



УНИВЕРСИТЕТ ИТМО

Сравнительный анализ библиотек для воспроизведения аудио и видео контента для операционной системы Android

Кадыров Амирджон Халимджонович

Романова Асель

Проблема и актуальность

Проблема:

Android разработчики тратят время на выбор библиотеки для воспроизведения медиа контента при разработке мобильного приложения

Актуальность:

Необходимо ответить на вопрос о том, какую библиотеку использовать в зависимости от требований при разработке приложения



Цель:

Проведение сравнительного анализа библиотек ОС Android для воспроизведения медиа-контента для сокращения времени выбора библиотеки при разработке мобильных приложений.

Задачи:

1. Обзор библиотек ОС Android;
2. Определить критерии сравнения рассматриваемых библиотек;
3. Реализация работы библиотек с аудио и видео контентом;
4. Сравнения библиотек по основным критериям;
5. Выбор наиболее подходящей библиотеки для работы с медиа контентом

Были рассмотрены библиотеки, рекомендованные компанией Google

Библиотека	Компонент для отображения видео	Поддержка технологий адаптивной потоковой передачи данных	Поддерживаемые устройства
MediaPlayer	SurfaceView, VideoView	HLS	Android version 1+ 100% устройств
ExoPlayer	StyledPlayerView	HLS, DASH, SmoothStreaming	Android version 4.1+ 99.8% устройств

Обе библиотеки предоставляют API для работы с медиа-контентом.

Критерии сравнения для эффективной работы плеера:

- Первый кадр. Пользователи не хотят ждать начала воспроизведения
- Переключение качества видео при изменении скорости интернета
- Возможность оптимизации

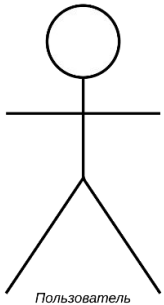
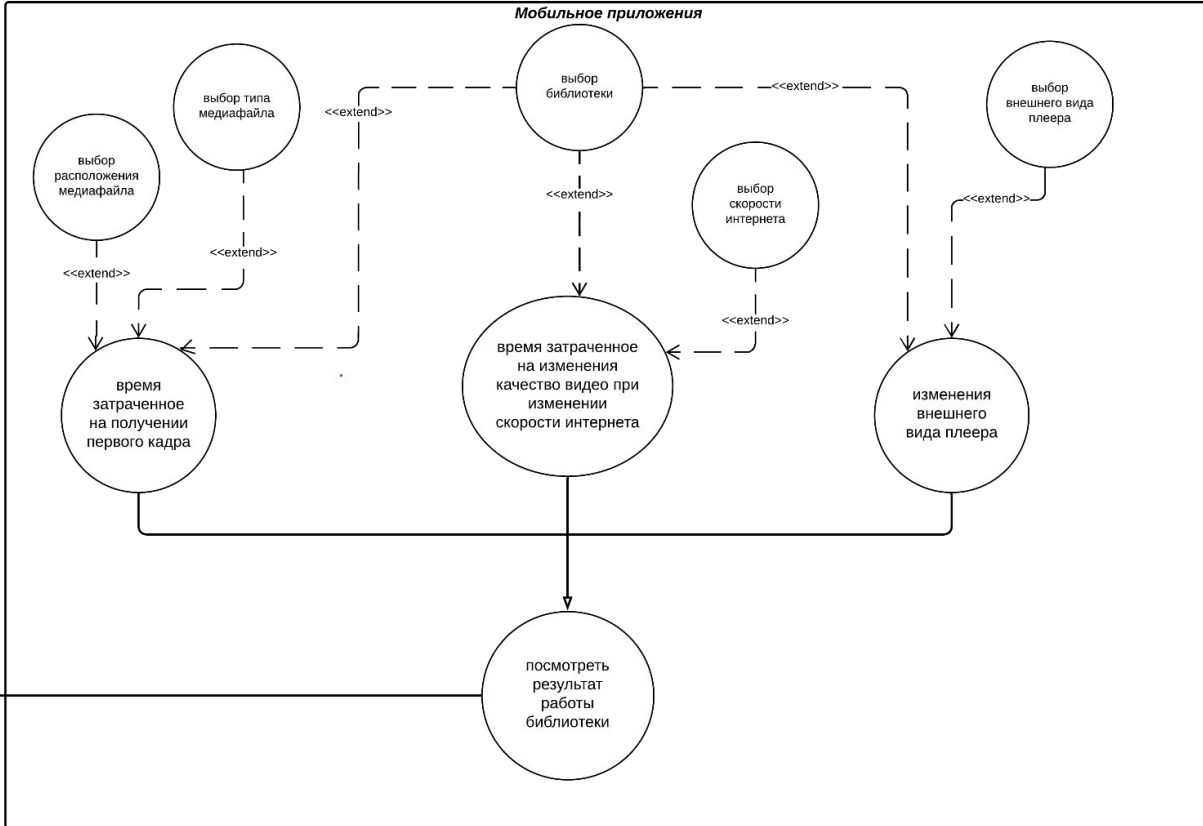
Критерии сравнения пользовательского интерфейса (дополнительные элементы управления):

