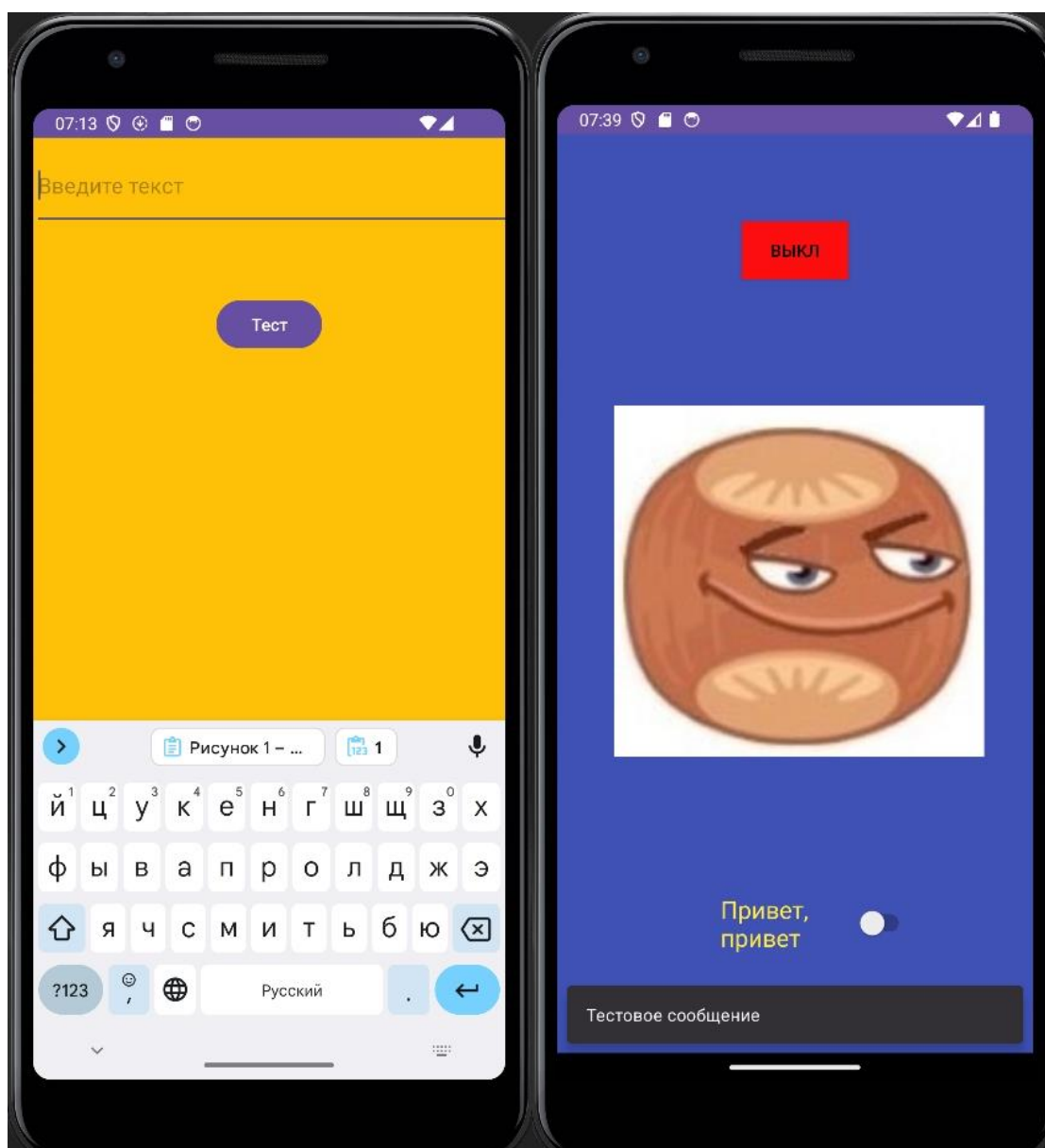


Рисунок 4,5 – Проверка локализации для русского языка.

