Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА

по дисциплине «Программирование мобильных устройств»

Студент группы 7ИСб-1

Преподаватель В.А. Тихомиров

2021

**Задания**

Разработать видеопроигрыватель (Видео.exe).

**Содержание**

[Введение 4](#_Toc35433989)

[Ход выполнения работы 5](#_Toc35433991)

[Заключение 12](#_Toc35433992)

[Список использованных источников 13](#_Toc35433993)

Введение

Android Studio - это [интегрированная среда разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) (IDE) для работы с платформой [Android](https://ru.wikipedia.org/wiki/Android" \o "Android). [1]

Android Studio, основанная на программном обеспечении [IntelliJ IDEA](https://ru.wikipedia.org/wiki/IntelliJ_IDEA" \o "IntelliJ IDEA) от компании [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains), - официальное средство разработки Android приложений. Данная среда разработки доступна для [Windows](https://ru.wikipedia.org/wiki/Windows" \o "), [OS X](https://ru.wikipedia.org/wiki/OS_X) и Linux.

17 мая 2017 на ежегодной конференции Google I/O, [Google](https://ru.wikipedia.org/wiki/Google_(%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F)" \o "Google (компания)) анонсировал поддержку языка [Kotlin](https://ru.wikipedia.org/wiki/Kotlin" \o "Kotlin), используемого в Android Studio, как официального языка программирования для платформы Android в добавление к [Java](https://ru.wikipedia.org/wiki/Java" \o "Java) и [С++](https://ru.wikipedia.org/wiki/C%2B%2B).

Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

* Расширенный редактор макетов: [WYSIWYG](https://ru.wikipedia.org/wiki/WYSIWYG), способность работать с [UI](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) компонентами при помощи [Drag-and-Drop](https://ru.wikipedia.org/wiki/Drag-and-drop" \o "Drag-and-drop), функция предпросмотра макета на нескольких конфигурациях экрана.
* Сборка приложений, основанная на [Gradle](https://ru.wikipedia.org/wiki/Gradle" \o "Gradle).
* Различные виды сборок и генерация нескольких [.apk](https://ru.wikipedia.org/wiki/.APK) файлов
* [Рефакторинг](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3) кода
* [Статический анализатор кода](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80_%D0%BA%D0%BE%D0%B4%D0%B0&action=edit&redlink=1) (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
* Встроенный ProGuard и утилита для подписывания приложений.
* Шаблоны основных макетов и компонентов Android.
* Встроенная поддержка [Google Cloud Platform](https://ru.wikipedia.org/wiki/Google_Cloud_Platform" \o "Google Cloud Platform), которая включает в себя интеграцию с сервисами [Google Cloud Messaging](https://ru.wikipedia.org/wiki/Google_Cloud_Messaging" \o "Google Cloud Messaging) и [App Engine](https://ru.wikipedia.org/w/index.php?title=App_Engine&action=edit&redlink=1" \o "App Engine (страница отсутствует)).
* Новая версия Android Studio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию Instant Run.
* В Android Studio 3.0 будут по стандарту включены инструменты языка [Kotlin](https://ru.wikipedia.org/wiki/Kotlin" \o "Kotlin) основанные на [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains) [IDE](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8). [2]

Ход выполнения работы

При разработке приложения были использованы 2 файла в Android Studio и одно видео, расположенное в папке приложения. Содержимое файла main\_activity.xml приведено в листинге 1.

Листинг 1 - main\_activity.xml

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/trim\_container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** >  
  
 <**LinearLayout  
 android:id="@+id/buttonContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:paddingHorizontal="10dp"**>  
  
 <**Button  
 android:id="@+id/go\_back"  
 android:layout\_width="50dp"  
 android:layout\_height="match\_parent"  
 android:onClick="stop"  
 android:text="⏹"** />  
  
 <**Button  
 android:id="@+id/back\_to\_pick\_song"  
 android:layout\_width="50dp"  
 android:layout\_height="match\_parent"  
 android:onClick="play"  
 android:text="▶️"** />  
  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="50dp"  
 android:layout\_height="wrap\_content"  
 android:onClick="toStart"  
 android:text="⏮"** />  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="50dp"  
 android:layout\_height="wrap\_content"  
 android:onClick="back5sec"  
 android:text="⏪"** />  
  
 <**Button  
 android:id="@+id/button3"  
 android:layout\_width="50dp"  
 android:layout\_height="wrap\_content"  
 android:onClick="next5sec"  
 android:text="⏩"** />  
  
 <**Button  
 android:id="@+id/btn\_continue"  
 android:layout\_width="50dp"  
 android:layout\_height="wrap\_content"  
 android:onClick="toEnd"  
 android:text="⏭"** />  
  
 <**ToggleButton  
 android:id="@+id/toggleButton"  
 android:layout\_width="80dp"  
 android:layout\_height="match\_parent"  
 android:textOff="🔁 OFF"  
 android:textOn="🔁 ON"** />  
  
 <**SeekBar  
 android:id="@+id/seekBar2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
 </**LinearLayout**>  
  
 <**VideoView  
 android:id="@+id/VideoView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_below="@id/buttonContainer"**/>  
</**RelativeLayout**>

Содержимое файла MainActivity.java приведено в листинге 3.

Листинг 3 - MainActivity.java

**package** com.example.rgr;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.media.MediaPlayer;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.CompoundButton;  
**import** android.widget.MediaController;  
**import** android.widget.SeekBar;  
**import** android.widget.ToggleButton;  
**import** android.widget.VideoView;  
  
**public class** MainActivity **extends** AppCompatActivity **implements** CompoundButton.OnCheckedChangeListener {  
 **boolean loopVideo** = **false**;  
 VideoView **videoView**;  
 SeekBar **seekBar**;  
 Handler **updateHandler** = **new** Handler();  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// Установливаем свой путь к файлу* String path = getApplicationInfo().**dataDir** + **"/video.mp4"**;  
 **videoView** = (VideoView) findViewById(R.id.***VideoView***);  
 **videoView**.setVideoPath(path);  
 **videoView**.setMediaController(**new** MediaController(**this**));  
 **videoView**.requestFocus(0);  
 **videoView**.seekTo(1);  
  
 *// Получаем seekbar и устанавливаем обработчик* **seekBar** = (SeekBar) findViewById(R.id.***seekBar2***);  
 **seekBar**.setOnSeekBarChangeListener(**seekBarChangeListener**);  
  
 *// Событие при загрузке видео и готовности к воспроизведению в VideoView* **videoView**.setOnPreparedListener(mediaPlayer -> {  
 **seekBar**.setProgress(0);  
 **seekBar**.setMax(**videoView**.getDuration());  
 **updateHandler**.postDelayed(**updateVideoTime**, 100);  
 });  
  
 *// Получение кнопки переключение и установка обработчика* ToggleButton toogleButton = (ToggleButton) findViewById(R.id.***toggleButton***);  
 toogleButton.setOnCheckedChangeListener(**this**);  
 }  
  
 *// Обновление позиции SeekBar относительно VideoView каждые 100мл* **private** Runnable **updateVideoTime** = **new** Runnable() {  
 @Override  
 **public void** run() {  
 **long** currentPosition = **videoView**.getCurrentPosition();  
 **seekBar**.setProgress((**int**) currentPosition);  
 **updateHandler**.postDelayed(**this**, 100);  
 }  
 };  
  
 *// Обновление позиции VideoView при изменении SeekBar нажатием* **private** SeekBar.OnSeekBarChangeListener **seekBarChangeListener** = **new** SeekBar.OnSeekBarChangeListener() {  
 @Override  
 **public void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromUser) {}  
  
 @Override  
 **public void** onStartTrackingTouch(SeekBar seekBar) {  
 **videoView**.seekTo(seekBar.getProgress());  
 }  
  
 @Override  
 **public void** onStopTrackingTouch(SeekBar seekBar) {  
 **videoView**.seekTo(seekBar.getProgress());  
 }  
 };  
  
 *// События кнопок управления видео* **public void** stop(View view) {  
 **if** (**videoView**.canPause()) {  
 **videoView**.pause();  
 }  
 }  
  
 **public void** play(View view) {  
 **videoView**.start();  
 }  
  
 **public void** toStart(View view) {  
 **videoView**.pause();  
 **videoView**.seekTo(0);  
 }  
  
 **public void** toEnd(View view) {  
 **videoView**.pause();  
 **videoView**.seekTo(**videoView**.getDuration());  
 }  
  
 **public void** back5sec(View view) {  
 **videoView**.seekTo(**videoView**.getCurrentPosition() - 5000);  
 }  
  
 **public void** next5sec(View view) {  
 **videoView**.seekTo(**videoView**.getCurrentPosition() + 5000);  
 }  
  
 *// переключение кнопки репита видео* @Override  
 **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  
 **if** (isChecked) **videoView**.setOnCompletionListener(mp -> **videoView**.start());  
 **else videoView**.setOnCompletionListener(**null**);  
 }  
}

Результат работы приложения показан на рисунках 1-5.

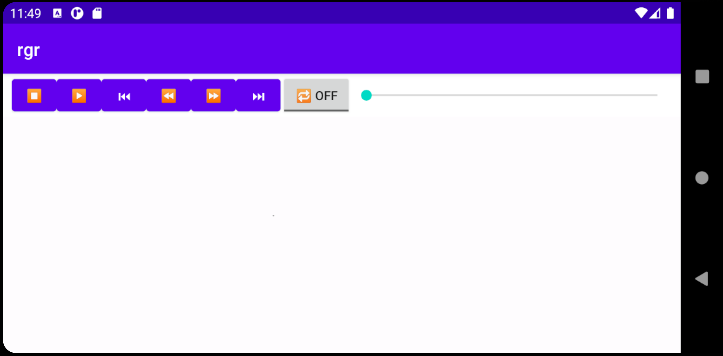


Рисунок 1 – Стартовый экран

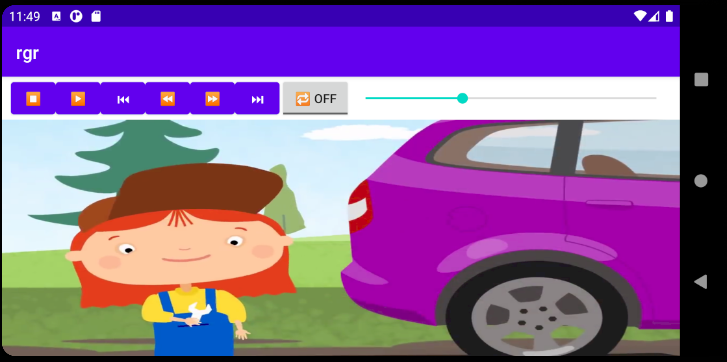


Рисунок 2 – Воспроизведение видео

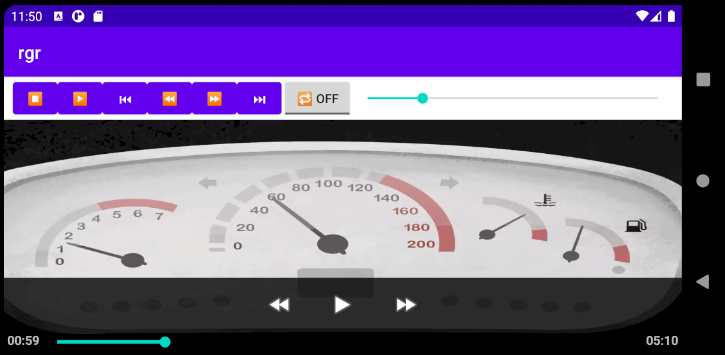


Рисунок 3 – Воспроизведение видео 2

Заключение

В ходе выполнения расчётно-графической работы были изучены и рассмотрены возможности [интегрированной среды разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) «Android Studio», в которой было разработано приложение с выпадающим меню.

Полученные навыки и знания будут использоваться в дальнейшем.

Список использованных источников

1. Дейтел, П. Android для разработчиков. 3-е изд. / П. Дейтел, Х. Дейтел, А. Уолд. — СПб.: Питер, 2016. — 512 с.
2. Гриффитс. Д. Head First. Программирование для Android / Дон Гриффитс, Дэвид Гриффитс — СПб.: Питер, 2016. — 704 с.