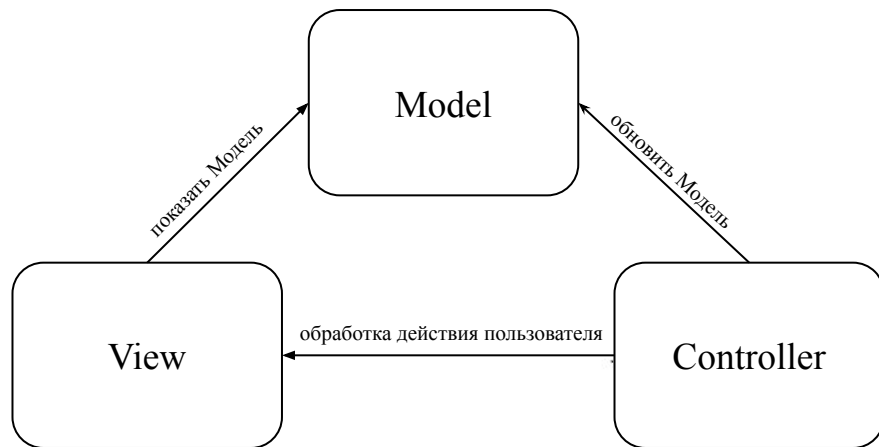
- Перемотка на 10 секунд (двойным нажатием, кнопки)
- Кнопка “Случайный порядок”, “Повтор”
- Предпросмотр контента через бегунок прогресса
- Кнопка блокировки от случайных нажатий на экран
- Кнопка переключения качество видео
- Поворот экрана

Требования к разрабатываемому приложению:

1. Воспроизведение медиафайла из локального хранилища или из сети;
2. Выбор параметров для сравнения:
 - a. параметр выборки библиотеки;
 - b. параметр выборки типа медиафайла;
 - c. параметр выборки местоположения медиафайла;
 - d. параметр выборки внешнего вида плеера
3. Вывод результатов измерений