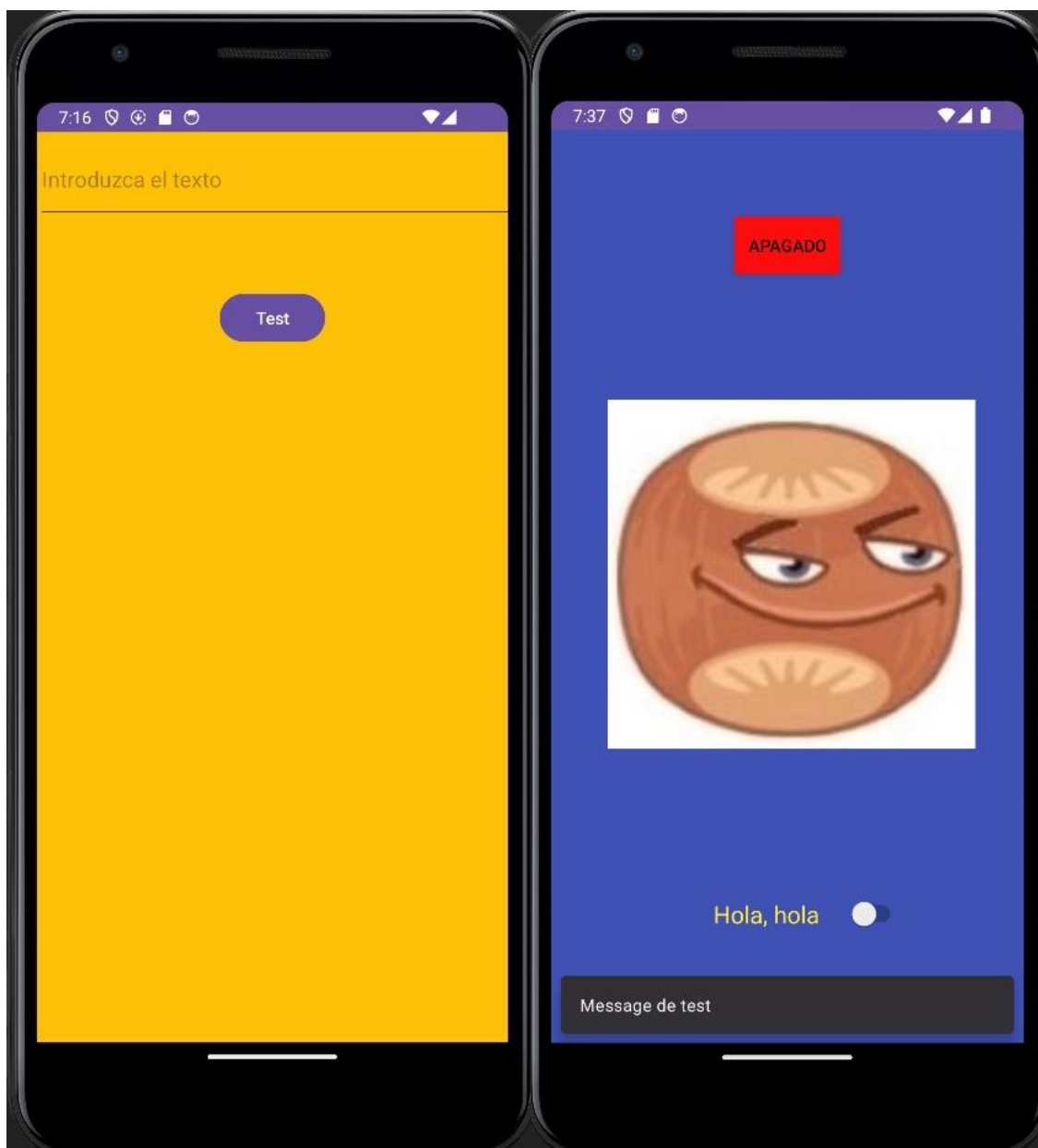


Рисунок 6,7 – Проверка локализации для испанского языка.

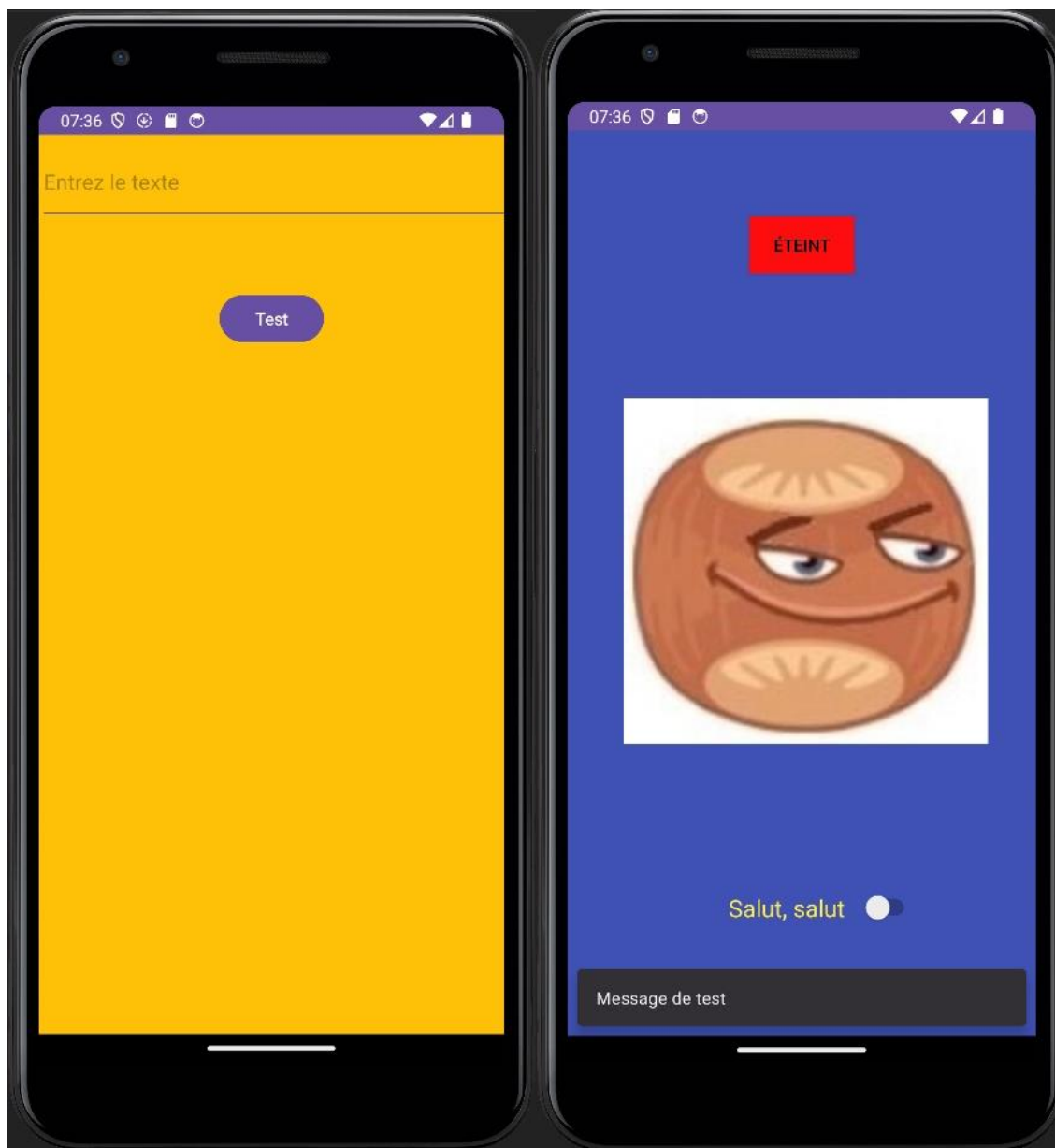


Рисунок 8,9 – Проверка локализации для французского языка.

