ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | С.В. Щекин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 |
| Использование геолокационных сервисов |
| по курсу: Программирование мобильных устройств |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Д.В. Самарин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

Задание:

Организовать навигацию между несколькими Activity (не менее 3-x) при помощи обработки нажатий на интерфейсные элементы, выполнить сборку, отладку и запуск приложения. Навигация должна позволять возможность перехода из любой экранной формы (Activity) в любую другую, возможны различные способы организации (переход из основной и обратно, прямой переход из каждой в каждую, циклический переход по цепочке) Описать результаты в отчете (в т.ч. исх. тексты java и xml для activity).

Цель работы:

Получение навыков организации навигации в многоэкранных мобильных приложениях

Ход работы:

В ходе работы были реализованы несколько Activity.

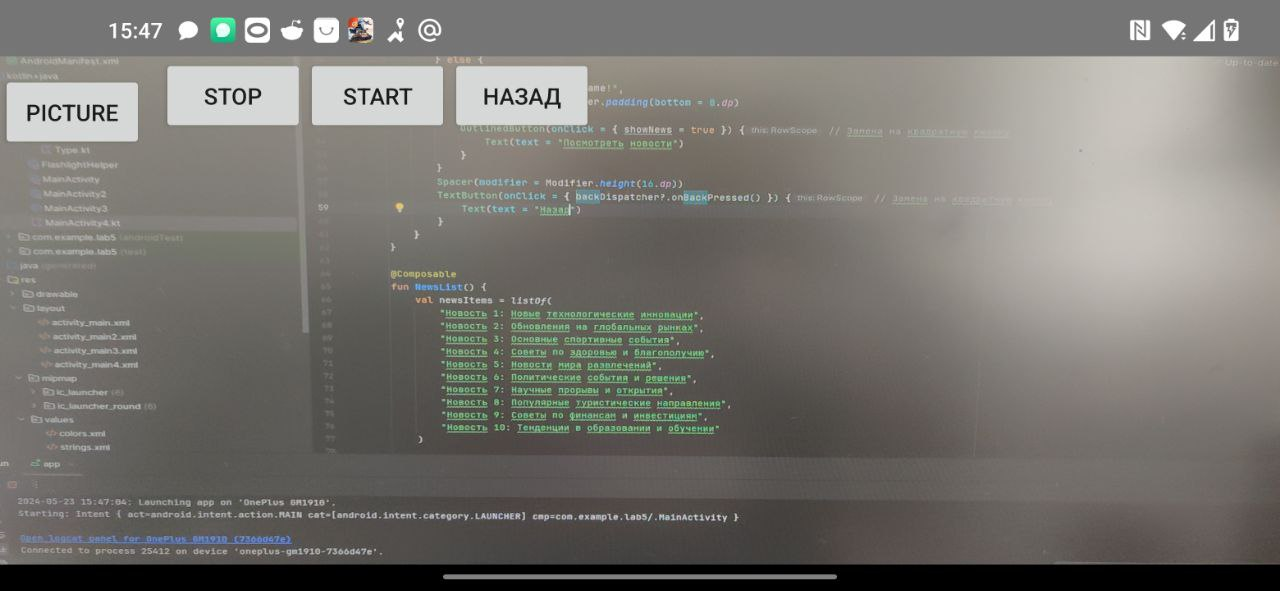
Главная страница:



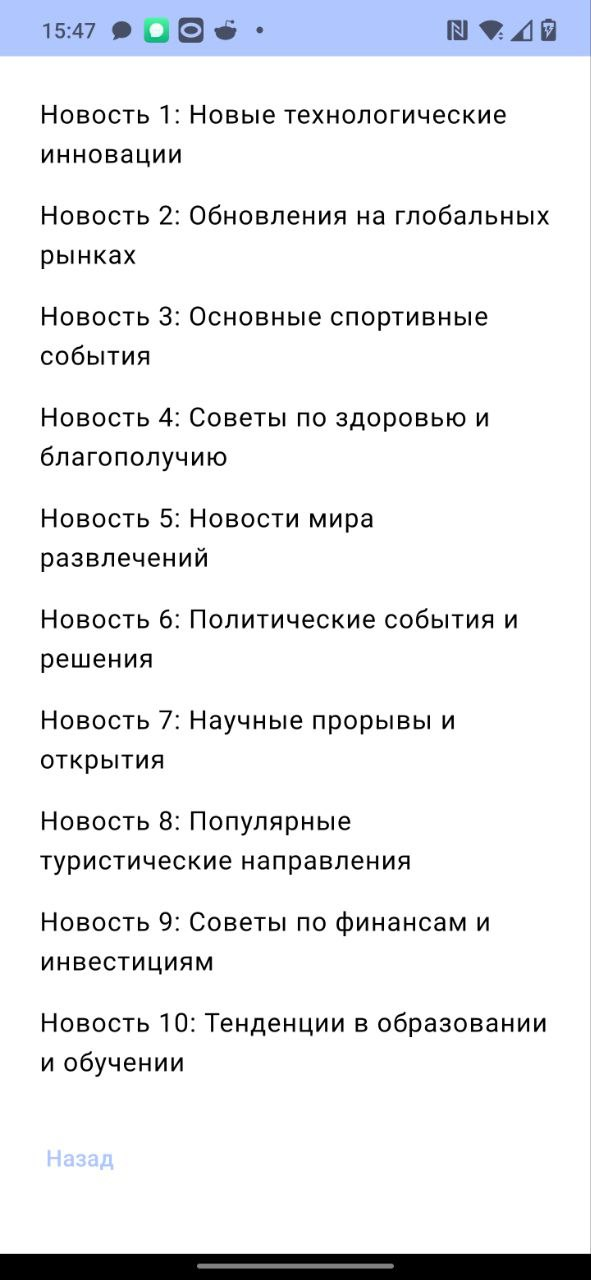
Страница фонарика:



Страница камеры:



Страница новостей:



Листинг

|  |
| --- |
| FlashlightHelper  package com.example.lab5  import android.content.Context import android.hardware.camera2.CameraManager  class FlashlightHelper(private val context: Context) {   private val cameraManager = context.getSystemService(Context.*CAMERA\_SERVICE*) as CameraManager   fun turnOn() {  val cameraId = cameraManager.*cameraIdList*[0]  cameraManager.setTorchMode(cameraId, true)  }   fun turnOff() {  val cameraId = cameraManager.*cameraIdList*[0]  cameraManager.setTorchMode(cameraId, false)  } }  MainActivity  package com.example.lab5  import android.content.Intent import android.os.Bundle import android.widget.Button import android.widget.TextView import androidx.activity.ComponentActivity import androidx.activity.compose.setContent import androidx.appcompat.app.AppCompatActivity import androidx.compose.foundation.layout.fillMaxSize import androidx.compose.material3.MaterialTheme import androidx.compose.material3.Surface import androidx.compose.material3.Text import androidx.compose.runtime.Composable import androidx.compose.ui.Modifier import androidx.compose.ui.tooling.preview.Preview import com.example.lab5.ui.theme.Lab5Theme  class MainActivity : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  findViewById<Button>(R.id.*buttonFlahlight*).setOnClickListener**{** val intent = Intent(this, MainActivity2::class.*java*)  startActivity(intent)  **}** findViewById<Button>(R.id.*buttonCamera*).setOnClickListener**{** val intent = Intent(this, MainActivity3::class.*java*)  startActivity(intent)  **}** findViewById<Button>(R.id.*buttonMap*).setOnClickListener **{** val intent = Intent(this, MainActivity4::class.*java*)  startActivity(intent)  **}** findViewById<Button>(R.id.*exitButton*).setOnClickListener **{** finish()  **}** } }  MainActivity2  package com.example.lab5 import com.example.lab5.FlashlightHelper import android.content.Intent import android.os.Bundle import android.widget.Button import android.widget.ToggleButton import androidx.activity.ComponentActivity import androidx.activity.compose.setContent import androidx.compose.foundation.layout.fillMaxSize import androidx.compose.material3.MaterialTheme import androidx.compose.material3.Surface import androidx.compose.material3.Text import androidx.compose.runtime.Composable import androidx.compose.ui.Modifier import androidx.compose.ui.tooling.preview.Preview import com.example.lab5.ui.theme.Lab5Theme   class MainActivity2 : ComponentActivity() {   private lateinit var flashlightHelper: FlashlightHelper   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main2*)   findViewById<Button>(R.id.*buttonBack*).setOnClickListener **{** finish()  **}** findViewById<ToggleButton>(R.id.*button4*).setOnClickListener **{** if ((**it** as ToggleButton).*isChecked*) {  flashlightHelper.turnOn()  } else {  flashlightHelper.turnOff()  }  **}** flashlightHelper = FlashlightHelper(this)  }   override fun onDestroy() {  super.onDestroy()   flashlightHelper.turnOff()  } }  MainActivity3  package com.example.lab5;  import android.app.Activity; import android.hardware.Camera; import android.media.CamcorderProfile; import android.media.MediaRecorder; import android.os.Bundle; import android.os.Environment; import android.view.SurfaceHolder; import android.view.SurfaceView; import android.view.View; import android.view.View; import android.widget.Button;  import com.example.lab5.R;  import java.io.File; import java.io.FileOutputStream;  public class MainActivity3 extends Activity {  SurfaceView surfaceView;  Camera camera;  MediaRecorder mediaRecorder;  File photoFile;  File videoFile;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main3*);  Button buttonBack = findViewById(R.id.*buttonBack*);  buttonBack.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  finish();  }  });  File pictures = Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_PICTURES*);  photoFile = new File(pictures, "myphoto.jpg");  videoFile = new File(pictures, "myvideo.3gp");   surfaceView = findViewById(R.id.*surfaceView*);  SurfaceHolder holder = surfaceView.getHolder();  holder.addCallback(new SurfaceHolder.Callback() {  @Override  public void surfaceCreated(SurfaceHolder holder) {  try {  camera.setPreviewDisplay(holder);  camera.startPreview();  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  public void surfaceChanged(SurfaceHolder holder, int format,  int width, int height) {  }   @Override  public void surfaceDestroyed(SurfaceHolder holder) {  }  });  }   @Override  protected void onResume() {  super.onResume();  camera = Camera.*open*();  }   @Override  protected void onPause() {  super.onPause();  releaseMediaRecorder();  if (camera != null)  camera.release();  camera = null;  }   public void onClickPicture(View view) {  camera.takePicture(null, null, new Camera.PictureCallback() {  @Override  public void onPictureTaken(byte[] data, Camera camera) {  try {  FileOutputStream fos = new FileOutputStream(photoFile);  fos.write(data);  fos.close();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }   public void onClickStartRecord(View view) {  if (prepareVideoRecorder()) {  mediaRecorder.start();  } else {  releaseMediaRecorder();  }  }   public void onClickStopRecord(View view) {  if (mediaRecorder != null) {  mediaRecorder.stop();  releaseMediaRecorder();  }  }   private boolean prepareVideoRecorder() {  camera.unlock();  mediaRecorder = new MediaRecorder();  mediaRecorder.setCamera(camera);  mediaRecorder.setAudioSource(MediaRecorder.AudioSource.*CAMCORDER*);  mediaRecorder.setVideoSource(MediaRecorder.VideoSource.*CAMERA*);  mediaRecorder.setProfile(CamcorderProfile.*get*(CamcorderProfile.*QUALITY\_HIGH*));  mediaRecorder.setOutputFile(videoFile.getAbsolutePath());  mediaRecorder.setPreviewDisplay(surfaceView.getHolder().getSurface());  try {  mediaRecorder.prepare();  } catch (Exception e) {  e.printStackTrace();  releaseMediaRecorder();  return false;  }  return true;  }   private void releaseMediaRecorder() {  if (mediaRecorder != null) {  mediaRecorder.reset();  mediaRecorder.release();  mediaRecorder = null;  camera.lock();  }  } }  MainActivity4  package com.example.lab5  import android.os.Bundle import androidx.activity.ComponentActivity import androidx.activity.compose.setContent import androidx.activity.compose.LocalOnBackPressedDispatcherOwner import androidx.compose.foundation.layout.\* import androidx.compose.foundation.lazy.LazyColumn import androidx.compose.foundation.lazy.items import androidx.compose.material3.\* import androidx.compose.runtime.\* import androidx.compose.ui.Modifier import androidx.compose.ui.graphics.Color import androidx.compose.ui.tooling.preview.Preview import androidx.compose.ui.unit.dp import com.example.lab5.ui.theme.Lab5Theme  class MainActivity4 : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  *setContent* **{** Lab5Theme **{** // A surface container using the 'background' color from the theme  Surface(  modifier = Modifier.*fillMaxSize*(),  color = Color.White // Установка белого цвета фона  ) **{** GreetingWithButton("Дмитрий")  **}  }  }** } }  @Composable fun GreetingWithButton(name: String) {  var showNews by remember **{** *mutableStateOf*(false) **}** val backDispatcher = LocalOnBackPressedDispatcherOwner.current?.onBackPressedDispatcher   Column(  modifier = Modifier  .*fillMaxSize*()  .*padding*(16.*dp*),  verticalArrangement = Arrangement.Top  ) **{** if (showNews) {  NewsList()  } else {  Text(  text = "Привет $name!",  modifier = Modifier.*padding*(bottom = 8.*dp*)  )  OutlinedButton(onClick = **{** showNews = true **}**) **{** // Замена на квадратную кнопку  Text(text = "Посмотреть новости")  **}** }  Spacer(modifier = Modifier.*height*(16.*dp*))  TextButton(onClick = **{** backDispatcher?.onBackPressed() **}**) **{** // Замена на квадратную кнопку  Text(text = "Назад")  **}  }** }  @Composable fun NewsList() {  val newsItems = *listOf*(  "Новость 1: Новые технологические инновации",  "Новость 2: Обновления на глобальных рынках",  "Новость 3: Основные спортивные события",  "Новость 4: Советы по здоровью и благополучию",  "Новость 5: Новости мира развлечений",  "Новость 6: Политические события и решения",  "Новость 7: Научные прорывы и открытия",  "Новость 8: Популярные туристические направления",  "Новость 9: Советы по финансам и инвестициям",  "Новость 10: Тенденции в образовании и обучении"  )   LazyColumn **{** *items*(newsItems) **{** newsItem **->** Text(text = newsItem, modifier = Modifier.*padding*(8.*dp*))  **}  }** }  @Preview(showBackground = true) @Composable fun GreetingWithButtonPreview() {  Lab5Theme **{** GreetingWithButton("Дмитрий")  **}** } |