Use case модель приложения



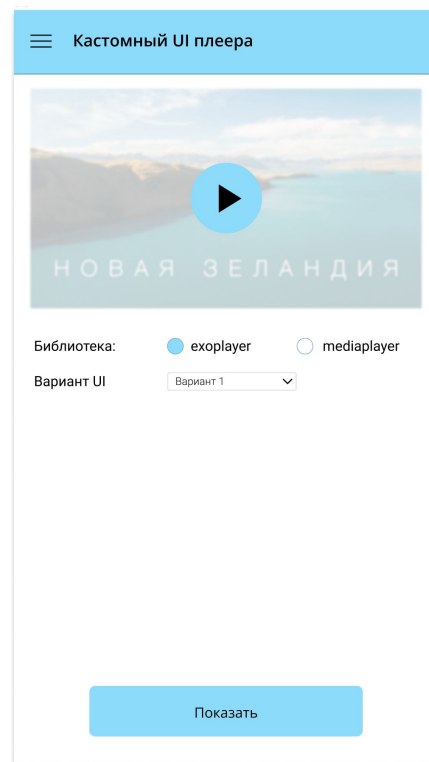
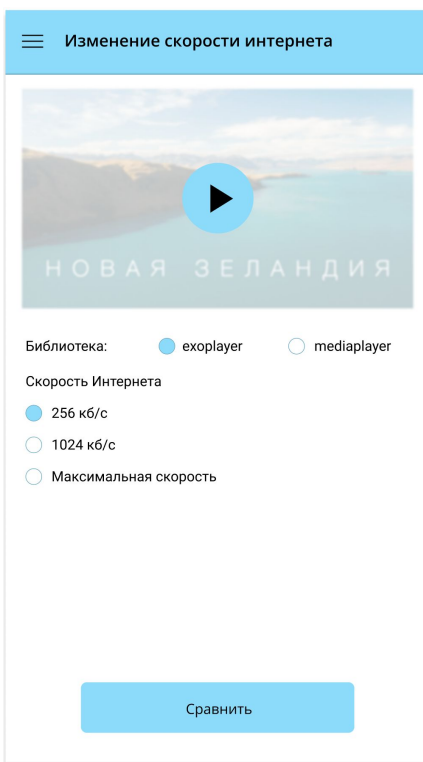
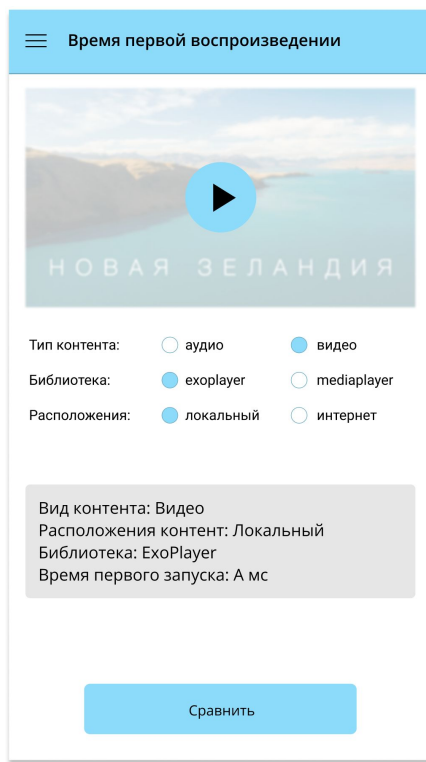
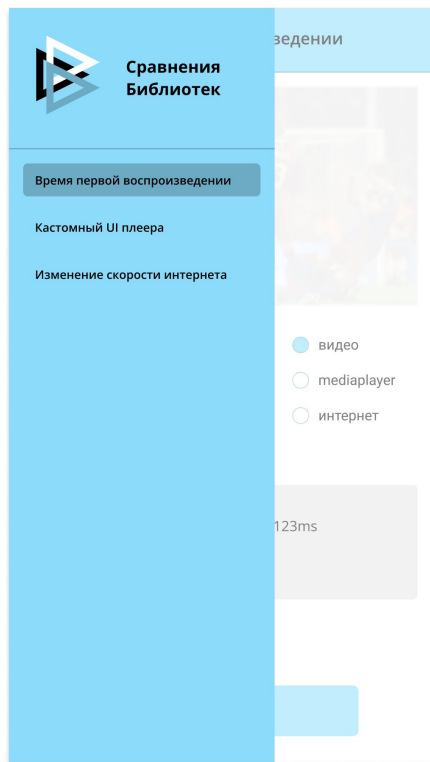


Model (Модель): Отдельный класс, содержащим данные для отображения информации пользователю.

View (Представление): Слой пользовательского интерфейса (xml файлы)

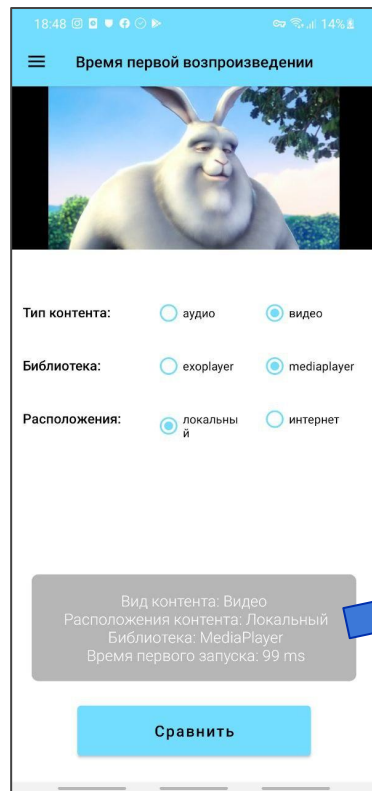
Controller (Контроллер): Компонент устанавливает связь между представлением и моделью (Activity классы)