2. Обеспечение различной разметки для поддержки устройств с различными экранами. Обеспечение использования различных изображений.

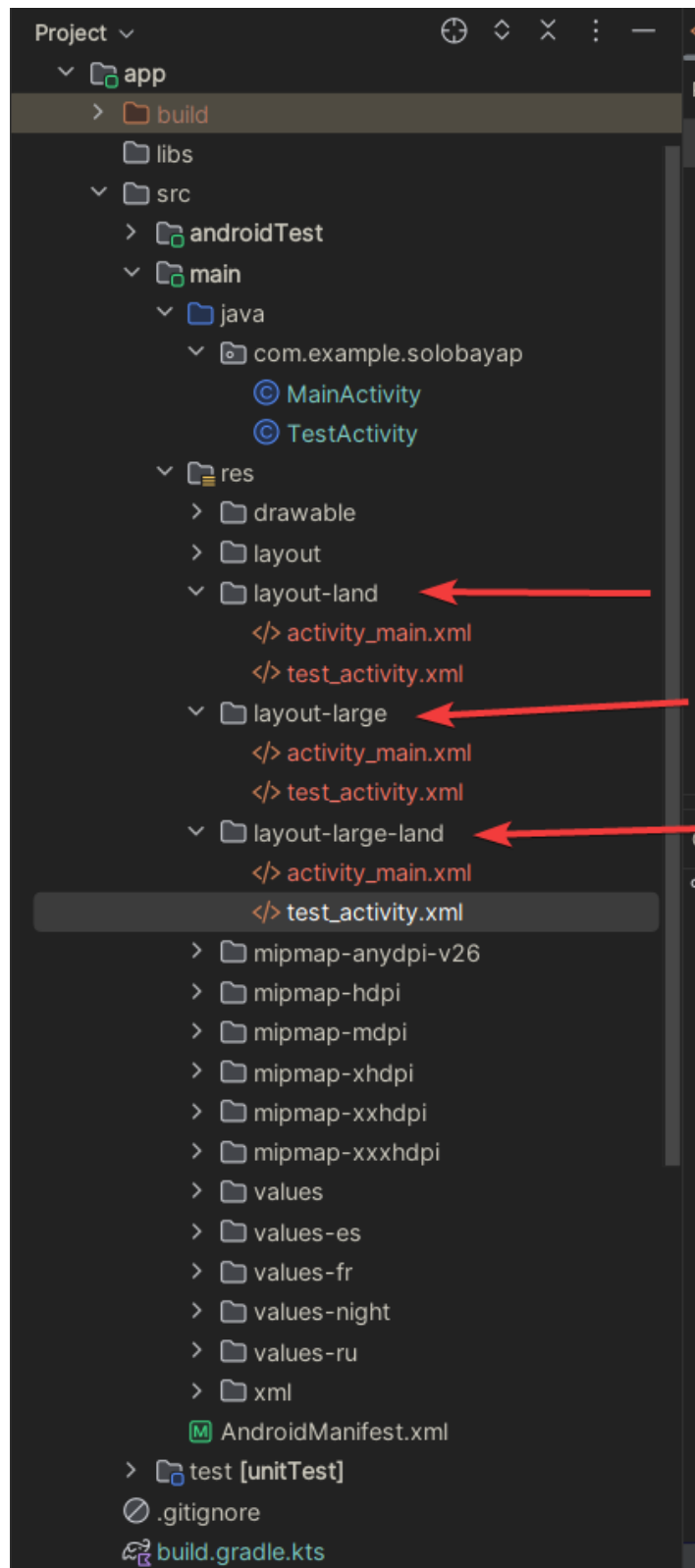


Рисунок 10 – Создание директорий для разных разметок и файлов разметок.

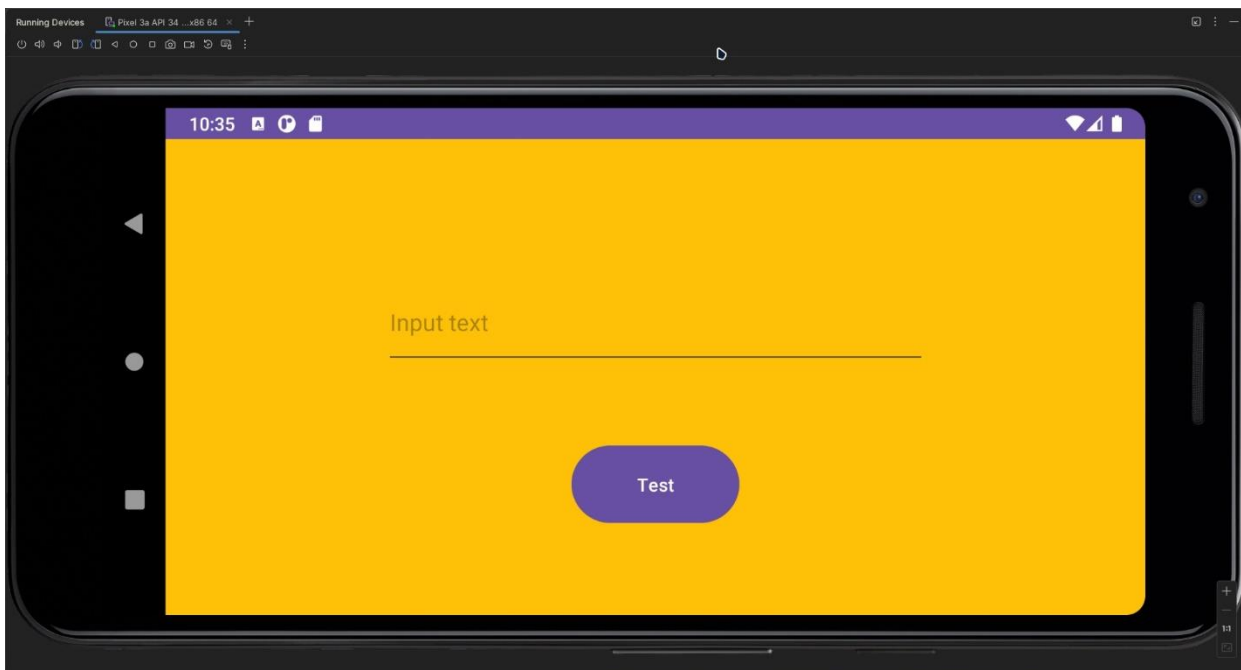


Рисунок 11 – Тестирование альбомной ориентации (Активность 1)

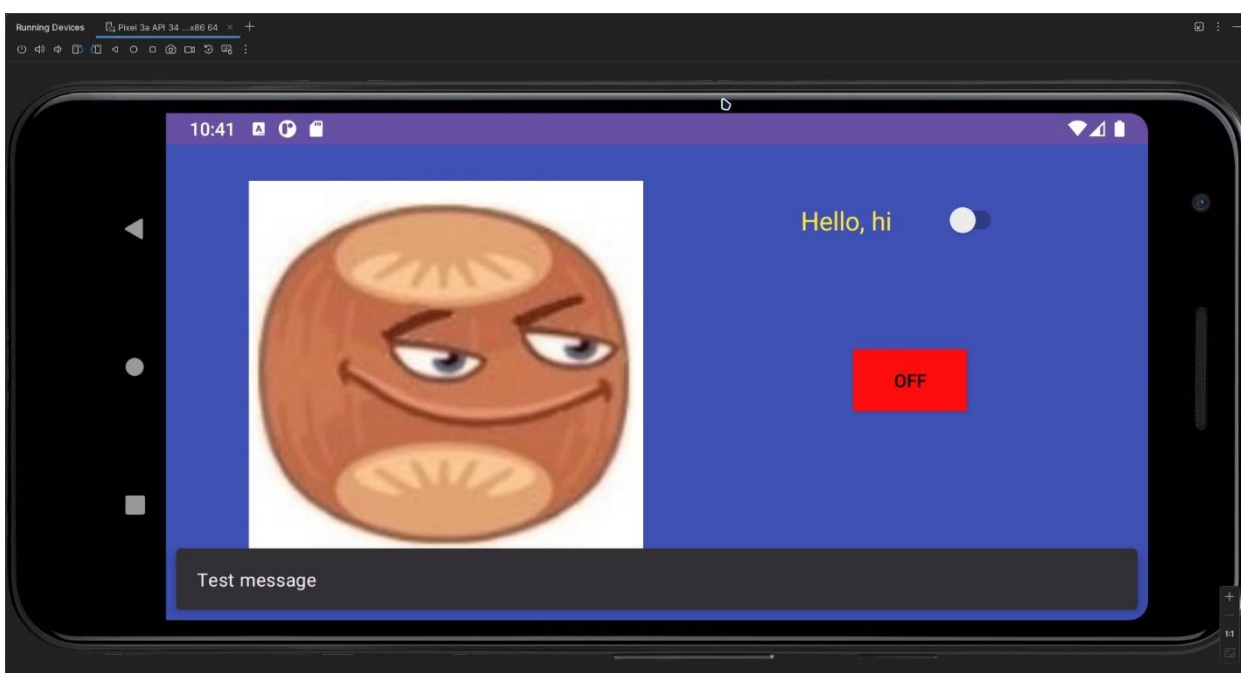


Рисунок 12 – Тестирование альбомной ориентации (Активность 2)

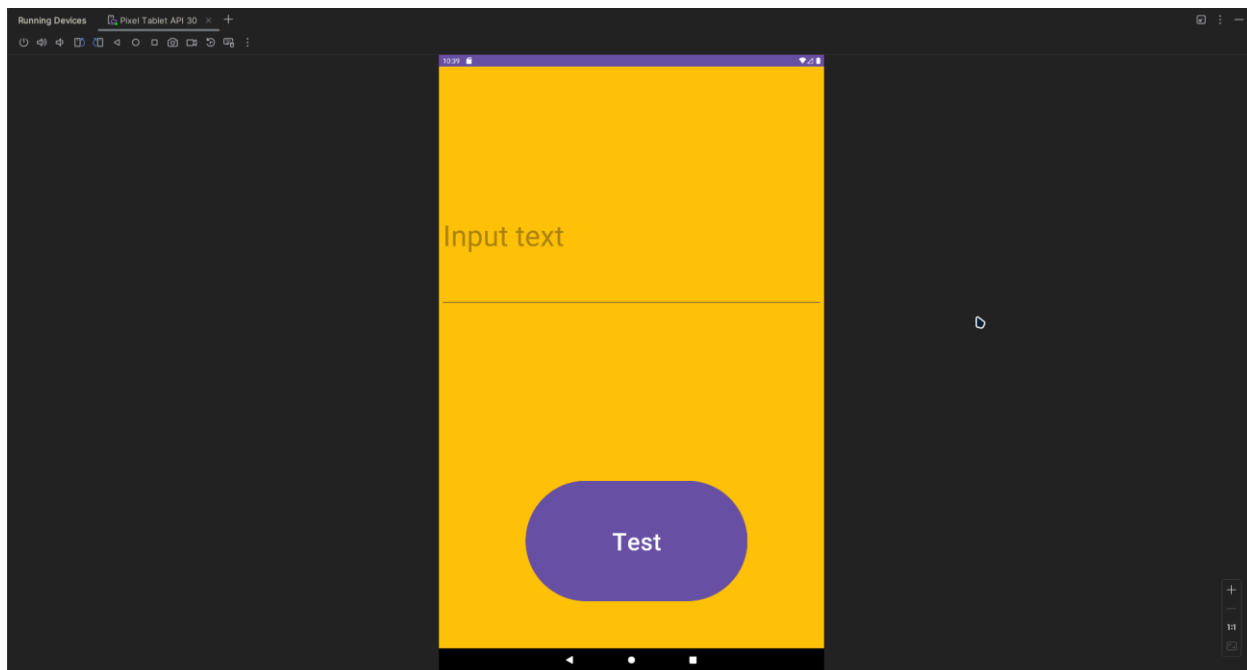


Рисунок 13 – Тестирование большого экрана (Активность 1)

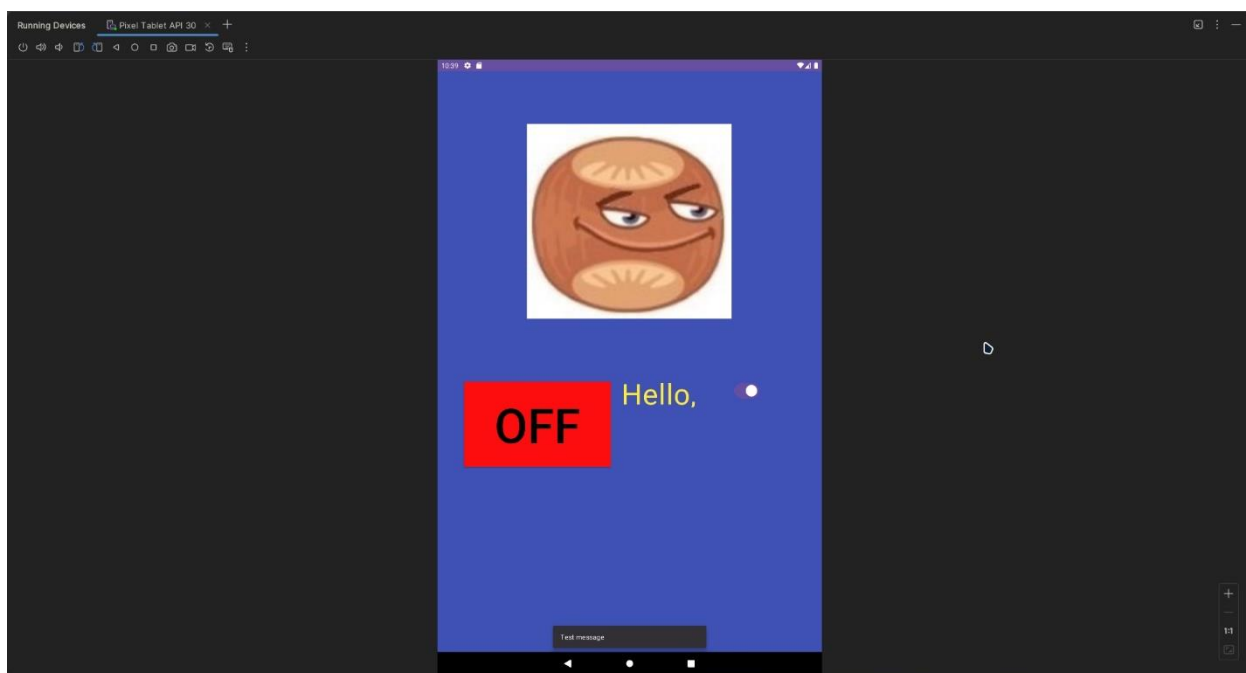


Рисунок 14 – Тестирование большого экрана (Активность 2)

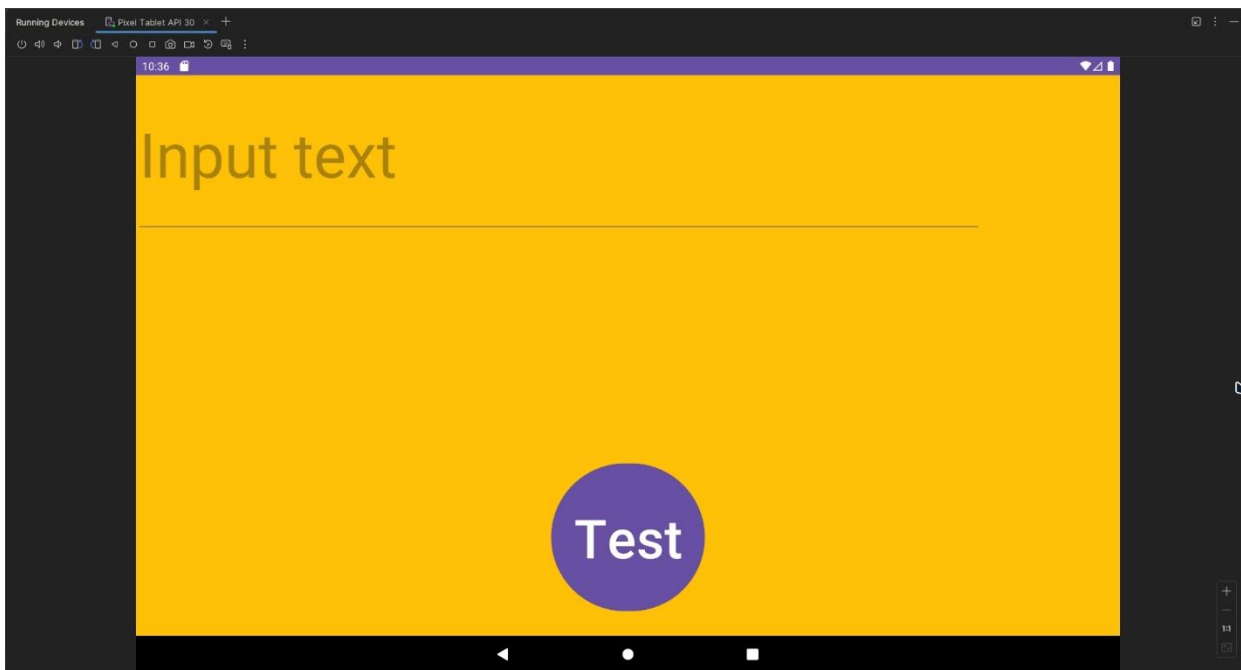


Рисунок 15 – Тестирование альбомной ориентации большого экрана
(Активность 1)

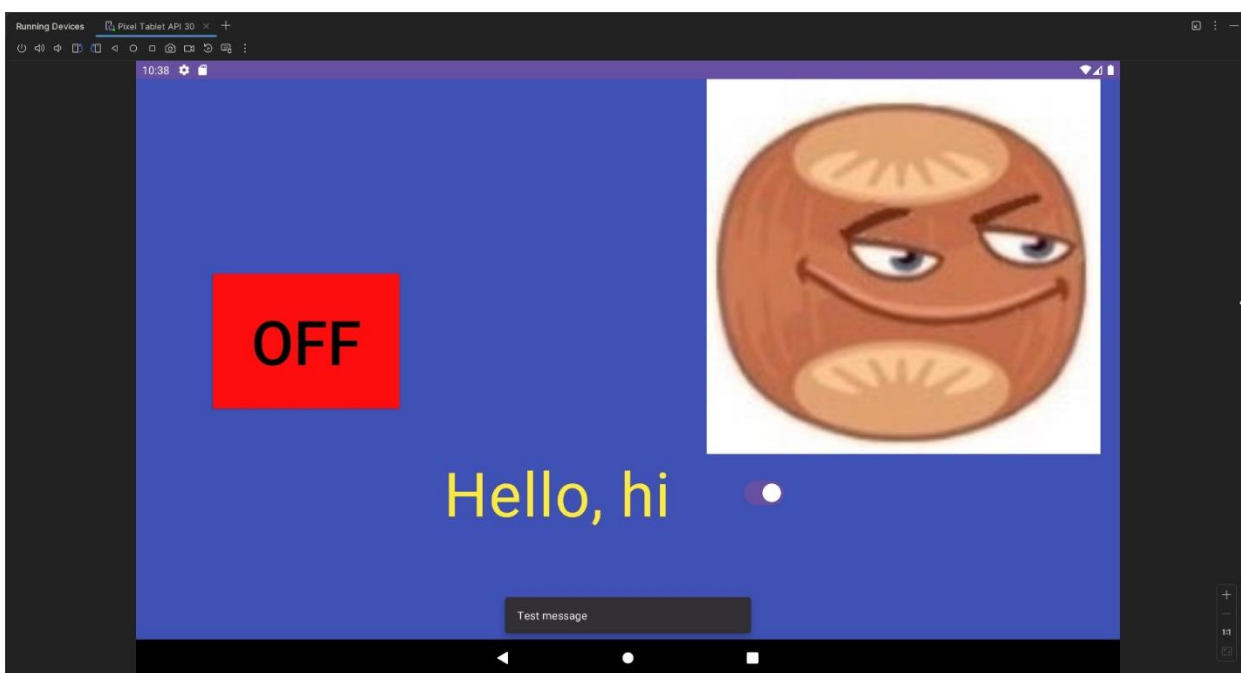


Рисунок 16 – Тестирование альбомной ориентации большого экрана
(Активность 2)

Создадим растровые изображения для каждого из основных разрешений: низкого, среднего, высокого и очень высокого. Это позволит добиться отличного качества и производительности на устройствах с любым разрешением.

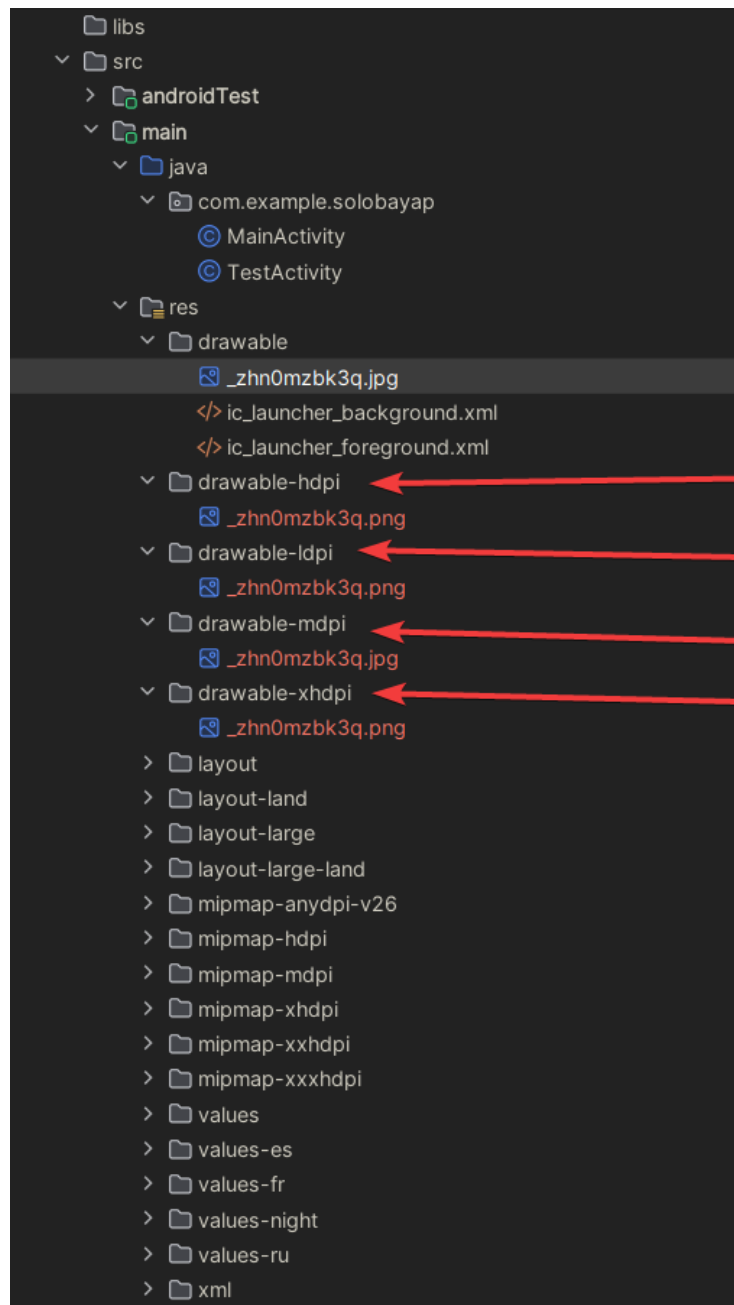


Рисунок 17 – Папки для разных разрешений (ldpi: 0.75, mdpi: 1.0 - базовый размер, hdpi: 1.5, xhdpi: 2.0)



Flavor/Library	Configuration	Value
main (SolobayAP.app)	drawable-ldpi	 <p>124×125 px (165×166 dp @ ldpi) @drawable/_zhn0mzbk3q => _zhn0mzbk3q.png</p>
	drawable	 <p>165×157 px (165×157 dp @ mdpi)</p>

Рисунок 18 – Изображения формата Ідрі и по умолчанию.

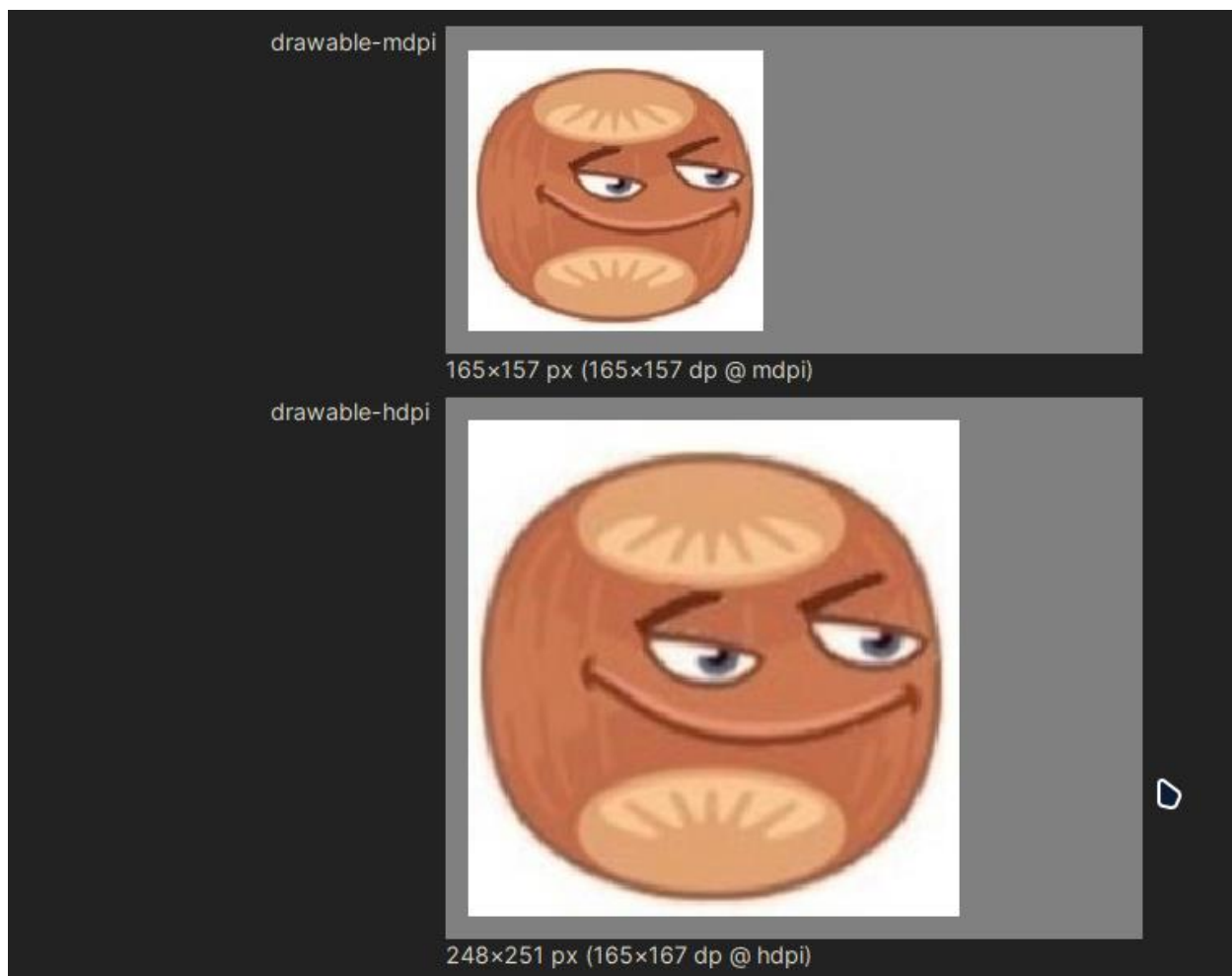


Рисунок 18 – Изображения формата mdpi (базовый размер) и hdpi



Рисунок 19 – Изображение формата xhdpi

3. Поддержка различных версий Android.

Листинг 5 – Указание минимальной и целевой версий API в файле AndroidManifest.xml

```
<uses-sdk android:minSdkVersion="24" android:targetSdkVersion="34"
          tools:ignore="GradleOverrides" />
```

Листинг 6 – получение версии Android и версии SDK во время работы приложения. Код метода onCreate с внесенными изменениями.

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btn=findViewById(R.id.button);

    int sdkVersion = Build.VERSION.SDK_INT;
    String versionName = Build.VERSION.RELEASE;

    final String android = "Android Version:" + versionName + " "
+ "SDK Version: " + sdkVersion;

    btn.setOnClickListener(new View.OnClickListener() {
        final String test_message =
getResources().getString(R.string.TestMessage);
```

```
        @Override
        public void onClick(View v) {

            // Создаем явное намерение
            Intent intent = new Intent(MainActivity.this,
TestActivity.class);
            intent.putExtra(KEY, android);

            // Запуск другой активности
            startActivity(intent);
        }
    });
}
```




Рисунок 18 – Тестирование кода.

4. Обеспечение использования встроенных тем и стилей.

Листинг 7 – содержимое файла themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.SolobayAP"
parent="Theme.Material3.DynamicColors.Dark">
    </style>
    <style name="Theme.SolobayAP" parent="Base.Theme.SolobayAP" />
</resources>
```

Листинг 8 – добавление стиля в AndroidManifest.xml

```
<application
android:theme="@style/Theme.SolobayAP"
</application>
```

5. Жизненный цикл явлений.

Листинг 9 – объявление главного явления в AndroidManifest.xml. В Android Studio прописывается автоматически.

```
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

Листинг 10 – Файл MainActivity.java с методами по обеспечению: запуска явления, функции обратного вызова жизненного цикла, объявления главного явления, создания экземпляра явления, функций приостановки и возобновления явления, функции остановки, функций запуска и перезапуска явлений, функции пересоздания явлений, функции сохранения состояния экземпляра явлений, функции восстановления состояния.

```
package com.example.solobayap;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity{
    private Button btn;
    public static final String KEY = "key";
```

```

private EditText editText;

@Override // Создание активности.
protected void onCreate(Bundle savedInstanceState) {

    //final String test_message =
    getResources().getString(R.string.TestMessage);

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    editText = findViewById(R.id.editText);

    btn=findViewById(R.id.button);

    int sdkVersion = Build.VERSION.SDK_INT;
    String versionName = Build.VERSION.RELEASE;

    final String android = "Android Version:" + versionName + " "
+ "SDK Version: " + sdkVersion;

    Log.d("On create", "onCreate()");

    btn.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {

            onSaveInstanceState(new Bundle());

            String userInput = editText.getText().toString();

            // Создаем явное намерение
            Intent intent = new Intent(MainActivity.this,
TestActivity.class);
            intent.putExtra(KEY, android);

            // Запуск другой активности
            startActivity(intent);
            //finish();

        }
    });

}
//
@Override //Активность становится видимой пользователю
protected void onStart() {
    super.onStart();
    Log.d("On start", "onStart()");
}

@Override // Восстановление состояния активности
protected void onRestoreInstanceState(@NonNull Bundle
savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
}

```

```

        String userInput = savedInstanceState.getString(KEY);
        editText.setText(userInput);
        Log.d("On restore instance", "Restored user input: " +
userInput);
    }

    @Override //Перезапуск активности, но перед тем, как становится
видимой пользователю
    protected void onRestart() {
        super.onRestart();
        Log.d("On restart", "onRestart()");
    }

    @Override //Возобновление активности (видимый)
    protected void onResume() {
        super.onResume();
        Log.d("On resume", "onResume()");
    }

    @Override // Приостановка активности (частично видимый)
    protected void onPause() {
        super.onPause();
        Log.d("On pause", "onPause()");
    }

    @Override // Сохранение состояния активности
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);
        String userInput = editText.getText().toString();
        outState.putString(KEY, userInput);
        Log.d("On save instance", "User input saved: " + userInput);
    }

    @Override //Остановка активности (скрыт)
    protected void onStop() {
        super.onStop();
        Log.d("On stop", "onStop()");
    }
}

```

```

2024-02-14 19:58:41.552 14694-14694 ImeTracker com.example.solobayap i com.example.solobayap:8957864e: onFailed at PHASE_CLIENT_VIEW_SERVED
2024-02-14 19:58:41.897 14694-14694 On stop com.example.solobayap D onStop()
2024-02-14 19:58:41.899 14694-14694 On save instance com.example.solobayap D User input saved: Test text from this field

```

Рисунок 19 – Тестирование работы методов onStop() и onSaveInstanceState()

```

2024-02-14 19:58:44.076 14744-14744 On create com.example.solobayap D onCreate()
2024-02-14 19:58:44.131 14744-14744 On start com.example.solobayap D onStart()
2024-02-14 19:58:44.139 14744-14744 On restore instance com.example.solobayap D Restored user input: Test text from this field
2024-02-14 19:58:44.140 14744-14744 On resume com.example.solobayap D onResume()

```

Рисунок 20 – Тестирование работы методов onCreate(), onStart(),
onRestoreInstanceState(), onResume()

```

2024-02-14 19:58:41.005 14694-14694 On pause com.example.solobayap D onPause()

```

Рисунок 21 – Тестирование работы метода onPause()

2024-02-14 19:42:57.382 14173-14173 On restart

com.example.solobayap

D onRestart()

Рисунок 22 – Тестирование работы метода onRestart()

Вывод

В процессе разработки мобильного приложения были выполнены следующие задачи:

Обеспечена поддержка различных языков путем создания директорий для локализации и использования строковых ресурсов, что позволяет приложению быть доступным для аудитории разных языковых групп.

Реализована поддержка устройств с различными экранами через создание различной разметки и использование адаптивных изображений, обеспечивая приятный пользовательский опыт на устройствах с разными размерами экранов.

Обеспечена совместимость с различными версиями Android, указав минимальную и целевую версии API, а также реализовав получение версии Android во время выполнения приложения, что позволяет приложению работать на широком спектре устройств.

Гарантировано использование встроенных тем и стилей, обеспечивая единый и стандартный внешний вид приложения в соответствии с дизайнерскими рекомендациями.

Реализован полный жизненный цикл явлений, включая запуск, функции обратного вызова, объявление, создание, приостановку, возобновление, остановку, запуск, перезапуск, пересоздание, сохранение и восстановление состояний экземпляров явлений, что обеспечивает корректную работу приложения и сохранение данных пользователя.