МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

**КОМПЛЕКСНЫЙ КУРСОВОЙ ПРОЕКТ**

по дисциплине: Технология разработки программного обеспечения

на тему «Конструктор интерьеров с использованием средств дополненной реальности»

Выполнили ст. гр. ИВТ-18

Сарманова В.В.

Лавров Р.В.

Проверил доцент кафедры ИВТ и ПМ,

к.т.н., доцент Валова О.В.

Чита

2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к комплексному курсовому проекту

по 09.03.01 Информатика и вычислительная техника

на тему: Конструктор интерьеров с использованием средств дополненной реальности

Выполнили студенты группы ИВТ–18 Сарманова Виталина Вячеславовна, Лавров Роман Викторович

Руководитель работы: доцент кафедры ИВТ и ПМ, к.т.н., доцент Валова Ольга Валерьевна

Чита

2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на комплексный курсовой проект

По дисциплине: Технология разработки программного обеспечения

Студентам: Сармановой Виталине Вячеславовне, Лаврову Роману Викторовичу

специальности: 09.03.01 Информатика и вычислительная техника

1 Тема курсовой работы: «Конструктор интерьеров с использованием средств дополненной реальности»

2 Срок подачи студентами законченной работы: 4.04.2022

3 Исходные данные к работе: описание предметной области;

4 Перечень подлежащих разработке в курсовом проекте вопросов:

1. Постановка и анализ задачи;
2. Анализ задачи;
3. Программная реализация;
4. Тестирование;
5. Техническое задание;
6. Руководство пользователя.

5 Перечень графического материала: –

Дата выдачи задания: 09.02.2022

Руководитель курсовой работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Валова О.В./

(подпись, расшифровка подписи)

Задание приняли к исполнению

«9» февраля 2022 г.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Сарманова В.В.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Лавров Р.В.

РЕФЕРАТ

Пояснительная записка – 22 с, 5 рис., 0 табл., 4 источника.

КОНСТРУКТОР, ИНТЕРЬЕР, ВЕБ СЕРВИС, БАЗА ДАННЫХ, ЯЗЫК JAVA, ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ, ANDROID-РАЗРАБОТКА

Каждый в своей жизни сталкивался с проблемой ремонта, для более комфортного решения этой проблемы создаются конструкторы интерьера.

При помощи такого инструмента можно без каких-либо материальным затрат разработать интерьер любого помещения.

Приложение позволяет разместить объект в пространстве и перейти к его странице покупки.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 6

1 Постановка и анализ задачи 7

1.1 Описание предметной области 7

1.2 Постановка задачи 9

1.3 Средства реализации 10

2 Анализ данных 11

2.1 Анализ данных клиентской части 11

2.2 Анализ данных серверной части 12

3 Программная реализация 13

3.1 Программная реализация клиентской части 15

3.2 Программная реализация серверной части 18

4 Тестирование 21

5 Документирование 22

5.1 Техническое задание 22

5.2 Руководство пользователя 24

ЗАКЛЮЧЕНИЕ 27

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 28

ПРИЛОЖЕНИЕ 29

ВВЕДЕНИЕ

В настоящее время многие люди сталкиваются с такой проблемой, как ремонт. Перед тем, как что-либо купить они должны понять, а подойдет ли тот или иной предмет мебели к нашему интерьеру.

Дополненная реальность все чаще появляется практически во всех сферах нашей жизни. Даже такая трудная задача как смена интерьера превращается в интересное и увлекательное занятие – сегодня AR дарит нам возможность без труда примерить новую мебель или обои, украсить интерьер с помощью картин и прочего декора.

В области обустройства дома AR уже демонстрирует положительные результаты – используя только смартфон и Интернет пользователь можете овладеть новым опытом покупок, который будет не только запоминающимся, но и весьма полезным.

Дополненная реальность для бизнеса является ценным инструментом, который может стать не только развлекательным способом вовлечения аудитории во взаимодействие, но и принести ощутимую прибыль компании.

Во-первых, в результате использования дополненной реальности снижается процент возвратов продуктов – согласно статистике, 89 % пользователей вернули продукт после покупки его в интернете.

Во-вторых, благодаря возможности посмотреть товар перед покупкой, пользователи более уверенно и быстро принимают решение об оплате продукта – 42 % опрошенных покупателей считают, что использование дополненной реальности позволяет быстрее принять финальное решение о покупке.

Разрабатываемое приложение позволит пользователям просмотреть предмет интерьера, не выходя из дома, а также получить ссылку на реальный товар в интернет-магазине.

# Постановка и анализ задачи

## 1.1 Описание предметной области

**Дополненная реальность (AR) —** технология, добавляющая в реальный мир цифровые объекты. Например, экран телефона, где изображение животного становится 3d объектом, отображающимся на экране; работа двигателя внутреннего сгорания, превращающаяся в анимацию при наведении на неё камеры.

Первоначально большинство новинок технического мира разрабатывались для военных нужд: например, сотовый телефон. Не стали исключением и технологии AR и VR.

В середине XX века, когда дополненная реальность и виртуальная реальность не отличалась друг от друга, транснациональная компания Philco Corporation спроектировала первый шлем виртуальной реальности для Пентагона. Учёный Том Коделл [ввёл](https://naked-science.ru/article/hi-tech/26-08-2013-371) понятие «дополненная реальность», в 1990 году. После этого AR и VR стали отдельными технологиями.

Дополненная реальность развивалась больше, как коммерческая технология для нужд бизнеса. Это и не удивительно, ведь достаточно иметь смартфон или планшет чтобы получить доступ к дополненной реальности, в свою очередь VR-технология подразумевает использование дорогостоящего оборудования: шлема и других гарнитур.

Аналогами разрабатываемого приложения выступают Wayfair и Houzz.

При помощи Wayfair вы получаете доступ к ассортименту мебели, декораций и других товаров.

Здесь есть незначительные недостатки, с которыми нужно бороться, от сбоев до недостатков в компоновке. А также данное приложение не доступно в Российской Федерации.

В Houzz также представлен набор предметов декора. Преимуществом приложения является сохранения различных настроек и возможность вернуться к ним при необходимости.

К сожалению, механика инструмента AR от Houzz требует некоторого маневрирования, чтобы все элементы отображались прямо на экране.

* 1. Постановка задачи

Целью работы является разработать приложение, конструктор интерьеров с использованием средств дополненной реальности.

На рисунке 1.1 изображена диаграмма вариантов использования

приложения для пользователя.



Рисунок 1 – Диаграмма вариантов использования

Пользователь сможет выбрать из двух предложенных вариантов: создать новый интерьер или открыть ранее созданный.

При выборе создания нового интерьера, пользователь сможет добавлять/удалять предметы декора, а также изменять цвет стен и пола. По окончанию работы он может сохранить свой вариант интерьера в файл, чтобы использовать его в будущем.

Для пользователя приложение должно предоставлять такие возможности как:

– предоставление пользователю возможности передвигать предмет интерьера;

– обеспечение возможности удалять предметы интерьера по одному или полностью очистить помещение от ранее размещённых объектов;

– предоставление возможности приобрести понравившейся предмет интерьера посредством перехода в магазин;

– сохранение и загрузка всей необходимой информации об разрабатываемом интерьере.

1.3. Средства реализации

В качестве платформы для разработки Android приложения была выбрана Android Studio, так как она наиболее удобная для разработки приложений, работающих под управлением операционной системы Android. Язык для написания этого приложения - Java.

Платформой для работы с дополненной реальностью была выбрана ARCore, потому что она обладает всеми необходимыми инструментами для работы, а также позволяет использовать как маркерную дополненную реальность (отображение модели происходит при распознавании маркера), так и без маркерную (платформа распознаёт плоскости и накладывает на них сетку, на которой располагаются объекты).

Веб сервис создан на платформе InteliJ IDEA, также на языке Java.

В качестве базы данных используется SQLite, благодаря своей компактности и быстроте работы.

**2 Анализ данных**

Данные, с которыми работает система можно разделить на несколько

групп. Входные данные – данные, поступающие в систему. Промежуточные данные – это данные, используемые системой во время работы. Выходные данные – информация, которую получает пользователь после выполнения какой-либо операции.

2.1 Анализ данных клиентской части

В качестве входных данных выступают:

– файлы сохранения;

– схема размещения: список и место положение объектов интерьера.

Файлы сохранения используются программой для хранения информации о расположении и характеристиках объектов.

Схему размещения можно сохранять в файл или загружать из файла.

Сведения об объектах хранятся в базе данных на сервере и дублируется на приложение в телефоне.

Базы данных описывается в серверной части приложения.

Выходными данными являются:

– файл сохранения со схемой размещения объектов. Будет находится на запоминающем устройстве смартфона.

– схема размещения на экране. Представлена массивом данных, элементы которого состоят из ID объекта и его местоположения.

– ссылки в магазин, на выбранные предметы интерьера. При переходе по ссылке, пользователь сможет рассмотреть и приобрести выбранный им товар.

2.2 Анализ данных серверной части

Входными данными являются HTTP запросы, отправленные смартфоном на сервер, в них входят:

– версия базы данных;

– файл базы данных, хранящийся на смартфоне.

Отправка базы данных и её версии, необходима для того, чтобы сравнить её с версией базы на сервере, и в случае обновления базы на сервере, отправить обновлённую базу на смартфон.

В таблице Models хранится идентификационный номер модели и её имя.

В таблице Textures хранится идентификационный номер текстуры и её имя.

Таблица Object отвечает за соединение двух вышеперечисленных таблиц.

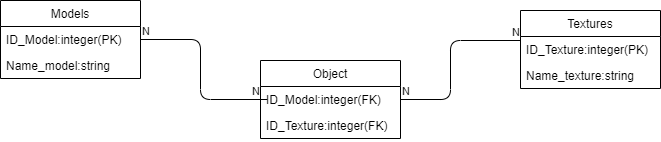


Рисунок 2.1 База данных проекта

В случае если на сервере база данных была модернизирована, то выходными данными будут:

– файл базы данных. Будет находится на запоминающем устройстве смартфона.

– новые 3d модели. После получения и преобразования данных в файл модели, он вместе со своим ID будет помещён в массив.

– текстуры к новым моделям. Подгружаются по тому же принципу что и модели, помещаются в отдельный массив.

1. **Программная реализация**

Создаваемый проект имеет модульную структуру, исходя из этого, он включает в себя несколько модулей, которые представлены в виде диаграммы компонентов на рисунке 3.1.

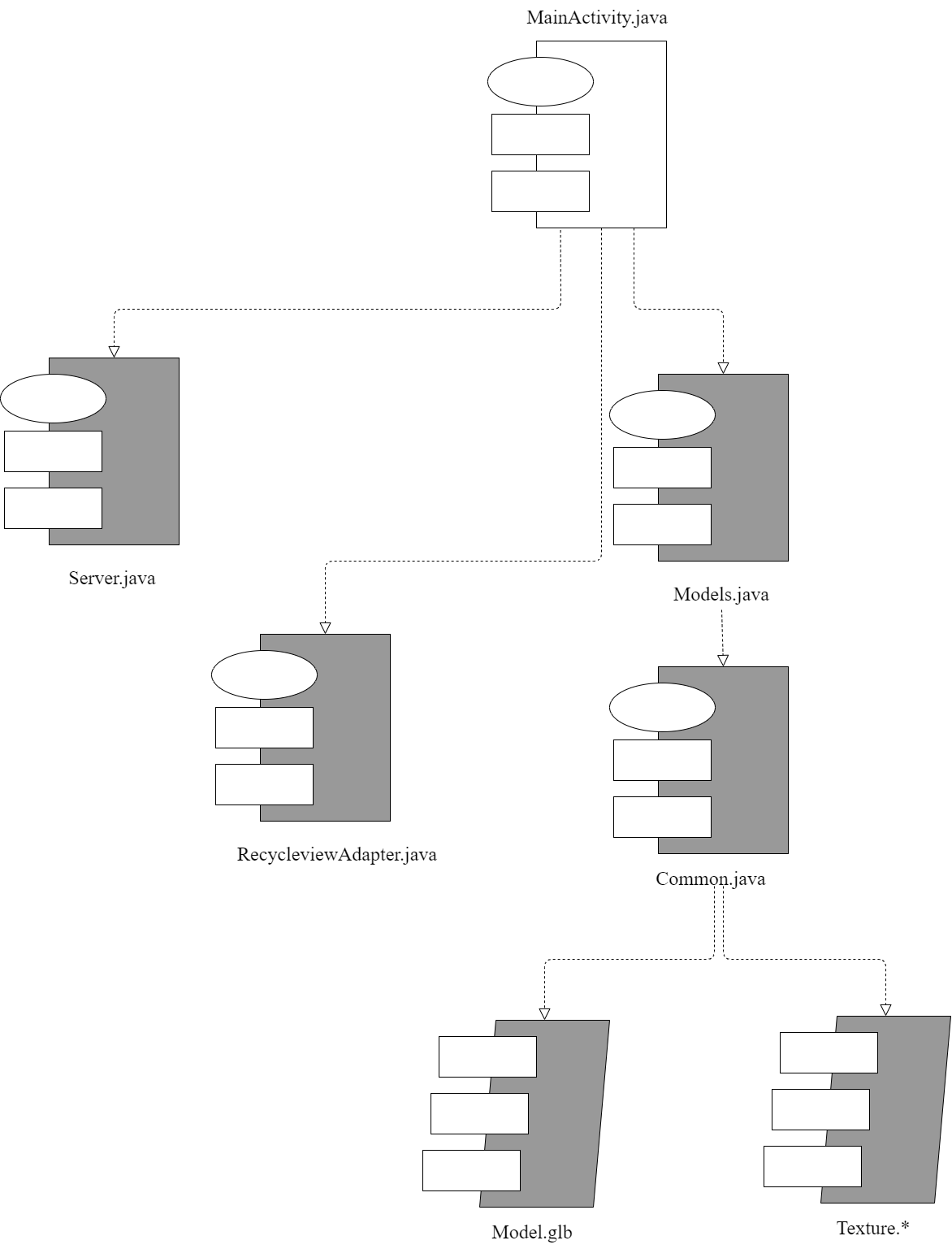


Рисунок 2 Диаграмма компонентов

Таблица 3.1 – Описание компонентов клиентской части

|  |  |
| --- | --- |
| Компоненты | Описание |
| MainActivity.java | Основной файл программы, отвечает за связь интерфейса с остальными файлами |
| Server.java | Клиентская часть приложения |
| Models.java | Отвечает за подгрузку, прорисовку и расположение моделей |
| RecyclerviewAdapter.java | Реализует работу интерфейса |
| Common.java | Отвечает за тип 3d моделей |
| Model.glb | 3d модель |
| Texture.\* | Отвечает за текстуры и цвета модели |

Таблица 3.2 – Описание компонентов серверной части

|  |  |
| --- | --- |
| Компоненты | Описание |
| index.php | Отвечает за принятие, обработку запросов, а также за отправление ответа смартфону, код этого файла расположен в приложении |
| db\_work.php | Реализует работу с базой данных (добавление, изменение, удаление); |
| main\_form.html | HTML-страница загрузки новых объектов в базу данных |

Данное приложение имеет клиент-серверную архитектуру.

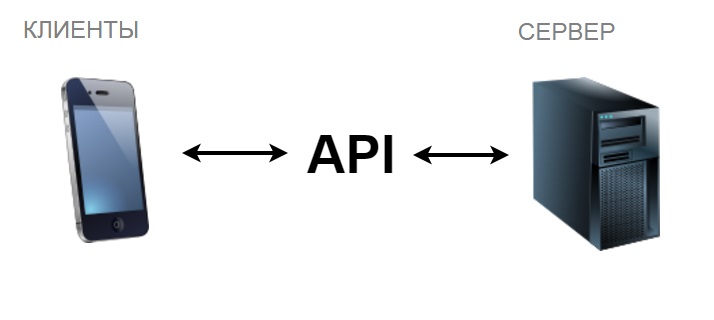


Рисунок 3 - Схема работы разрабатываемого приложения

**3.1 Программная реализация клиентской части**

Для работы с дополненной реальностью используется библиотека ARCore. Эта библиотека использует технологию безмаркерной дополненной реальности, то есть для отображения модели не нужен определённый маркер. Приложение распознаёт поверхности и создаёт на них сетку, на которую можно поместить, выбранную модель, а также переместить и развернуть в пространстве.

Модели хранятся в массиве, вместе с их ID из базы данных, для точного идентифицирования модели. Во время работы приложения создаётся массив названий моделей и их изображений для отображения в меню. После выбора объекта и нажатия на экран для его размещения, вызывается функция placeModel().Она создаёт экземпляр класса ModelRenderable, который содержит в себе информацию о выбранной модели:

- путь к модели на смартфоне;

- масштаб отображаемой модели;

- положение модели относительно места нажатия.

Код метода placeModel:

private void placeModel(Anchor anchor) {  
 ModelRenderable  
 .*builder*()  
 .setSource(this, RenderableSource  
 .*builder*()  
 .setSource(this, Uri.*parse*("/data/data/com.example.secondar/files/"+Common.*model*),RenderableSource.SourceType.*GLB*)  
 .setScale(0.75f)  
 .setRecenterMode(RenderableSource.RecenterMode.*ROOT*)  
 .build()  
 ).setRegistryId(Common.*model*)  
 .build()  
 .thenAccept(modelRenderable-> addNodeToScene( modelRenderable,anchor,Uri.*parse*(Common.*model*)))  
 .exceptionally(throwable -> {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setMessage(throwable.getMessage()).show();  
 return null;  
  
 });  
}

Затем в этой же функции вызывается метод addNodeToScene(), он создаёт «якорь», к которому будет привязаны выбранная модель и помещает его на сцену. Также здесь создаётся объект класса TransformableNode, при помощи которого есть возможность перемещать/вращать/изменять размер модели.

Код метода addNodeToScene:

private void addNodeToScene(ModelRenderable modelRenderable,Anchor anchor,Uri model) {  
 AnchorNode anchorNode = new AnchorNode(anchor);  
 String a = model.getPath();

if (a.equals("arrow.glb")) {  
 TransformableNode node = new TransformableNode(arFragment.getTransformationSystem());  
 node.setParent(anchorNode);  
 node.setRenderable(modelRenderable);  
 arFragment.getArSceneView().getScene().addChild(anchorNode);  
 node.select();  
 View.OnClickListener save\_pose = new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (a.equals("arrow.glb")) {  
 removeAnchorNode(anchorNode);  
 Pose pose = anchor.getPose();  
 coor1 = pose.getRotationQuaternion();  
 coorx = pose.getTranslation();  
 coor1 = *round*(coor1, 4);  
 coorx = *round*(coorx, 4);  
 Pose p = new Pose(coorx, coor1);  
 Session session1 = arFragment.getArSceneView().getSession();  
 Anchor anchor1 = session1.createAnchor(p);  
 anchorNode.setAnchor(anchor1);  
 anchorNode.setRenderable(modelRenderable);  
  
 arFragment.getArSceneView().getScene().addChild(anchorNode);  
 }  
 }  
 };  
 save.setOnClickListener(save\_pose);  
} else {  
 TransformableNode node = new TransformableNode(arFragment.getTransformationSystem());  
 node.setParent(anchorNode);  
 node.setRenderable(modelRenderable);  
 arFragment.getArSceneView().getScene().addChild(anchorNode);  
 node.select();

* 1. **Программная реализация серверной части**

Для реализации данной архитектуры, со стороны клиента, используется библиотека OkHTTP3 языка Java. Из её название становится ясно, что данные отправляются при помощи протокола HTTP.

Код отправки запроса на сервер:

@RequiresApi(api = Build.VERSION\_CODES.*O*)  
public void postQueryExample(View view) throws IOException, JSONException {  
 String filename = "models\_phone.db";  
 String filepath = "sampleData";  
 String fileContent = "Hello world!";  
 Db\_helper databaseHelper = Db\_helper.*getInstance*(getApplicationContext());  
 SQLiteDatabase sqLiteDatabase = databaseHelper.getWritableDatabase();  
 byte[] bytes = sqLiteDatabase.toString().getBytes();  
 File file = new File("/data/user/0/com.example.secondar/databases/", filename);  
 if(file.exists())  
 {int a = 0;}  
 File file\_json = new File("/data/user/0/com.example.secondar/files/", "response.json");  
 new Thread(()-> {  
 final MediaType JSON = MediaType.*parse*("application/json; charset=utf-8");  
 OkHttpClient client = new OkHttpClient();  
 RequestBody body = RequestBody.*create*(JSON, "{"+myString+":"+10+"}");  
 RequestBody requestBody = new MultipartBody.Builder()  
 .setType(MultipartBody.*FORM*)  
 .addFormDataPart("version","1")  
 .addFormDataPart("file" , file.getName() , RequestBody.*create*(MediaType.*parse*("file/\*;charset=utf-8"), file) )  
 .build();  
 Request request = new Request.Builder()  
 .url(Urladdress)  
 .post(requestBody)  
 .build();  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 //do failure stuff  
 }  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 //do success stuff  
 if (response.isSuccessful()) {  
 try {  
 int count = 0;  
 response = client.newCall(request).execute();  
 assert response.body() != null;  
 InputStream is = response.body().byteStream();  
 BufferedInputStream input = new BufferedInputStream(is);  
 OutputStream output = new FileOutputStream(file\_json);  
 byte[] data = new byte[1024];  
 long total = 0;  
 while ((count = input.read(data)) != -1) {  
 total += count;  
 output.write(data, 0, count);  
 }  
 output.flush();  
 output.close();  
 input.close();  
 } catch (IOException e) {  
 Toast toast = Toast.*makeText*(getApplicationContext(), "Exception: " + Log.*getStackTraceString*(e), Toast.*LENGTH\_LONG*);  
 toast.show();  
 }  
 }  
 }  
 });  
 }).start();

При работе программа через определённое количество времени будет отправлять, так называемый POST-запрос, содержащий в себе базу данных 3d моделей. Это запрос будет обработан модулем index.php, в частности, этот модуль сравнит базу данных, которая была на сервере, с полученной от смартфона. Если будут найдены различия, то сервер обработает базы, и отправит ответ смартфону, который будет содержать недостающие объекты.

Перед отправкой все нужные объекты кодируются в base64 кодировку, после чего кодируются ещё раз, но уже в формат JSON. JSON хранит в себе новую версию базы, а также: файлы 3d-моделей, их названия, ID, и набор текстур для каждой модели.

Смартфон получает в качестве ответа от сервера этот JSON файл и декодирует его в массив байт, который затем декодируется в base64 и записывается в файл с нужным именем и расширением. После чего вызывается метод Add\_from\_response(), отвечающий за добавление полученных моделей в основной массив объектов, и отображение в меню выбора объектов.

**4 Тестирование**

Тестирование программного обеспечения – проверка соответствия между реальным и ожидаемым поведением программы, выполняемое на конечном наборе тестов. В более широком смысле, тестирование – это одна из техник контроля качества, в которую входят:

– активности по планированию работ и проектированию тестов;

– выполнение тестирования;

– анализ полученных результатов.

Перед тем как продемонстрировать работу приложения, было проведено большое количество тестов, для выявления слабых сторон и возможных ошибок.

В ходе тестирования были обнаружены ошибки, связанные с масштабом определённых 3d-объектов, решением было изменение настроек объектов в программе Blender 2.8.

Также были обнаружены критические ошибки приводящие к аварийному завершению работы приложения, они связаны с некорректностью данных поступающих в метод AddNodeToScene(), в частности неправильно указанный путь до какого-либо 3d-объекта.

В серверной части приложения существовали ошибки с отправкой ответа на POST-запрос от смартфона, а именно кодированием данных в base64 кодировку и помещение их в JSON файл[4].

Большинство ошибок были обнаружены при помощи многократных запусков приложения.

В результате тестирования было обнаружено не мало ошибок, как в клиентской, так и в серверной частях, они были успешно исправлены.

Благодаря этому приложение работает корректно и стабильно, и вывести его из строя практически невозможно.

**5 Документирование**

**5.1 Техническое задание**

5.1.1 Введение

Программа «Конструктор Интерьеров» предназначена для построения модели интерьера, подходит как для физического лица, так и для организации.

5.1.2 Назначение разработки

Программа:

* решает вопросы выбора варианта обустройства помещения;
* каждый объект имеет ссылку на магазин, в котором можно приобрести;

5.1.3 Требования к программе или программному изделию

5.1.3.1 Требования к функциональным характеристикам

При работе с программой предусматривается одна категория пользователей: пользователь.

Программы должна выполнять следующие функции:

* открытие сохранённого интерьера;
* изменение обоев или цвета стен;
* размещение мебели;
* удаление предмета;
* очистить интерьер;
* сохранение интерьера;
* иметь ссылку на выбранном объекте в магазине.

5.1.3.2 Требования к надежности

В программе должна присутствовать проверка входной информации на соответствие типов, принадлежность диапазону допустимых значений и соответствие структурной корректности. В случае возникновения ошибок предусмотреть возможность вывода информативных диагностических сообщений.

5.1.3.3 Требования к составу и параметрам технических средств

Программа должна функционировать на смартфоне: с микропроцессором не ниже Qualcomm Snapdragon 620G, с оперативной памятью объемом не менее 4 Гб.

5.1.3.4 Требования к информационной и программной совместимости

Программа должна функционировать под управлением ОС семейства Android не ниже версии 8.

5.1.3.5 Требования к программной документации

Программная документация должна быть представлена встроенной справочной системой.

**5.2 Руководство пользователя**

При запуске приложения пользователю будет предложена 2 варианта действий: создать новый интерьер или открыть ранее созданный.

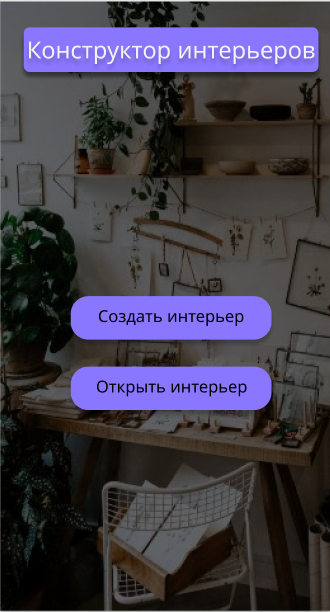


Рисунок 4 – Окно выбора способа создания интерьера

При выборе создать новый интерьер, откроется экран создания интерьера. В первую очередь пользователь должен определить центр комнаты, для правильного отображения объектов. Для этого нужно нажать на определение центра комнаты и переместить «стрелочку» в нужное место, после чего нажать на кнопку «Выбрать центр комнаты». После всех этих действий пользователь сможет расставить различные объекты как ему удобно.

Затем этого он может сохранить созданный им интерьер в типизированный файл, чтобы при следующем запуске открыть его. В файл сохраняются все координаты моделей, а также их названия, ID и набор текстур.

Открытие ранее созданного позволяет открыть готовый файл и разместить все объекты, как они были размещены в тот сеанс работы, который был сохранён. Однако, для правильного отображения всех объектов нужно установить центр комнаты там же, где он был при создании этого интерьера.

После размещения понравившейся модели пользователь может получить ссылку на магазин, в котором можно купить нужный объект интерьера.



Рисунок 5 – Пример размещения объекта интерьера

Модели в приложение практически идентичны реальным предметам по масштабу, однако, всегда полезно тщательно оценивать размеры виртуальных предметов по сравнению с реальной комнатой. Такие вещи, как экран телефона и ваше положение в пространстве, могут ввести вас в заблуждение. Помните о таких факторах, чтобы избежать ошибок.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы было разработано приложение, позволяющее использовать дополненную реальность для создания интерьера.

Для пользователя приложение реализовывает такие возможности как:

– возможность размещать, передвигать предмет интерьера;

– обеспечивает возможности удалять предметы интерьера по одному или полностью очистить помещение от ранее размещённых объектов;

– предоставляет возможность приобрести понравившейся предмет интерьера посредством перехода по ссылке в интернет-магазин;

– сохранение и загрузка всей необходимой информации об разрабатываемом интерьере.

Со временем можно будет расширить функциональные возможности программного продукта, путём добавления дополнительных функций, к примеру, добавление поддержки Depth API (распознавание объектов реального мира для более точного отображения 3d-объектов).

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дополненная реальность (AR) – что это и какую пользу приносит [Электронный ресурс]. – Режим доступа: <https://netology.ru/blog/09-2020-what-is-ar> (дата обращения 28.02.2022).
2. Дополненная реальность на рынке дизайна интерьера: возможности, преимущества, перспективы [Электронный ресурс]. – Режим доступа: https://vc.ru/marketing/196399-dopolnennaya-realnost-na-rynke-dizayna-interera-vozmozhnosti-preimushchestva-perspektivy (дата обращения 4.03.2022).
3. Обрабатываем POST- запросы в PHP [Электронный ресурс]. – Режим доступа: <https://php.zone/kurs-php-dlya-nachinayushih/post-zaprosy-v-php> дата обращения 10.03.2022)
4. Дополненная реальность [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Дополненная_реальность> (дата обращения 15.03.2022)

ПРИЛОЖЕНИЕ

Код файла index.php:

<?php

class db\_listener

{

public $version = "не определено ";

public $info = "не определен ";

public $models\_server;

public $data;

public $files;

public function db\_work()

{

$models\_server = fopen("models\_server.db", "r");

$data = fread($models\_server, filesize("models\_server.db"));

if (isset($\_POST["version"])) {

$version = $\_POST['version'];

}

fclose($models\_server);

$name\_model;

$dbp = new SQLite3("models\_phone.db");

$dbs = new SQLite3("models\_server.db");

$sql = 'SELECT \* FROM models ORDER BY \_id DESC LIMIT 1';

$select\_phone = $dbp->querySingle($sql, true);

$select\_server = $dbs->querySingle($sql, true);

$phone\_count = $select\_phone["\_id"];

$server\_count = $select\_server["\_id"];

if($phone\_count!= $server\_count)

{

$value = abs($server\_count-$phone\_count);

for ($i = $phone\_count; $i <= $server\_count; $i++) {

$sql = 'SELECT \* FROM models WHERE \_id = '.$i;

$select = $dbs->querySingle($sql, true);

$name\_model = $select["name"];

foreach(glob($name\_model) as $file) {

$files = basename($file);

}

}

}

$filename = "Lamp.glb";

$texture = "texture.jpg";

if(file\_exists($filename)){

$files = array($filename, $texture);

$zipname = 'file.zip';

$zip = new ZipArchive;

$zip->open($zipname, ZipArchive::CREATE);

if($zip !== true) {

foreach ($files as $file) {

$zip->addFile($file);

}

}

$zip->close();

$file1 = file\_get\_contents($filename);

$file\_tex = file\_get\_contents($texture);

$named\_array = array(

"files" => array(

array(

"name" => $filename,

"data" => base64\_encode($file1)

),

array(

"name" => $texture,

"data" => base64\_encode($file\_tex)

)

)

);

// encode the object in JSON format

$file\_json = 'results.json';

$json = json\_encode(array('files' => $named\_array));

$finfo = finfo\_open(FILEINFO\_MIME\_TYPE);

header('Content-Type: application/json');

finfo\_close($finfo);

header('Content-Disposition: attachment; filename='.$file\_json);

header('Content-Length: ' . filesize($file\_json));

ob\_clean();

flush();

readfile($file\_json);

exit;

}

}

public function db\_file()

{

$uploaddir = "C:/Xampp/htdocs/";

$uploadfile = $uploaddir . basename($\_FILES['file']['name']);

if (move\_uploaded\_file($\_FILES['file']['tmp\_name'], $uploadfile)) {

echo "The file has been uploaded successfully";

} else {

echo "There was an error uploading the file";

}

}

}

$db\_listen = new db\_listener;

$db\_listen->db\_work();

$db\_listen->db\_file();

?>