Макет экранов приложения



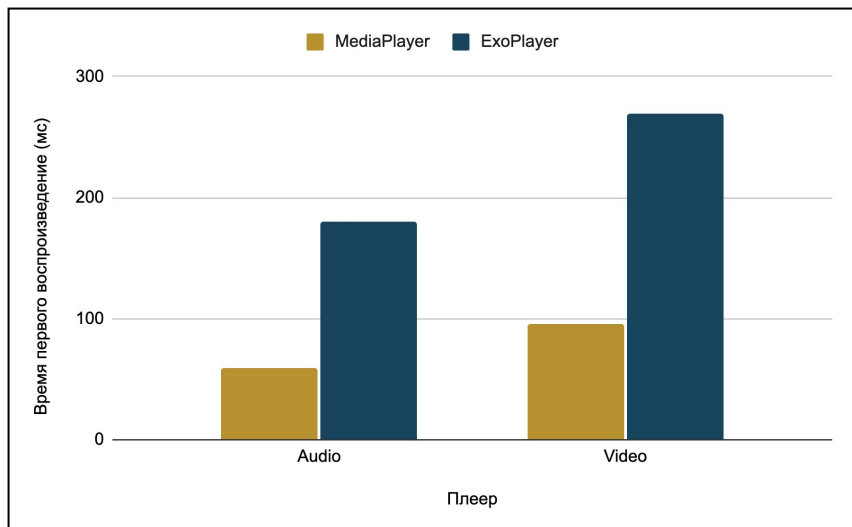
Результаты измерения [1.1]

Пример работы приложения и получении результатов

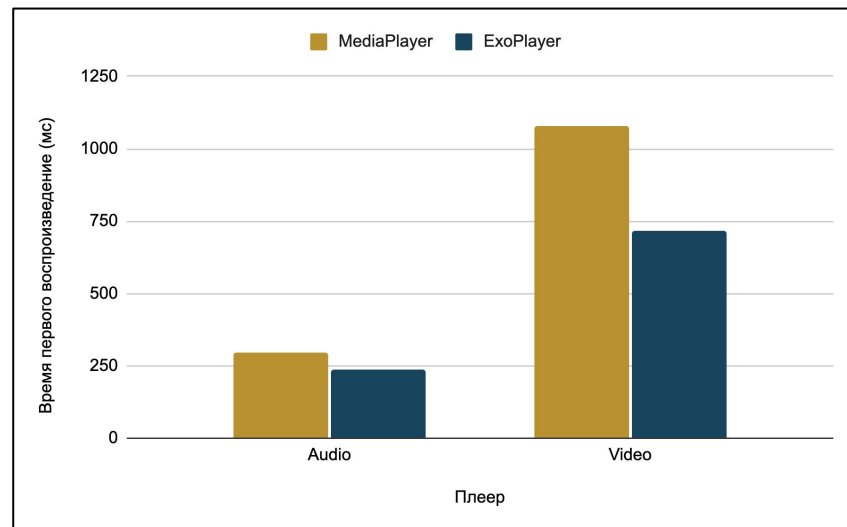


Попытка	Вид контента	
	аудио	видео
1	51	93
2	67	100
3	70	107
4	57	96
5	60	101
6	54	87
7	56	92
8	51	87
9	61	93
10	66	100
Среднее значение	59	96

Результаты измерения [1.2]

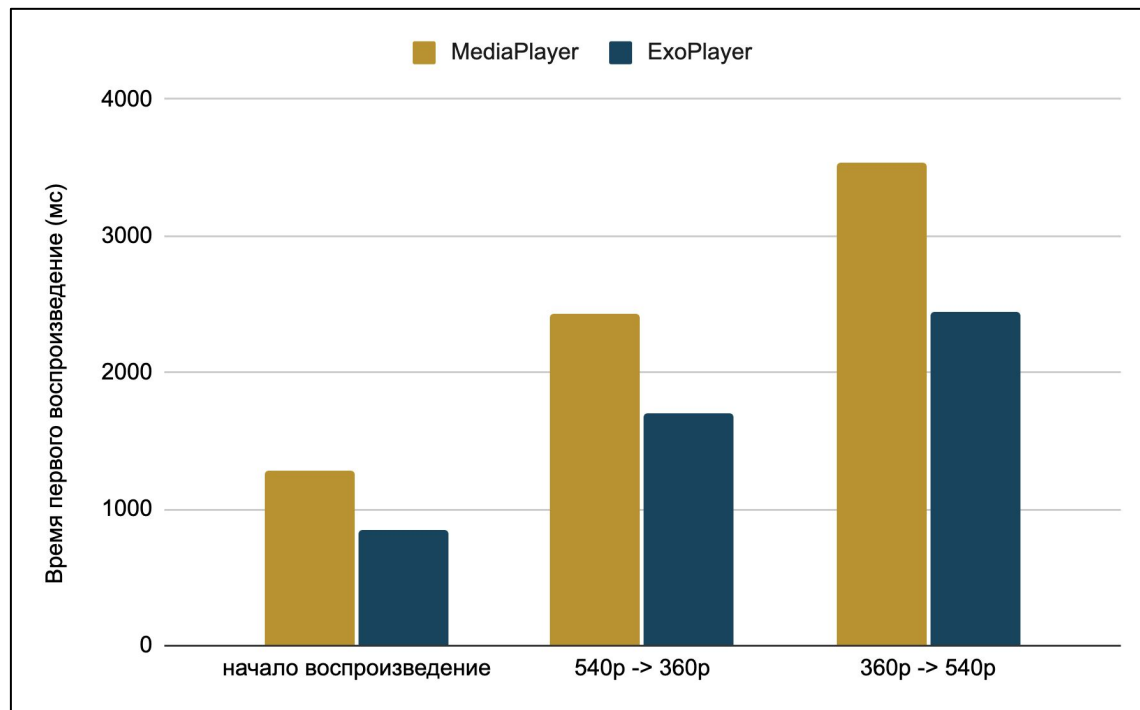


Время первого воспроизведения медиафайла из локального хранилища



Время первого воспроизведения медиафайла из сети

Результаты измерения [1.2]



Время затраченное на изменение качества видео при изменении скорости интернета

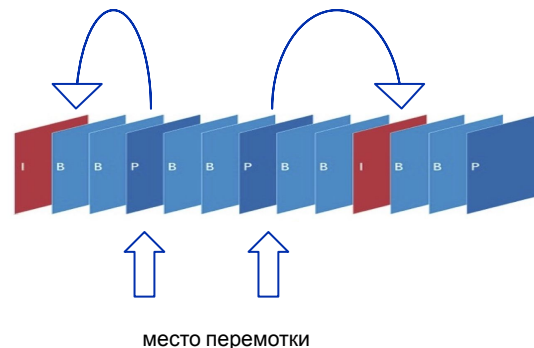
1. Изначальное качество: (На примере протокола HLS)

При воспроизведении контента через сеть можно указать минимальное качество, чтобы быстро загрузить первый кадр

2. Ускорить перемотку

Когда пользователь перематывает видео в конкретное место, мы можем попасть не в опорный кадр, а между ними.

Соответственно, всё что было с предшествующего опорного кадра до него нужно загрузить. Можно переходить на ближайший опорный кадр, тем самым сделав плеер еще быстрее



3. Запуск видео с определенного момента:

Если видео запускается не с начала и нужно перейти в определенную позицию

```
...  
player.prepare(mediaSource);  
player.seekTo(resumePositionMs);  
player.setPlayWhenReady(true);
```

```
...  
player.seekTo(resumePositionMs);  
player.prepare(mediaSource, /* resetPosition= */ false, ...);  
player.setPlayWhenReady(true);
```

4. Не отпускать кодек:

Следующее видео будет работать с теми же настройками декодера

```
void swapVideo(Uri uri)  
player.stop();  
mediaSource = buildMediaSource(uri);  
player.prepare(mediaSource);  
}
```

Возможности кастомизации UI

Возможность	Библиотека	
	MediaPlayer	ExoPlayer
Блокировка экрана	нет готового атрибута, легко реализуется	нет готового атрибута, легко реализуется
Перемотка на 10 секунд	нет готового атрибута, легко реализуется	есть готовый атрибут
Предпросмотр через бегунок прогресса	не реализуется	нет готового атрибута, легко реализуется
Перемотка быстрым нажатием	нет готового атрибута, легко реализуется	есть готовый атрибут
Кнопка переключения качества	нет готового атрибута, легко реализуется	нет готового атрибута, легко реализуется
Элемент управление громкостью	нет готового атрибута, легко реализуется	есть готовый атрибут
Поворот экрана	нет готового атрибута, легко реализуется	нет готового атрибута, легко реализуется

Результат сравнительного анализа

Требования	Библиотека	Преимущество
Воспроизведение локальных медиафайлов	MediaPlayer	аудио: быстрее на 120 мс видео: быстрее на 170 мс
Воспроизведение медиафайлов из сети	ExoPlayer	аудио: быстрее на 55 мс видео: быстрее на 410 мс (первый кадр) видео: быстрее на 735 мс (низкое качество) видео: быстрее на 1100 мс (высокое качество)
Необходимость воспроизведения несколько медиафайлов друг за другом	ExoPlayer	видео: быстрее на 100 мс
Наличие сложных элементов управления	ExoPlayer	наличие готовых атрибутов

- Проведен обзор библиотек для воспроизведения медиафайлов
- Описаны критерии сравнения
- Разработано мобильное приложение для реализации работы библиотек
- Получены результаты сравнения с помощью приложения по описанным критериям



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание