

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информатики, информационных технологий
и методики обучения информатике

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ
ПРОВЕДЕНИЯ ПСИХОЛОГИЧЕСКОГО
ЭКСПЕРИМЕНТА ПО ИЗУЧЕНИЮ
ОСОБЕННОСТЕЙ ВОСПРИЯТИЯ
ДОШКОЛЬНИКОВ ПРИ ИСПОЛЬЗОВАНИИ
ИНФОРМАЦИОННЫХ УСТРОЙСТВ С
СЕНСОРНЫМ ЭКРАНОМ**

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Работа допущена к защите
« ____ » _____ 2022 г.
Зав. кафедрой _____

Исполнитель: студент группы ИСиТ 1801
Института математики, физики,
информатики и технологий
Петров А.А.

Руководитель: кандидат педагогических наук,
доцент
Газейкина А.И.

Екатеринбург – 2022

Реферат

Петров А. А. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПРОВЕДЕНИЯ ПСИХОЛОГИЧЕСКОГО ЭКСПЕРИМЕНТА ПО ИЗУЧЕНИЮ ОСОБЕННОСТЕЙ ВОСПРИЯТИЯ ДОШКОЛЬНИКОВ ПРИ ИСПОЛЬЗОВАНИИ ИНФОРМАЦИОННЫХ УСТРОЙСТВ С СЕНСОРНЫМ ЭКРАНОМ, выпускная квалификационная работа: 50 стр., рис. 24, библи. 32 назв.

Ключевые слова: ПРИЛОЖЕНИЕ, ПСИХОЛОГИЯ, РАЗРАБОТКА, ANDROID, JAVA, LIBGDX.

Предмет разработки – мобильное приложение для манипуляции трехмерными объектами.

Цель работы – спроектировать и произвести программную реализацию мобильного приложения для манипуляции трехмерными объектами.

В настоящей работе описаны результаты проектирования и программной реализации мобильного приложения, позволяющего манипулировать трехмерными объектами и получать кинематический профиль движения фигуры.

Система реализована на языке Java с использованием фреймворка LibGDX. Программный продукт реализован в виде приложения для устройств с сенсорным экраном. Приложение предназначено для преподавателей института Психологии, кафедры общей психологии и конфликтологии, Уральского государственного педагогического университета, разработанное в рамках исследования «Распознавание виртуальных моделей объектов».

Оглавление

Реферат	2
Введение	4
Глава 1. Теоретические основы разработки мобильных приложений для ОС Android	6
1.1 Операционная система Android	6
1.2 Выбор метода разработки	9
1.3 Формализованное описание технического задания	17
Глава 2. Разработка мобильного приложения средствами межплатформенной среды LibGDX для выполнения манипуляций с трехмерными объектами	24
2.1 Модель мобильного приложения	24
2.2 Описание мобильного приложения для изучения особенностей зрительного и тактильного восприятия при использовании дошкольниками информационных устройств с сенсорным экраном «PsychoTest»	27
2.3 Результаты апробации, техническая документация	40
Заключение	47
Список информационных источников	48

Введение

Сегодня компьютерные технологии стали неотъемлемой частью нашей жизни. Они очень помогают в медицине, машиностроении, обучении, производстве, а также и в других сферах. Применение технологий открывает новые возможности.

Применяемые в работе педагога-психолога компьютерные технологии органично дополняют традиционные методы, в том числе, расширяют возможности взаимодействия психолога с детьми и подростками.

Так как теперь у каждого ребенка есть возможность притронуться к миру технологий с ранних лет, преподаватели Института психологии, кафедры общей психологии и конфликтологии, Уральского государственного педагогического университета, проводят исследование «Распознавание виртуальных моделей объектов». Для проведения эксперимента они обратились с просьбой разработать приложение, которое поможет им в этом. Данное приложение будет разработано под операционную систему Android на базе фреймворка LibGDX.

На данный момент самой доступной платформой для разработки мобильных приложения является Android. Данная платформа настолько популярна благодаря своей открытости и простоте в использовании. Устройства под управлением Android являются доступнее, чем устройства под управлением IOS, а также дают пользователям беспрепятственно устанавливать сторонние приложения.

LibGDX предлагает единый API, который поддерживает: Windows, Linux, macOS, Android, iOS и Web. Разработчики могут использовать различные серверные части для доступа к возможностям хост-платформы без необходимости написания кода, специфичного для конкретной платформы. Рендеринг выполняется на всех платформах через Open GL ES 2.0/3.0.

Предмет разработки: мобильное приложение на базе межплатформенной среды разработки игр LibGDX, работающее под управление операционной системы Android, для различных манипуляций с трехмерными объектами.

Цель разработки: разработать мобильное приложение на базе межплатформенной среды разработки игр LibGDX, работающее под управление операционной системы Android, для различных манипуляций с трехмерными объектами и выводом статистических данных.

Задачи:

1. Проанализировать особенности разработки и функционирования приложений под управлением ОС Android.
2. Освоить межплатформенную среду разработки приложений LibGDX.
3. В соответствии с техническим заданием провести разработку мобильного приложения на основе выбранных технологий и программного обеспечения.
4. Подготовить техническую и сопроводительную документацию.
5. Провести апробацию разработанного приложения в процессе реализации гранта РФФИ «Изучение феномена несоответствия зрительного и тактильного восприятия при использовании дошкольников устройств с сенсорным экраном» (Институт психологии и ИМФИиТ УрГПУ).

Глава 1. Теоретические основы разработки мобильных приложений для ОС Android

1.1 Операционная система Android

Android – универсальная операционная система, на которой функционирует огромное количество устройств: не только смартфоны и планшеты, а также электронные книги, цифровые проигрыватели, наручные часы, фитнес-браслеты, игровые приставки, ноутбуки, нетбуки, смартбуки, очки Google Glass, телевизоры, проекторы, автомобильные системы, бытовые приборы и т.д. На данный момент Android является самой популярной мобильной операционной системой в мире. Благодаря своей доступности и открытости Android является фаворитом для пользователей и разработчиков, занимая лидирующие позиции на рынке.

Разработка операционной системы Android началась в 2003 году, когда Энди Рубин с товарищами (Ник Сирс, Крис Уайт и Рич Майнер) решил создать мобильную операционную систему и зарегистрировал компанию Android Inc [31]. Разработчики сначала сосредоточились на устройствах, которые могли бы постоянно находиться у пользователей, определять местоположение по GPS. К 2005 году Энди и друзья потратили все средства, но по счастливой случайности к ним присмотрелись из Google и 17 августа 2005 года корпорация стала полноправным владельцем маленькой Android Inc. Стоит отметить, что Google на тот момент не имела каких-то особых планов на гаджеты, а была больше сосредоточена на улучшении собственного ПО и алгоритмов поиска. В этом же году на фоне судебных разбирательств Oracle и Google решается, что Android будет свободной операционной системой [1].

В сводной таблице (см. *Таблица 1.*) представлены основные улучшения системы вводимые с новой версией ОС Android.

Таблица 1.

Версии ОС Android и их основные улучшения

Название	Номер версии	Дата стабильного релиза	Основные улучшения
нет официального названия	1.0	23 сентября 2008 г.	Первая стабильная версия системы. Добавлен магазин приложение Android Market. [30]
Cupcake	1.5	27 апреля 2009 г.	Поддержка виджетов и папок. Добавлена функция автозаполнения в системную клавиатуру. Добавлены инструменты для работы с файловыми системами карт памяти. [30]
Donut	1.6	15 сентября 2009 г.	Обновленный дизайн. Добавлен новый интерфейс для работы с фото, видеокамерой и галереей. [30]
Eclair	2.0—2.1	26 октября 2009 г.	Добавлена функция мультиаккаунтов. Обновление графической архитектуры. Добавлена поддержка вспышки. [30]
Froyo	2.2—2.2.3	20 мая 2010 г.	Оптимизация ОС, памяти и производительности. Поддержка Adobe Flash 10.2 и OpenGL Es 2.0. [30]
Gingerbread	2.3—2.3.7	6 декабря 2010 г.	Обновленный дизайн. Поддержка сверхвысоких размеров экранов и разрешений. Системная поддержка копирования и вставки. Поддержка нескольких камер. Переход с YAFFS на ext4. [30]
Honeycomb	3.0—3.2.6	22 февраля 2011 г.	Поддержка многоядерных процессоров. Улучшена многозадачность. [30]
Ice Cream Sandwich	4.0—4.0.4	18 октября 2011 г.	Создана единая оболочка для смартфонов, планшетов и других устройств. Улучшение систем безопасности. [30]
Jelly Bean	4.1—4.3.1	9 июля 2012 г.	Поддержка устройств ввода с шрифтом Брайля.

			Поддержка нескольких пользователей на устройстве. Обновление ядра до ветки 3.4. [30]
KitKat	4.4—4.4.4	31 октября 2013 г.	Оптимизация многозадачности. Запуск приложений в песочнице. Поддержка Chromecast. [6]
Lollipop	5.0—5.1.1	12 ноября 2014 г.	Material design. Уведомления на экране блокировки. Улучшена стабильность системы. [7]
Marshmallow	6.0—6.0.1	5 октября 2015 г.	Введена платежная система Android Pay. Улучшена экономия заряда аккумулятора. [8]
Nougat	7.0—7.1.2	22 августа 2016 г.	Добавлена функция «Разделение экрана». Фоновое переключение задач. Объединение уведомлений одного приложения. Поддержка режима «Картинка в картинке». Программный аналог 3D Touch. [9]
Oreo	8.0—8.1	21 августа 2017 г.	Обновление дизайна. Ограничение фоновых процессов. Динамические иконки. Экранные жесты. [10]
Pie	9	6 августа 2018 г.	Поддержка вырезов в дисплее. Дополненный системный интерфейс на основе жестов. Поддержка Vulkan 1.1. [11]
Queen Cake	10	3 сентября 2019 г.	Поддержка складных телефонов. Контроль использования местоположения приложениями. Встроенная запись экрана. Улучшенная конфиденциальность. [3]
Red Velvet Cake	11	8 сентября 2020 г.	Поддержка 5G. Поддержка новых типов экранов. Улучшение совместимости приложений.

			Улучшенная поддержка нейросетей. [4]
Snow Cone	12—12.1	4 октября 2021 г.	Поддержка Ultra-Wideband. Новые жесты. Увеличен контроль за фоновыми приложениями. Встроенная поддержка создания скриншотов с прокручиваемым экраном. [5]

Операционная система Android развивается уже на протяжении 19 лет. С каждой новой версией система становится стабильнее и безопаснее. Устройства на данной ОС являются одними из самых доступных продуктов, конечное число таких устройств превышает 3 млрд.

Благодаря своей популярности система имеет большой выбор среди фреймворков и движков, что тоже является большим преимуществом для разработчика.

Таким образом, на основании вышеизложенного можно сделать вывод, что разработка приложения под данную ОС не будет вызывать особых проблем у разработчика, связанных с нехваткой информации или невозможностью использовать те или иные функции устройства или системы.

1.2 Выбор метода разработки

Сенсорные устройства стали частью нашей жизни, с каждым годом выпускается все больше и больше различных моделей. Ведущими операционными системами для таких устройств являются Android (Google Inc.) и iOS (Apple).

Android — операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, нетбуков, смартбуков, очков Google Glass, телевизоров, проекторов и других устройств (в 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов).

Изначально разрабатывалась компанией Android, Inc., которую затем приобрела Google. Основана на ядре Linux и собственной реализации виртуальной машины Java компании Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который занимается поддержкой и дальнейшим развитием платформы [2].

iOS (до 24 июня 2010 года — iPhone OS) — мобильная операционная система для смартфонов, электронных планшетов, носимых проигрывателей, разрабатываемая и выпускаемая американской компанией Apple. Выпущена в 2007 году; первоначально — для iPhone и iPod touch, позже — для таких устройств, как iPad. В 2014 году появилась поддержка автомобильных мультимедийных систем Apple CarPlay. В отличие от Android (Google), выпускается только для устройств, производимых фирмой Apple.

В iOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. Ядро iOS почти идентично ядру настольной операционной системы Apple macOS (ранее называвшейся OS X). Начиная с самой первой версии iOS работает только на планшетных компьютерах и смартфонах с процессорами архитектуры ARM. [16]

Выбор операционной системы

Для разработки приложения было решено использовать систему Android, она имеет открытый код, что важно для разработчика, девайсы под управлением данной ОС значительно дешевле и доступнее, а также имеет большое количество разнообразных инструментов для решения поставленной задачи.

Так же Android позиционирует себя как открытая мобильная платформа, что является преимуществом при разработке приложения для данной платформы, так как не возникнут проблемы с установкой с неизвестного источника.

Так как в приложении будут присутствовать трехмерные объекты и у них должны быть физические свойства (размер, вес) - нужно выбрать фреймворк. Существует множество различных фреймворков для разработки трехмерных игр для мобильных устройств, работающих на операционной системе Android. Рассмотрим самые популярные из них.

- Unity,
- LibGDX,
- Unreal Engine.

Unity

Unity — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие [22].

Отличительными преимуществами Unity являются простой Drag&Drop интерфейс и установка плагинов в среде разработки, благодаря чему можно производить отладку игры прямо в редакторе.

Для написания скриптов движок использует язык программирования C#. Для 3D физики все расчеты производит физический движок от NVIDIA - PhysX. За расчет физики в 2D отвечает – Vox2D.

Как правило, игровой движок предоставляет множество функциональных возможностей, позволяющих их задействовать в различных играх, в которые входят моделирование физических сред, карты нормалей, динамические тени и многое другое [21].

Основные преимущества игрового движка Unity:

- Межплатформенная поддержка – учитывает не только места развертывания (персональный компьютер, мобильное устройство,

консоль и т.д.), но также и наличие инструментария разработки (IDE может использоваться на устройствах под управлением Windows или Mac OS).

- Наличие визуальной среды разработки – включает в себя инструментарий визуального моделирования, интегрированную среду, автоматический инструмент сборки, который повышает производительность разработчиков, в частности, на этапах прототипирования и тестирования.
- Модульная система компонентов – с её помощью производится конструирование игровых объектов, которые представляют собой комбинируемые пакеты функциональных элементов. В Unity объекты создаются с помощью объединения функциональных блоков, а не размещением их в узлах дерева наследования. Данный подход упрощает создание прототипов.

Не смотря на все свои преимущества Unity имеет ряд недостатков:

- Ограничение визуального редактора при работе с многокомпонентными схемами, когда в сложных сценах визуальная работа затрудняется [32].
- В Unity отсутствует возможность ссылаться на внешние библиотеки, что усложняет работу для программистов, в том числе и командную, т.к. эти библиотеки приходится настраивать самостоятельно.
- Используемый API для 3D-графики в браузерах – WebGL обладает рядом проблем с производительностью, повышенным потреблением памяти и надёжной и бесперебойной работой на мобильных устройствах.

LibGDX

libGDX – кроссплатформенный фреймворк для разработки игр и визуализации, основанный на языке программирования Java с некоторыми

компонентами, написанными на C и C++ для повышения производительности определенного кода. В настоящее время поддерживает Windows, Linux, Mac OS X, Android, iOS и HTML5 как целевые платформы [17].

В LibGDX реализована кроссплатформенность, которая позволяет написать код однажды и после развертывать приложение для различных платформ без предварительных модификаций. В данном фреймворке реализована полная поддержка языка программирования Java, что позволяет использовать полный инструментарий данного языка.

OpenGL ES 3.0 – это программный интерфейс для рендеринга сложной 3D-графики для мобильных и встроенных устройств. OpenGL ES – это основная графическая библиотека для мобильных и встроенных устройств с программируемой 3D-графикой, включая мобильные телефоны, PDA, консоли, транспортные средства и т. п. [25]. Данный интерфейс позволяет получить прямой доступ к файловой системе, аудио устройствам, устройствам ввода, а также возможность работать напрямую с спецификацией OpenGL.

Поверх низкоуровневых возможностей предлагается колоссальный набор API, который позволяет решать стандартные задачи, возникающие при разработке игр:

- Отображение на экране спрайтов, текста
- Построение пользовательских интерфейсов
- Воспроизведение звуковых файлов
- Математические вычисления
- Работа с форматами JSON и XML и т.д.

LibGDX способен перейти от Java к нативному коду, что позволяет повысить производительность при работе приложения. Данный функционал реализован с помощью Java API, поэтому пользователь не должен беспокоиться о кросс-компилировании нативного кода на всех платформах.

LibGDX больше позиционирует себя как фреймворк, нежели движок. Разработчики придерживаются мысли, что нет универсального решения, пользователь сам в праве решать, как реализовать сложившуюся у него задачу.

LibGDX поддерживает работу с сторонними библиотеками и фреймворками:

- `gdxAI` – это фреймворк искусственного интеллекта, полностью написанный на Java, для разработки игр [14]. Изначально предполагалось использование только с фреймворком LibGDX, но после версии 1.4.1 был отделен от основной библиотеки в собственный репозиторий.
- `gdx freetype` – используется для работы с FreeType. FreeType, – библиотека, которая используется для растеризации шрифтов и операций над ними. В частности, используется для отображения шрифтов в X11 [13].
- `Box2D` – свободный открытый физический движок. Box2D является физическим движком реального времени и предназначен для работы с двумерными физическими объектами. Движок разработан Эрином Катто (англ. Erin Catto), написан на языке программирования C++ и распространяется на условиях лицензии MIT. [12]
- `Packr` – упаковывает JAR, ресурсы и JVM для распространения на Windows, Linux и macOS, добавляя собственный исполняемый файл, чтобы он выглядел как собственное приложение [20]. Благодаря этому инструменту пользователю не требуется установка собственной JRE.
- `Hyperlap2D` – визуальный редактор общего назначения, который помогает описывать сложные сцены с помощью простого в использовании пользовательского интерфейса Drag&Drop и множеством полезных инструментов [15]. Поставляется с

официальной средой выполнения для LibGDX, библиотеки, в которой есть инструментарий для анализа файлов JSON и работы с сценами.

- OpenGL – спецификация, определяющая платформонезависимый (независимый от языка программирования) программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику. [19]
- OpenAL – кроссплатформенный интерфейс программирования приложений (API) для работы с аудиоданными. Ключевой особенностью является работа со звуком в 3D-пространстве и использование эффектов EAX. Поддерживается компанией Creative. [18]

Unreal Engine

Unreal Engine — игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. Первой игрой на этом движке был шутер от первого лица Unreal, выпущенный в 1998 году. Хотя движок первоначально был предназначен для разработки шутеров от первого лица, его последующие версии успешно применялись в играх самых различных жанров, в том числе стелс-играх, файтингах и массовых многопользовательских ролевых онлайн-играх. [23]

Движок написан на языке программирования C++, позволяет создавать игры для большинства операционных систем и платформ:

- Microsoft Windows,
- Linux,
- Mac OS

консолей:

- Xbox 360/One,
- PlayStation 2/3/4/5,
- PSP,

- PS Vita,
- Wii,
- Dreamcast,
- GameCube

и др., а также для различных устройств под управлением систем iOS или Android. [24]

Основные преимущества игрового движка Unreal Engine:

- Имеет лучшие возможности для работы с 3D графикой, по сравнению с конкурентами
- Развивает систему визуального программирования. На начальных этапах разработчику не требуется установка дополнительных сред и библиотек.
- Высокая оптимизация движка.

Не смотря на все свои преимущества Unreal Engine имеет ряд недостатков:

- Высокие требования к системе. Для простого проекта нет смысла использовать столь мощный движок.
- Высокий порог вхождения. Требуется знания языка программирования C++.

Выбор фреймворка

После изучения возможностей различных фреймворков, был выбран LibGDX, он полностью подходит для решения поставленной задачи, его функционала полностью хватит для поставленных задач. Имеет в наличии продвинутые библиотеки и является менее требовательным в сравнении с Unreal Engine и Unity.

1.3 Формализованное описание технического задания

на разработку информационной системы,

«Мобильное приложение для проведения психологического эксперимента по изучению особенностей восприятия дошкольников при использовании информационных устройств с сенсорным экраном обучения»

Составлено на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» [26].

1. Общие сведения.

1.1. Название организации-заказчика.

ФГБОУ ВО «Уральский Государственный Педагогический Университет»

1.2. Название продукта разработки (проектирования).

«Мобильное приложение для проведения психологического эксперимента по изучению особенностей восприятия дошкольников при использовании информационных устройств с сенсорным экраном обучения»

Рабочее название продукта – «PsychoTest»

1.3. Назначение продукта.

Предназначен для преподавателей института Психологии, кафедры общей психологии и конфликтологии, Уральского государственного педагогического университета, для проведения исследования «Распознавание виртуальных моделей объектов»

1.4. Плановые сроки начала и окончания работ.

В соответствии с планом выполнения ВКР (01.09.2021 – 21.05.2022).

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Данный продукт будет применяться в детских садах преподавателями института Психологии, кафедры общей психологии и конфликтологии,

Уральского государственного педагогического университета в рамках исследования.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

Разработчик – должен знать язык программирования Java, уметь работать с межплатформенной средой разработки компьютерных игр LibGDX, создавать объёмные модели в открытом ПО для создания компьютерной трехмерной графики Blender.

Пользователь – должен уметь работать с устройством на базе OS Android.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Разработать приложение для операционной системы Android. Приложение должно содержать два экрана: экран для манипуляции трехмерными объектами и экран «кубик», на котором надо будет переместить куб на полку. При перемещении куба на полку создавать файл, в котором будут записаны его координаты и пройденное расстояние за каждые 10мс времени перемещения куба. Куб должен иметь различные текстуры, возможность изменять размер и вес.

3.2. Аппаратные требования.

Устройство на базе OS Android с процессором на базе архитектуры ARMv7 с поддержкой Neon (32-bit) или ARM64, 1Gb и более ОЗУ, с графическим API: OpenGL ES 2.0+, OpenGL ES 3.0+

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

OS Android версия 8.0 и выше

3.4. Указание программного обеспечения, используемого для реализации.

- OS Windows
- LibGDX

- Android Studio
- Blender

3.5. Особенности реализации серверной и клиентской частей.

Не предусмотрено.

3.6. Форматы входных и выходных данных

- Входные данные: сенсорные касания по экрану устройства
- Выходные данные: изменение отображаемых данных (перемещение/вращение модели, смена экрана, изменение настроек приложения). Если было перемещение куба на полку, то создается файл с кинематическим профилем движения объекта.

3.7. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

Для данного приложения потребуются файлы 3D моделей и текстур, который будут храниться в специальной папке “Assets”.

3.8. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

3.9. Меры защиты информации.

Не предусмотрено.

4. Требования к пользовательскому интерфейсу.

4.1. Общая характеристика пользовательского интерфейса.

WIMP интерфейс.

Интерфейс должен быть интуитивно понятен, окна для проведения тестов должны быть максимально освобождены от различных элементов интерфейса, чтоб там мог работать без особых затруднений - ребенок детсадовского возраста.

4.2. Размещение информации на экране, дизайн экрана.



Рис. 1. Эскиз главного экрана приложения

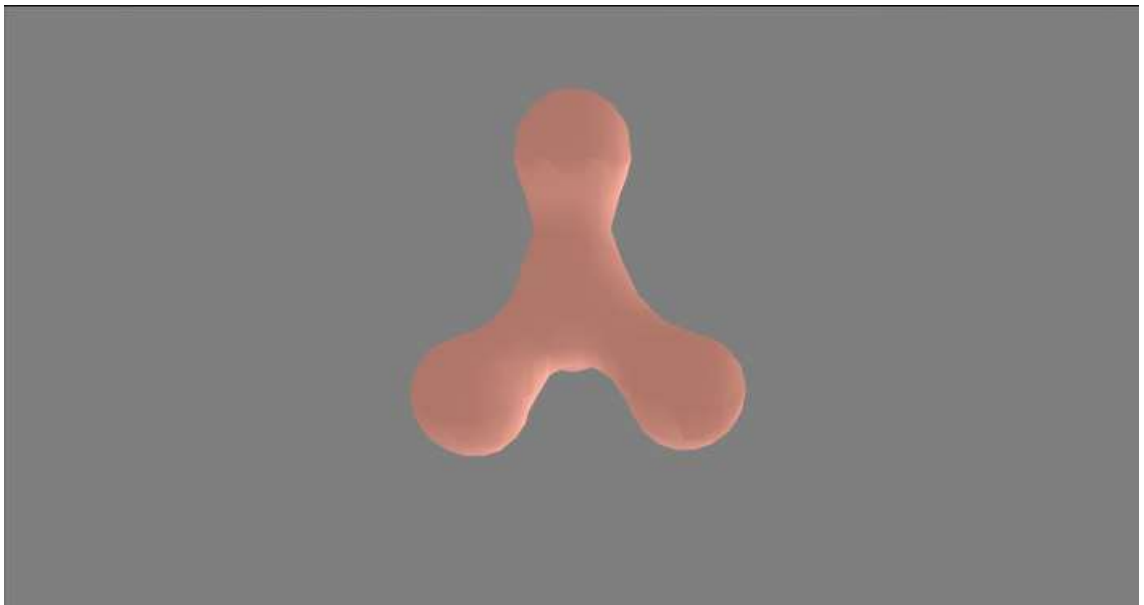


Рис. 2. Эскиз экрана «Модельки»

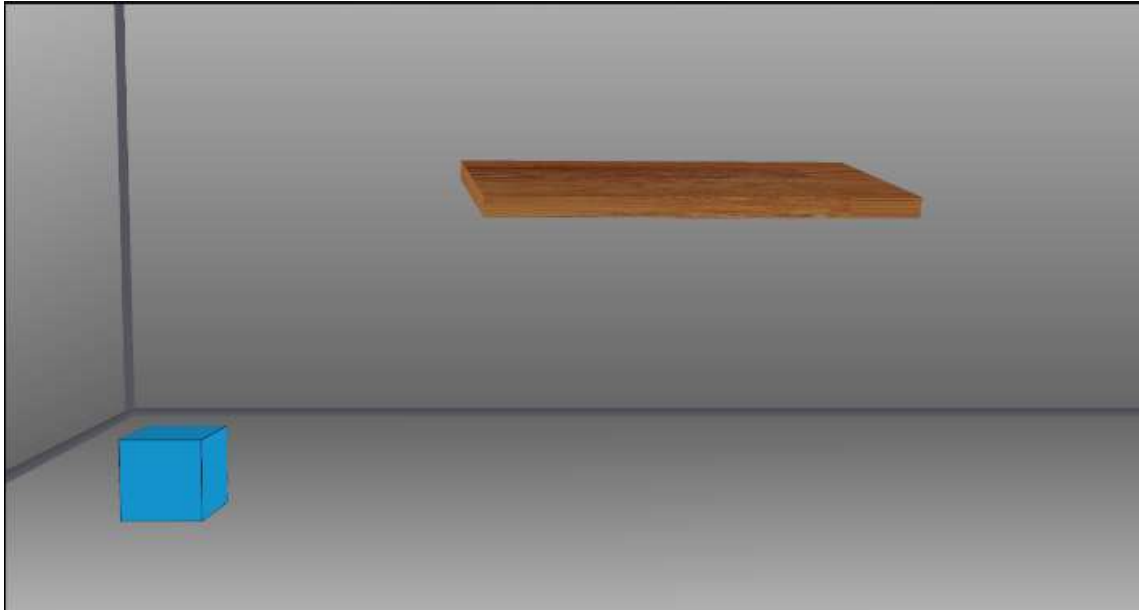


Рис. 3. Эскиз экрана «Кубик»



Рис. 4. Эскиз экрана настроек приложения

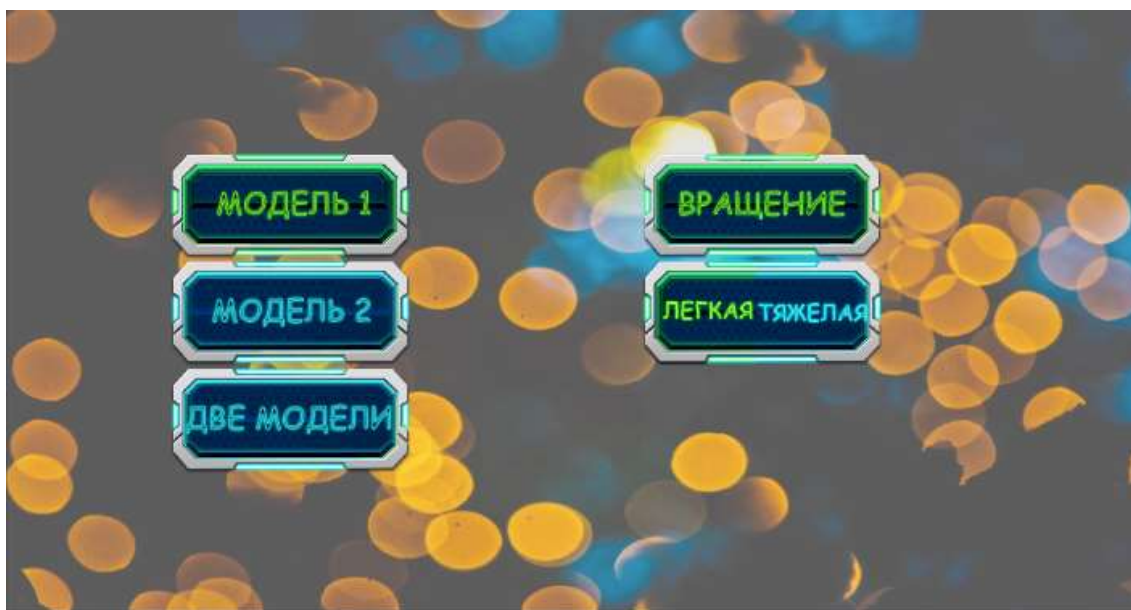


Рис. 5. Эскиз экрана настроек для окна «Модельки»



Рис. 6. Эскиз экрана настроек для окна «Кубик»

4.3. Особенности ввода информации пользователем, представление выходных данных.

Все взаимодействия происходят с помощью сенсорного экрана устройства.

При нажатии по экрану в области кнопок будет смена экрана или смена пункта настроек.

При движении пальца по экрану будет производиться вращение или перемещение модели.

При нажатии системной кнопки «назад» приложение будет возвращаться к предыдущему экрану.

При установке куба на полку создавать файл с кинематическим профилем движения объекта за каждые 10мс времени перемещения куба.

x	y	time	timeNormal	distance
6	0	10	0m0s.10ms	0
6	0	20	0m0s.20ms	0
16	0	30	0m0s.30ms	100
47	8	40	0m0s.40ms	1025

Рис. 7. Фрагмент файла с кинематическим профилем движения

5. Требования к документированию.

5.1. Перечень сопроводительной документации.

Руководство пользователя приложения «Psycho Test».

5.2. Требования к содержанию отдельных документов.

Не предусмотрено.

6. Порядок сдачи-приемки продукта.

В соответствии с планом выполнения ВКР.

Глава 2. Разработка мобильного приложения средствами межплатформенной среды LibGDX для выполнения манипуляций с трехмерными объектами

2.1 Модель мобильного приложения

Перед началом разработки программного обеспечения необходимо получить сведения о структуре, описать функциональные требования к системе, а также решить каким образом система будет взаимодействовать с пользователем.

Основу методологии IDEF0 составляет графический язык описания процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе. Одной из наиболее заметных особенностей нотации IDEF0 является явное разграничение материальных и информационных потоков на входные, выходные, управляющие и инструментальные. Благодаря этой особенности описания отдельных функций и всего процесса в целом получают достаточно полными и четкими.

В процессе проектирования была разработана функциональная модель IDEF0, и её декомпозиция, которые описывают основные части работы приложения и их взаимодействие между собой.

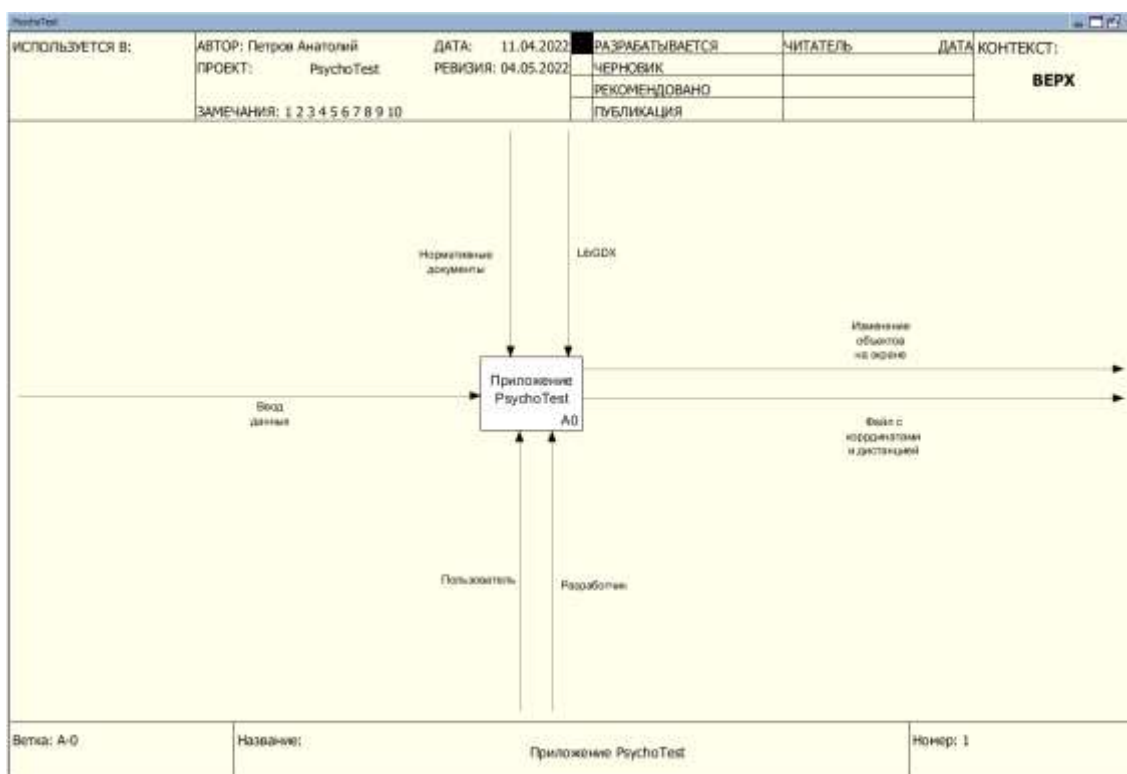


Рис. 8. Функциональная модель «Мобильного приложения для проведения психологического эксперимента по изучению особенностей восприятия дошкольников при использовании информационных устройств с сенсорным экраном обучения»

Функциональная модель приложения

На вход в систему подаются данные, введенные пользователем с сенсорного экрана устройства.

Выходом является результат работы приложения – различные манипуляции с трехмерными объектами, смена экрана, изменение настроек, файл с кинематическим профилем движения объекта.

Управляющими элементами являются:

- Нормативные документы;
- Фреймворк LibGDX, работающий на языке Java, обрабатывающий и преобразующий входные данные.

Механизмами системы являются:

- Пользователь, управляющий объектами, представленными системой;
- Разработчик, настраивающий функционал для обработки действий пользователя;

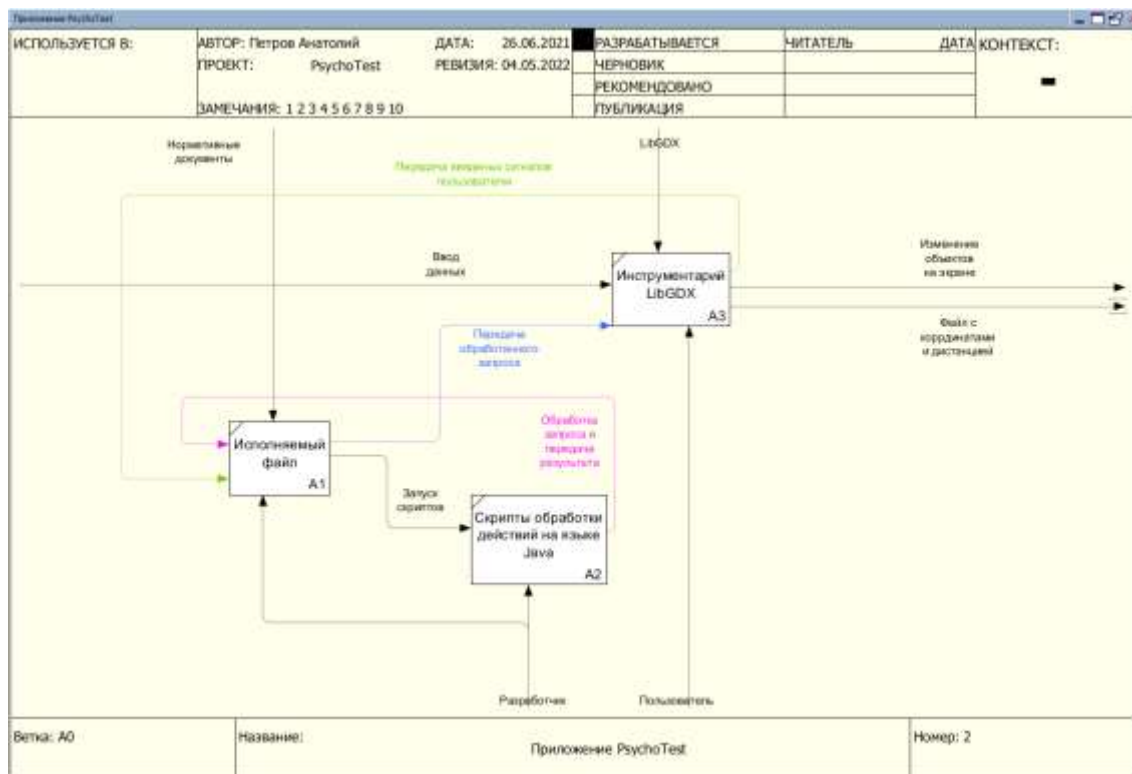


Рис. 9. Декомпозиция функциональной модели «Мобильного приложения для проведения психологического эксперимента по изучению особенностей восприятия дошкольников при использовании информационных устройств с сенсорным экраном обучения»

На данной модели представлены три блока:

- Исполняемый файл – принимает различные сигналы и распределяет их между другими компонентами программы;
- Скрипты обработки действий на языке Java- обрабатывает введенные данные и возвращает результат в исполняемый файл;
- Инструментарий LibGDX – получает запрос и передает его исполняемому файлу, после получает обработанный запрос и на основе полученных данных выполняет программу.

2.2 Описание мобильного приложения для изучения особенностей зрительного и тактильного восприятия при использовании дошкольниками информационных устройств с сенсорным экраном «PsychoTest»

В итоге было разработано «Мобильное приложение для проведения психологического эксперимента по изучению особенностей восприятия дошкольников при использовании информационных устройств с сенсорным экраном обучения», с использованием фреймворка LibGDX, который имеет большую базу различных средств для работы с 3D.

Приложение позволяет испытуемым манипулировать моделями трехмерных объектов с помощью сенсорного экрана.

При выборе режима «Модельки» приложение предоставляет возможность исследования трехмерной фигуры с целью дальнейшей идентификации реального предмета.

При выборе режима «Кубик» на основном экране изображено трехмерное помещение с расположенной на стене полкой и кубом, размещенным на «полу». Задача испытуемого: перенести куб на полку. При этом приложение сохраняет кинематический профиль движения объекта во внешней памяти устройства в виде текстового файла. Файл состоит из строк, каждая из которых содержит четыре действительных чисел: первое число – время, третье и четвертое – координаты X и Y центра тяжести куба, четвертое число – перемещение, которое совершил куб. Точкой отсчета является начальное положение куба. Координаты записываются с момента начала движения на протяжении всего времени перемещения куба с интервалом 0.1 секунды до момента установки куба на полку. Поскольку каждый испытуемый имеет возможность сделать несколько попыток, а количество испытуемых может быть любым, создаваемые имена генерируемых приложением файлов содержат время испытания.

При постановке кубика на полку приложение генерирует звуковой сигнал, свидетельствующий об успешном выполнении задания. Если испытуемый прерывает контакт с экраном до постановки кубика на полку или перемещает кубик в точку, не совпадающую с полкой, то кубик возвращается в исходное положение в нижней левой части экрана. Файл при этом не создается.

Разработанное приложение позволяет выбирать вес куба (легкий/тяжелый), размер куба (малый/большой) и цвет (синий/прозрачный) – выбор нужной модели осуществляется с помощью меню настроек. При выполнении действий с легким кубом траектория его движения в точности совпадает с траекторией движения пальца испытуемого по экрану устройства. Перемещение тяжелого куба реализовано несколько иначе. Поскольку устройства, работающие под управление операционной системы Android, не способны распознавать силу нажатия на экран (не реализована технология 3D Touch), то при манипуляции тяжелым кубом его перемещение запаздывает относительно траектории движения пальца испытуемого по экрану устройства. Величина задержки подобрана таким образом, чтобы у испытуемого складывалось впечатление о том, что для перемещения куба нужно приложить некоторое заметное усилие (см. *Листинг 1*). Сохраняемый при этом кинематический профиль движения фигуры содержит данные о перемещении именно куба (не пальца испытуемого).

```

float stepX = Gdx.graphics.getDeltaTime() * (input.getSetX() - positionModelForShelf.x);
float stepY = Gdx.graphics.getDeltaTime() * (input.getSetY() - positionModelForShelf.y);

if ((positionModelForShelf.x <= -input.getCoefX() / 2 || positionModelForShelf.x >=
input.getCoefX() / 2)) {
    stepX = 0;
}
if ((positionModelForShelf.y <= -input.getCoefY() / 2 || positionModelForShelf.y >=
input.getCoefY() / 2)) {
    stepY = 0;
}

if (input.getModelDraggedInShelf() && !input.getModelInShelf()) {
    ShelfManager.getModel("cubeForShelf").transform.translate(stepX, stepY, 0);
    positionModelForShelf = modelForShelf.transform.getTranslation(new Vector3());
    ShelfManager.getModel("cubeForShelf").transform.setTranslation(positionModelForShe
lf.x, positionModelForShelf.y, input.getSetZ());
}
break;

```

Листинг 1. Перемещение тяжелого куба

В данном приложении реализованы 6 экранов:

- MenuScreen – экран с главным меню. На нем можно выбрать следующий экран: «модельки», «кубик», «настройки».

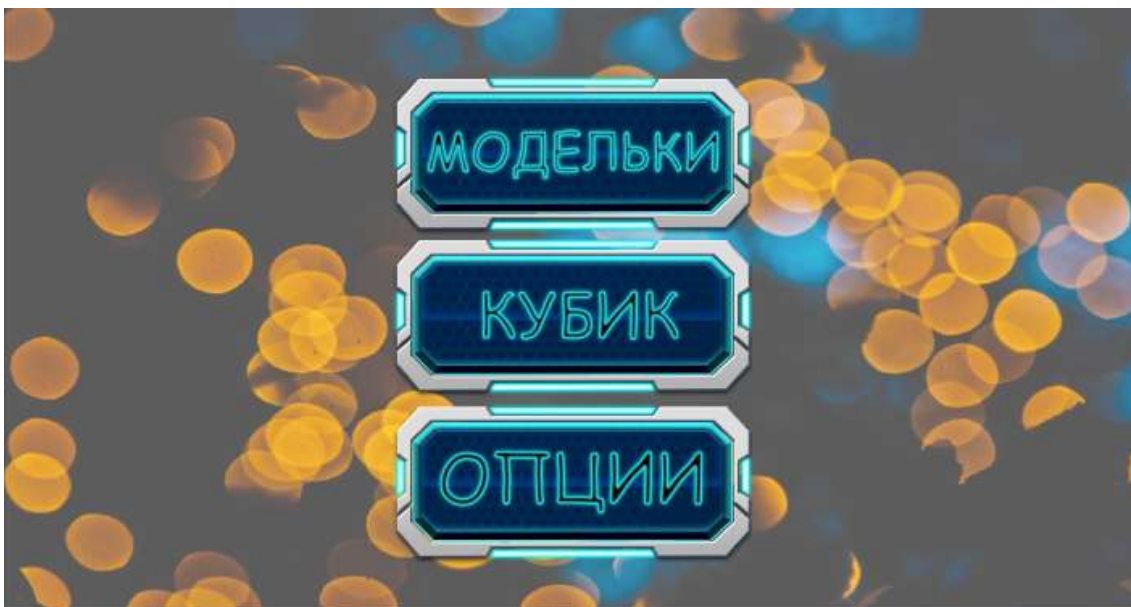


Рис. 10. Сцена SampleScene

- OptionsScreen – экран настроек. На экране настроек производится выбор экрана, для которого настройки будут изменяться.

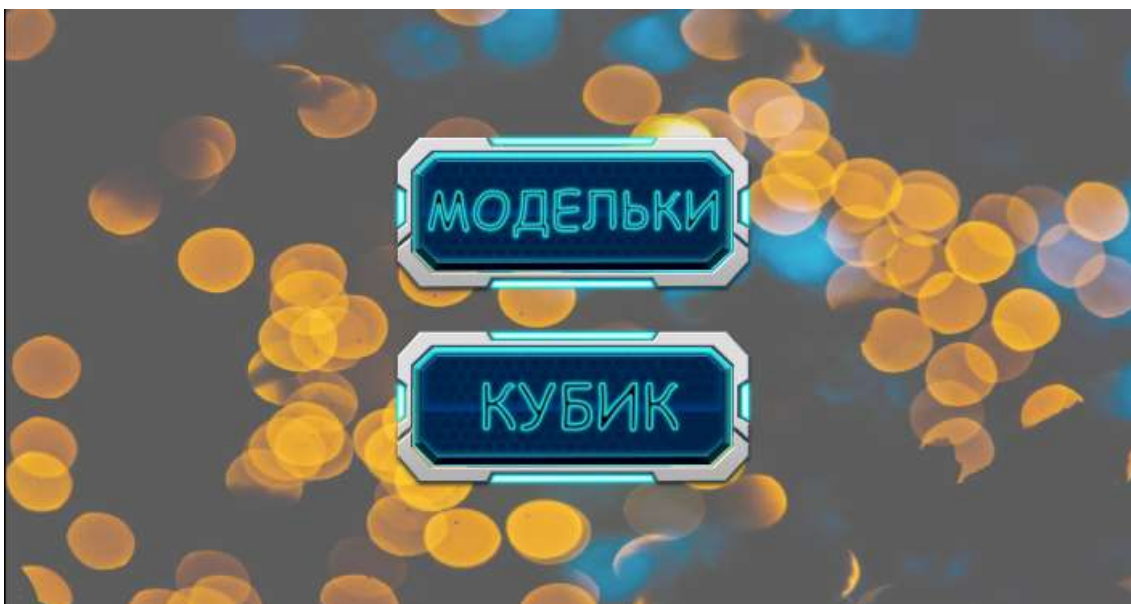


Рис. 11. Экран настроек

- ModelsOptionsScreen – экран настроек для экрана «Модельки». На этом экране настроек выбирается вес у каждой модели, выбирается

какая модель будет отображаться на экране «Модельки», а также какие манипуляции будут проводиться с моделью на экране – перемещение или вращение.

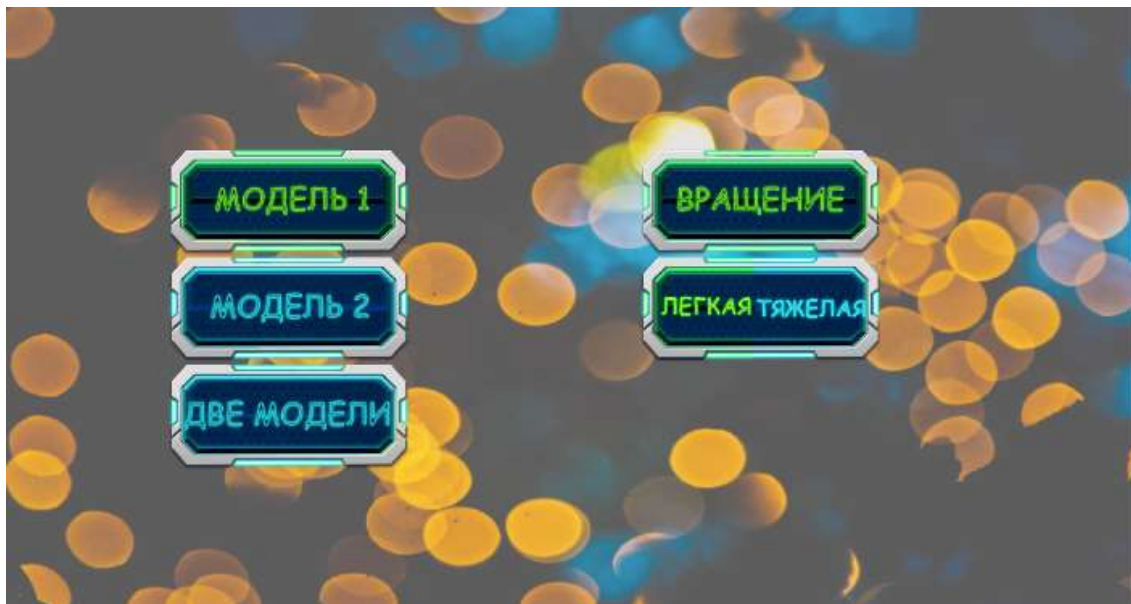


Рис. 12. Экран настроек для экрана «Модельки»

- ShelfOptionsScreen – экран настроек для экрана «Кубик». На этом экране настроек выбирается текстура кубика (Синий/Прозрачный), его вес (Легкий/Тяжелый) и размер (Маленький/Большой).



Рис. 13. Экран настроек для экрана «Кубик»

- PsychoGame – экран «Модельки». На этом экране отображается одна из выбранных моделей или обе сразу. Одиночную модель можно вращать или перемещать. Если модели 2 на экране, то их можно только вращать.

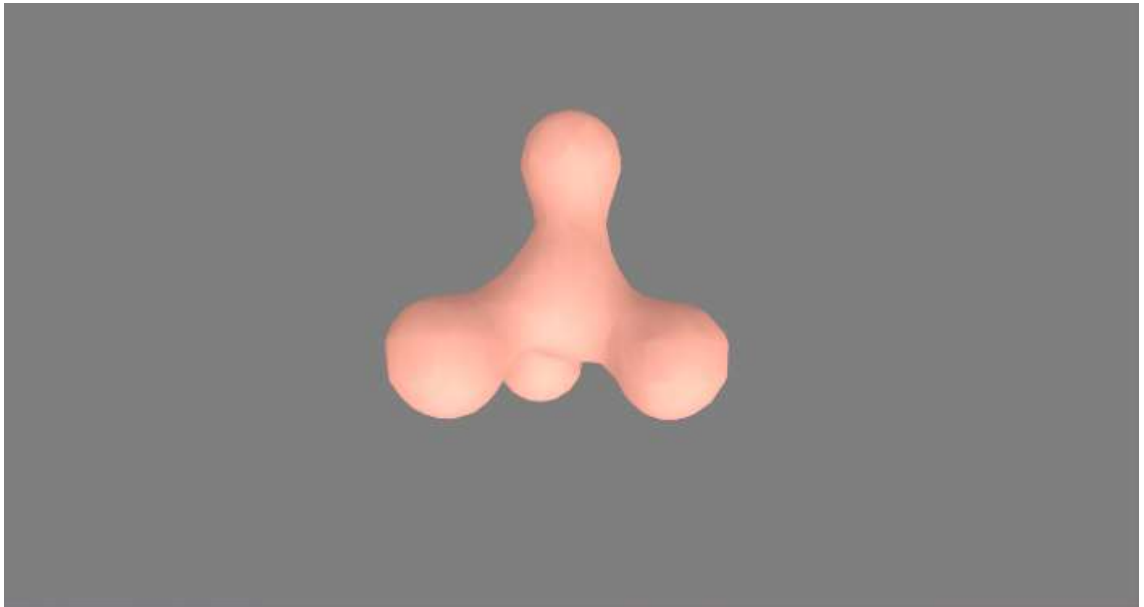


Рис. 14. Экран PsychoGame с выбранным режимом – Модель 1

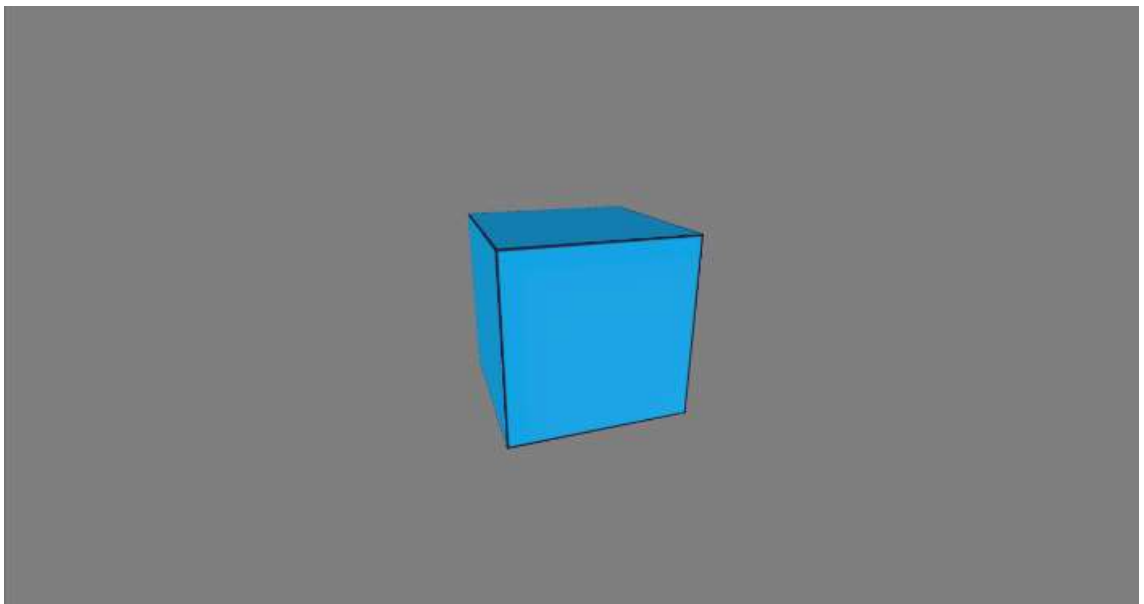


Рис. 15. Экран PsychoGame с выбранным режимом – Модель 2

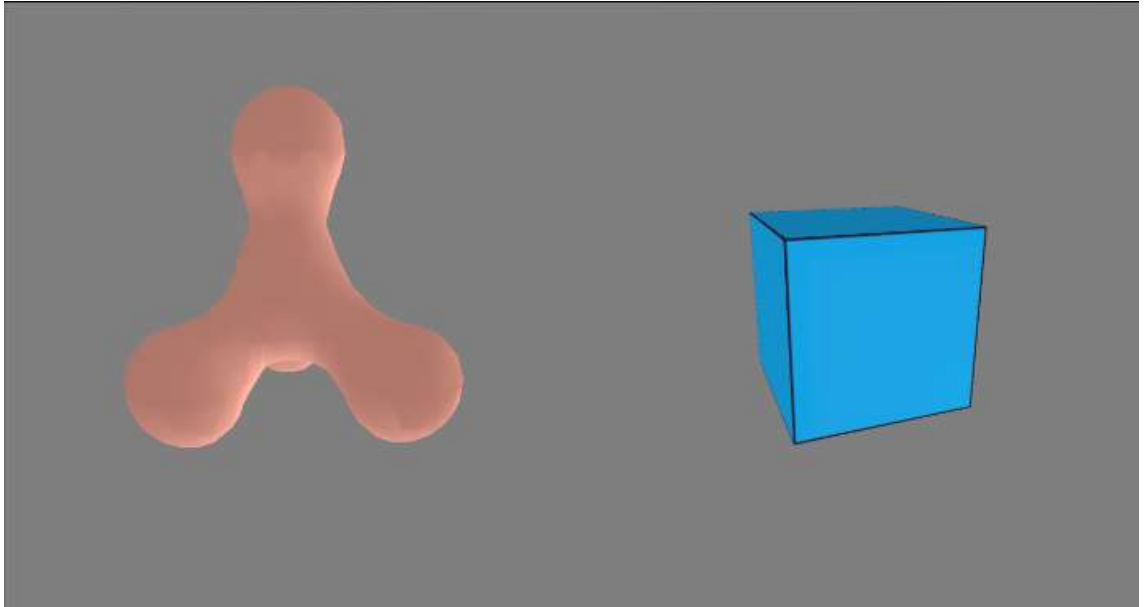


Рис. 16. Экран PsychoGame с выбранным режимом – Две модели

- GameShelf – экран «Кубик». На этом экране отображаются куб и полка. Здесь требуется установить куб на полку. При установке на полку создается файл с кинематическим профилем движения объекта за каждые 10мс времени перемещения куба. Для куба предлагаются 2 текстуры «Синяя» и «Прозрачная». Если куб тяжелый, то меняется текстура у полки на более темную. Куб может быть легким или тяжелым, большим или маленьким.

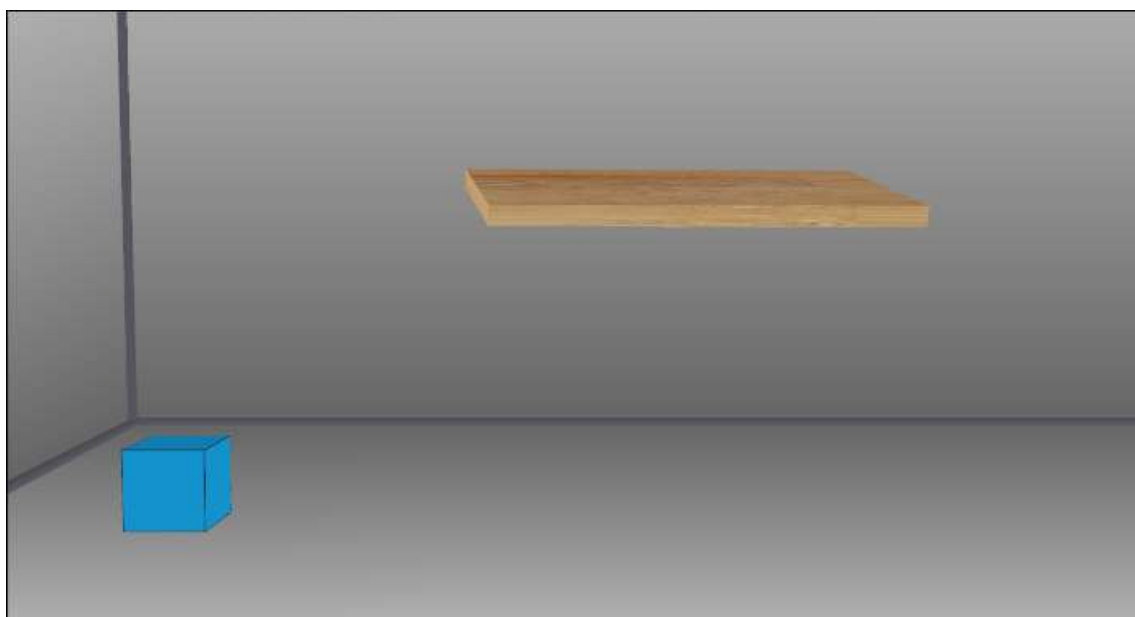


Рис. 17. Экран GameShelf с выбранной текстурой – Синий, весом модели – Легкая и размером - Маленький

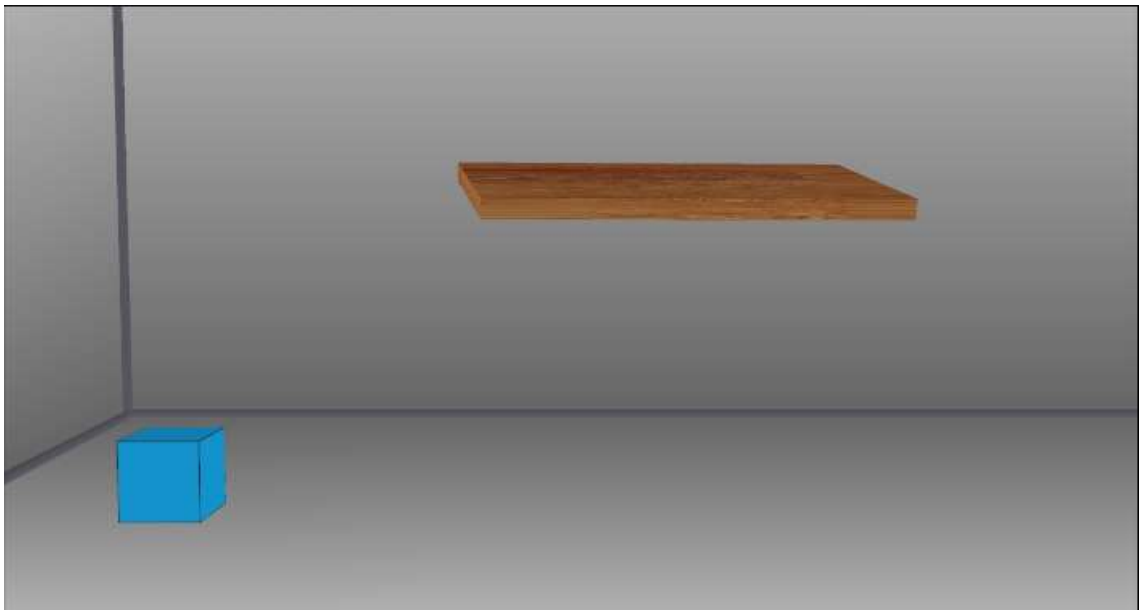


Рис. 18. Экран GameShelf с выбранной текстурой – Синий, весом модели – Тяжелая и размером - Маленький

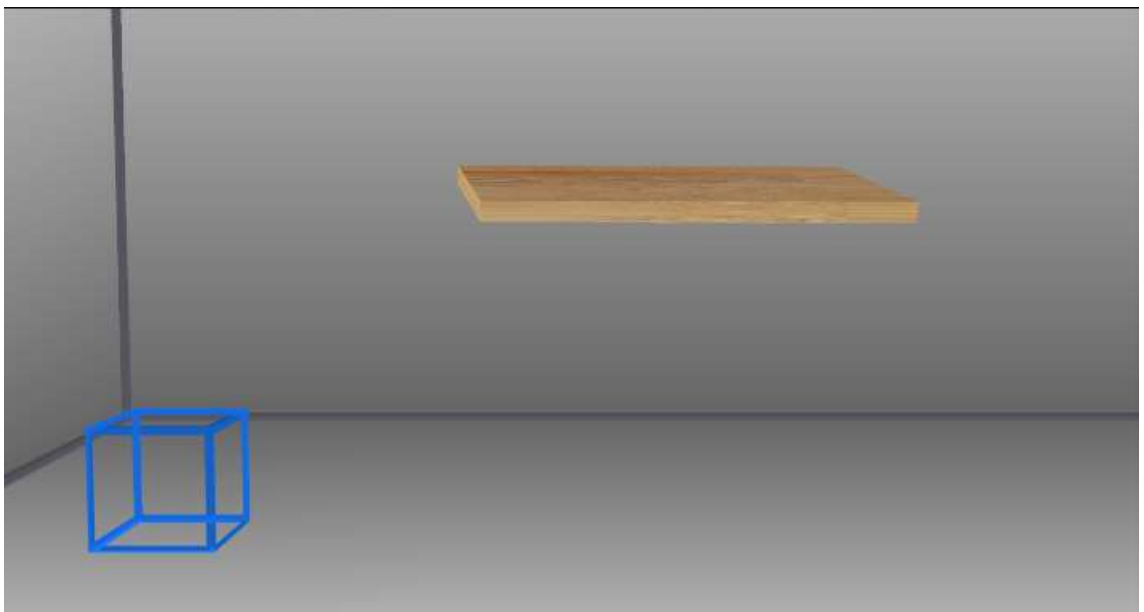


Рис. 19. Экран GameShelf с выбранной текстурой – Прозрачный, весом модели – Легкая и размером - Большой

В приложении реализован ряд классов – менеджеры.

GameManager – менеджер для экрана «Модельки». Уменьшает количество кода для работы экрана «Модельки» с разными настройками. Позволяет в зависимости от выбранных моделей создать нужные на экране.

```
static void setCube() {
    envCube = new Environment();
    cCube = new Color(0.201f, 0.201f, 0.201f, 1);
    envCube.add(new DirectionalLight().set(cCube, 0f, 200f, 0f));
    envCube.add(new DirectionalLight().set(cCube, 200, 200, 200));
    envCube.add(new DirectionalLight().set(cCube, -200, -200, -200));
    envCube.add(new DirectionalLight().set(cCube, 10, 0, 10));
    envCube.add(new DirectionalLight().set(cCube, -10, 0, -10));
    envCube.add(new DirectionalLight().set(Color.WHITE, 200, 100, 200));
    envCube.set(new ColorAttribute(ColorAttribute.AmbientLight, 0.755f, 0.722f, 0.722f, 1));
    ObjLoader loader = new ObjLoader();
    modelCube = new ModelInstance(loader.loadModel(Gdx.files.internal("cube.obj")));
    textureCube = new Texture("rebra.001.png");
    modelCube.materials.get(0).set(TextureAttribute.createDiffuse(textureCube));
    modelCube.transform.scl(2);
    model = modelCube;
}
```

Листинг 2. Подключение «Модель 2» на экране «Модельки»

Представленный код (см. *Листинг 2*) создает «кубик» на экране «Модельки». Похожий код разработан и для объекта «Модель 1». Если пользователь выберет в настройках «Две модели», то достаточно будет вызвать эти два метода (см. *Листинг 3*), а не прописывать код создания для каждого объекта заново.

```
static void setTwoModel() {  
    setMessage();  
    setCube();  
}
```

Листинг 3. Вызов методов для создания двух моделей на одном экране

ShelfManager – менеджер для экрана «Кубик». Разработан для работы с каждым объектом на экране. Отрисовывает не только кубик и полку, а также градиентный фон и границы комнаты. После инициализации позволяет с помощью метода `renderGame()` (см. *Листинг 4*) обновлять объекты на экране.

```
public static void renderGame(ModelBatch batch, String nameModel) {  
    switch (nameModel) {  
        case "cubeForShelf":  
            batch.render(modelCubeForShelf, envCubeForShelf);  
            break;  
        case "shelf":  
            batch.render(modelShelf, envShelf);  
            break;  
        case "axisX":  
            batch.render(modelX, envX);  
            break;  
        case "axisY":  
            batch.render(modelY, envY);  
            break;  
        case "axisZ":  
            batch.render(modelZ, envZ);  
            break;  
    }  
}
```

Листинг 4. Метод обновления объектов на экране `renderGame()`

PrefsManager – менеджер настроек для приложения. Менеджер состоит из главного метода, который задает стандартные настройки, если приложение запущено впервые, а также инициализирует внутренние поля. В менеджере настроек реализован набор «сеттеров» и «геттеров» (пример см. *Листинг 5*) для удобства работы с настройками из разных мест кода.

```
public void setWtMessage(String wtMessage){
    this.wtMessage = wtMessage;
    pref.putString("wtMessage",wtMessage);
    pref.flush();
}

public String getWtMessage(){
    return wtMessage;
}
```

Листинг 5. «Сеттер»и «геттер» для веса «Модели 1»

InputManager – менеджер управления для приложения. Менеджер реализовывает вращение и перемещение моделей на экране «Модельки». Для экрана «Кубик» рассчитывает местоположение пальца на экране, учитывая область нажатия по кубику, рассчитывает координаты для кубика в зависимости от местоположения пальца с учетом границ экрана (см . Листинг 6)

```
public void setSetXSetY(int screenX, int screenY) {
    setX = ((Gdx.graphics.getWidth() / 2) - screenX) / coeffScreenX;
    setY = ((Gdx.graphics.getHeight() / 2) - screenY) / coeffScreenY;
    if ((screenX > (Gdx.graphics.getWidth() / 2) && setX < 0) || (screenX <
(Gdx.graphics.getWidth() / 2) && setX > 0)) {
        setX = -1 * setX;
    }
    if ((screenY > (Gdx.graphics.getHeight() / 2) && setY > 0) || (screenY <
(Gdx.graphics.getHeight() / 2) && setY < 0)) {
        setY = -1 * setY;
    }

    if (setX > coefX / 2 || setX < -coefX / 2) {
        setX = positionModelForShelf.x;
    }
    if (!(positionModelForShelf.y >= -coefY / 2 || setY > 0) && (positionModelForShelf.y
<= coefY / 2 || setY < 0)) {
        setY = positionModelForShelf.y;
        if (setY > 0) {
            setY -= 0.1f;
        } else {
            setY += 0.1f;
        }
    }
}
```

Листинг 6. Расчет x и y для установки кубика

2.3 Результаты апробации, техническая документация

Руководство пользователя

1. Требования к аппаратному и программному обеспечению.

Устройство должно отвечать следующим аппаратным и программным требованиям:

- OS Android версия 8.0 и выше;
- Оперативная память 1 Gb и более;
- Сенсорный экран;

2. Запуск мобильного приложения

Запуск мобильного приложения «PsychoTest» производится с помощью иконки на рабочем столе устройства или в разделе меню.



Рис. 20. Иконка приложения «PsychoTest»

Если такой иконки нет на устройстве, следует попробовать установить приложение заново.

3. Работа с интерфейсом. Основные моменты управления.

После запуска приложения на экране отобразиться главное меню.

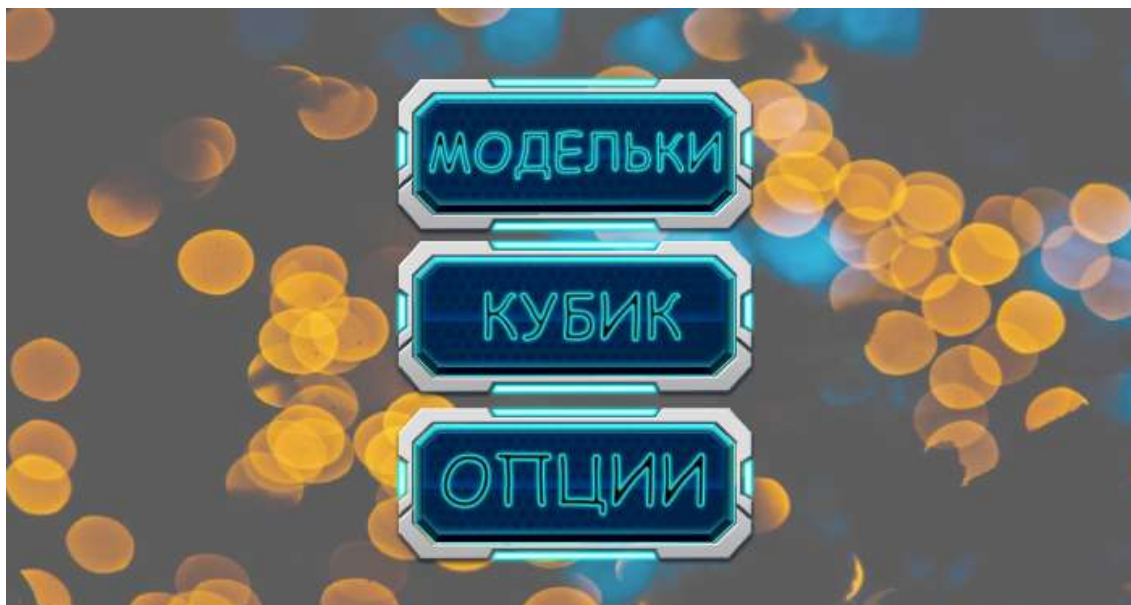


Рис. 21. Главное меню приложения «PsychoTest»

Переход между экранами осуществляется с помощью нажатия экранных кнопок. Нажатие осуществляется попаданием пальцем на экране по области кнопки. После нажатия на кнопку будет осуществлен переход на выбранный экран.

При нажатии системной кнопки «назад» приложение будет возвращаться к предыдущему экрану. Нажатие системной кнопки «назад» на экране с меню вызовет процедуру выхода из приложения.

4. Настройки приложения.

Все настройки осуществляются с помощью экрана «Опции».

После открытия экрана «Опции» следует выбрать для какого из режимов работы будут произведены настройки.

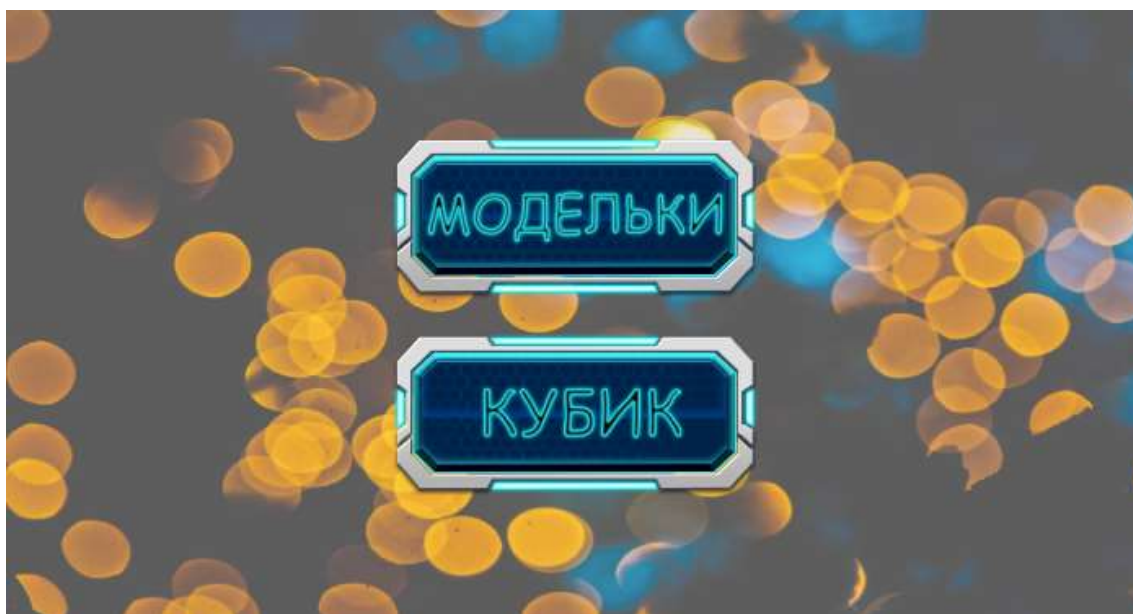


Рис. 22. Экран приложения «Опции»

1. Настройка режима работы приложения «Модельки».

Для настройки режима работы приложения «Модельки» следует нажать по кнопке «Модельки», будет открыт экран настроек для режима работы приложения «Модельки».

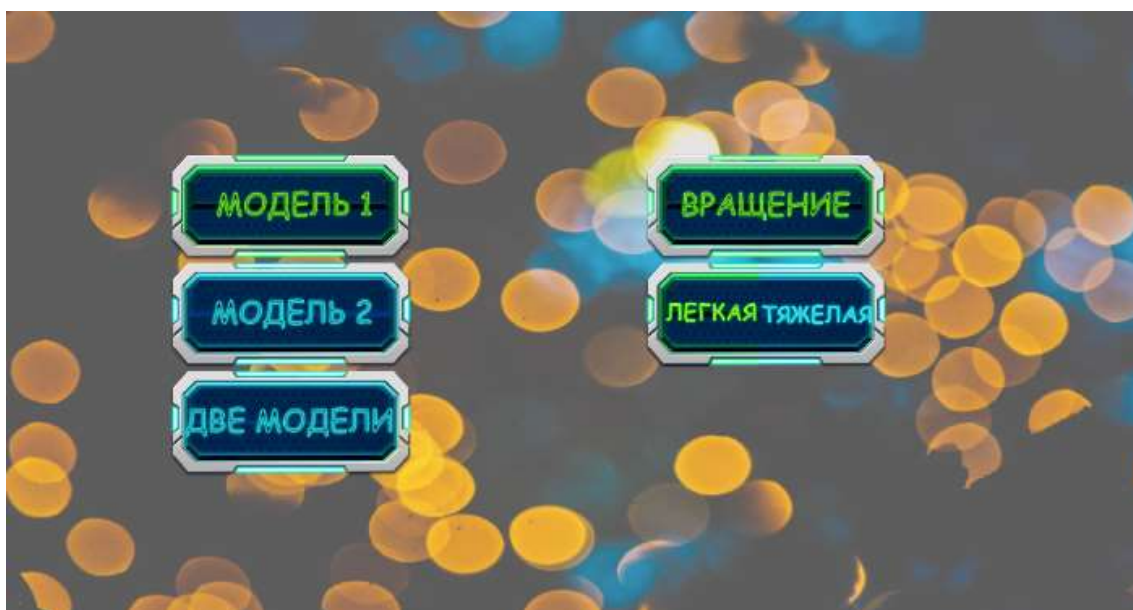


Рис. 23. Экран настроек режима работы приложения «Модельки»

На данном экране с помощью кнопок «Модель 1», «Модель 2», «Две модели» можно выбрать какие трехмерные объекты будут отображаться на

экране. При нажатии на кнопки «Модель 1» и «Модель 2» справа можно будет выбрать вес объекта, а также операции, проводимые с ним.

При активации кнопки «Вращение» объект можно будет вращать на экране, при деактивации объект можно будет перемещать по экрану.

При нажатии кнопки «Две модели» на экране будут отображаться две модели, их можно будет только вращать, вес у моделей будет такой, какой был ранее задан для каждой при нажатии на кнопки «Модель 1» и «Модель 2», правое меню будет не активным.

2. Настройка режима работы приложения «Кубик».

Для настройки режима работы приложения «Кубик» следует нажать по кнопке «Кубик», будет открыт экран настроек для режима работы приложения «Кубик».



Рис. 24. Экран настроек режима работы приложения «Кубик»

На данном экране с помощью кнопок «Синий» и «Прозрачный» происходит выбор текстуры кубика, соответственно синяя или прозрачная.

Переключатель «Маленький/Большой» задает размер куба.

Переключатель «Легкий/Тяжелый» задает вес куба.

5. Режим работы приложения «Модельки».

Для перемещения модели на экране следует поставить палец на экране в области объекта и не отрывая его переместить по экрану. Модель последует за пальцем.

Вращение модели осуществлено свайпами – движением пальцем по экрану устройства в одном направлении. Вместе со свайпом модель будет вращаться.

6. Режим работы приложения «Кубик».

Основной задачей данного режима является установка кубика на полку.

Для перемещения кубика на экране следует поставить палец на экране в области объекта и не отрывая его переместить по экрану. Кубик последует за пальцем.

Если оторвать палец от экрана, не донеся кубик, то он вернется в исходное положение.

После установки кубика на полку прозвучит звуковой сигнал и будет создан файл с кинематическим профилем движения объекта за каждые 10мс времени перемещения куба. Кубик зафиксируется на полке. Если требуется повторить эксперимент, то нужно с помощью системной кнопки «назад» вернуться на главное меню и снова запустить режим работы приложения «Кубик».

7. Файл с кинематическим профилем движения объекта за каждые 10мс времени перемещения куба.

Созданный файл будет сохраняться в папке приложения. Чтобы в нее попасть следует:

1. Открыть файловый менеджер на устройстве.
2. Открыть папку Android, затем перейти в папку data.
3. Найти папку приложения – com.psychogame, открыть ее.
4. Далее открыть папку files, а в ней будет папка PsychoTests, в этой папке хранятся созданные файлы приложением.

Структура имени файла

Имя файла состоит из:

- Даты с временем установки куба на полку;
- Указания текстуры куба: F – синий, T – прозрачный;
- Указания веса куба: H – тяжелый, L – легкий;
- Указания размера куба: B – большой, S – маленький.

Проведена апробация разработанного приложения в процессе реализации гранта РФФИ «Изучение феномена несоответствия зрительного и тактильного восприятия при использовании дошкольников устройств с сенсорным экраном» (Институт психологии и ИМФИиТ УрГПУ).

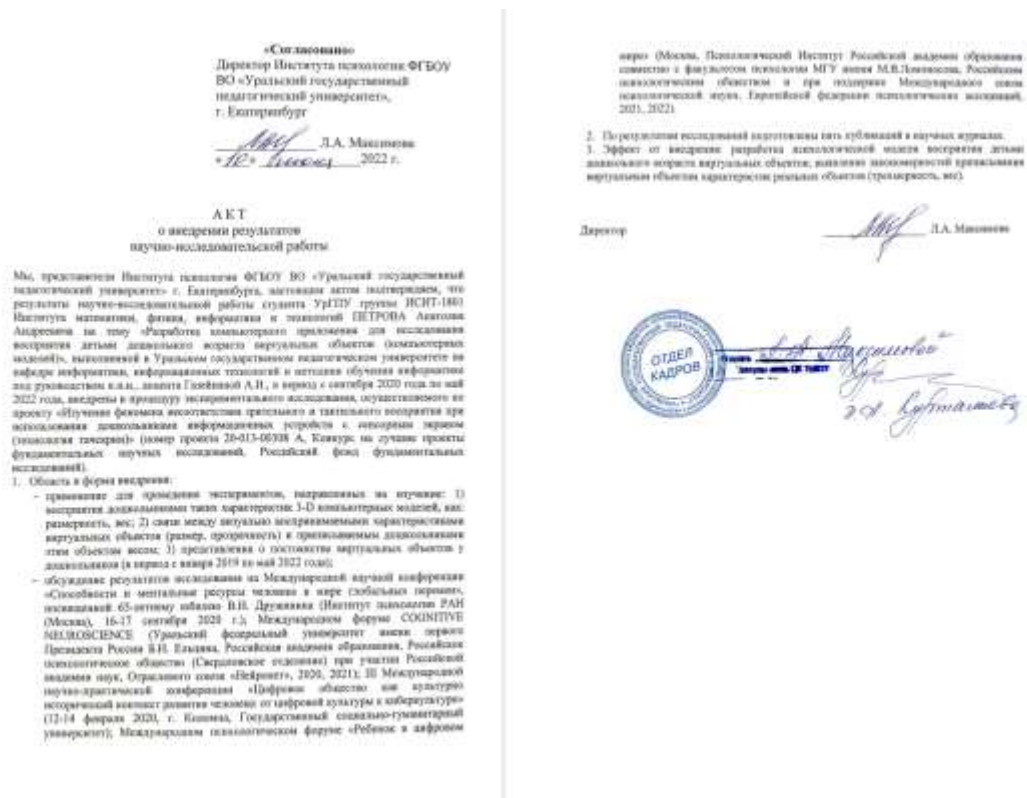


Рис. 25. Акт о внедрении результатов научно-исследовательской работы

БЛАГОДАРСТВЕННОЕ ПИСЬМО

Институт психологии ФГБОУ ВО «Уральский
государственный педагогический университет»
выражает благодарность и искреннюю
признательность

**ПЕТРОВУ
АНАТОЛИЮ АНДРЕЕВИЧУ**

студенту 4 курса Института математики, физики,
информатики и технологий
(руководитель – Газейкина А.И., канд. пед. н.,
доцент каф. информатики, информационных
технологий и методики обучения информатике)

за разработку компьютерных приложений для
проведения исследований по проекту «Изучение
феномена несоответствия зрительного и
тактильного восприятия при использовании
дошкольниками информационных устройств с
сенсорным экраном (технология тачскрин)»
Российского фонда фундаментальных
исследований (номер проекта 20-013-00308 А).

Ректор



Минюрова С.А.

2022, Екатеринбург

Рис. 26. Благодарственное письмо

Оформлены документы на получение университетом исключительного
права на результат интеллектуальной деятельности (РИД).

Заключение

В результате было разработано приложение, позволяющее пользователю манипулировать трехмерными моделями на экране планшета или мобильного телефона.

В ходе работы:

1. Изучены теоретические основы разработки мобильных приложений для ОС Android
2. Произведен анализ доступных фреймворков и выбран LibGDX в качестве фреймворка для реализации проекта.
3. В соответствии с техническим заданием произведена разработка мобильного приложения на основе выбранных технологий и программного обеспечения.
4. Разработано руководство пользователя.
5. Проведена апробация разработанного приложения в процессе реализации гранта РФФИ «Изучение феномена несоответствия зрительного и тактильного восприятия при использовании дошкольников устройств с сенсорным экраном» (Институт психологии и ИМФИиТ УрГПУ).

При разработке приложения был изучен и использовался фреймворк LibGDX, предназначенный для создания кроссплатформенных игр и приложений.

Весь выбранный инструментарий относится к непроприетарному программному обеспечению, что позволяет свободно использовать его в образовательных и научных целях.

Оформлены документы на получение университетом исключительного права на результат интеллектуальной деятельности (РИД).

В приложении реализованы все поставленные задачи, что является отличным результатом.

Список информационных источников

1. Android – Текст: электронный // studwood.net. URL: <https://studwood.net/2437096/tovarovedenie/android> (дата обращения: 04.06.2022)
2. Android – Текст: электронный // Википедия свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/Android> (дата обращения: 04.06.2022).
3. Android 10 – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_10 (дата обращения: 04.06.2022).
4. Android 11 – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_11 (дата обращения: 04.06.2022).
5. Android 12 – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_12 (дата обращения: 04.06.2022).
6. Android KitKat – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_KitKat (дата обращения: 04.06.2022).
7. Android Lollipop – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_Lollipop (дата обращения: 04.06.2022).
8. Android Marshmallow – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_Marshmallow (дата обращения: 04.06.2022).
9. Android Nougat – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_Nougat (дата обращения: 04.06.2022).
10. Android Oreo – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_Oreo (дата обращения: 04.06.2022).
11. Android Pie – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Android_Pie (дата обращения: 04.06.2022).

12. Vox2D – Текст: электронный // Википедия свободная энциклопедия.
URL: <https://ru.wikipedia.org/wiki/Vox2D> (дата обращения: 04.06.2022)
13. FreeType – Текст: электронный // Википедия свободная энциклопедия.
URL: <https://ru.wikipedia.org/wiki/FreeType> (дата обращения: 04.06.2022).
14. gdxAi – Текст: электронный // GitHub. URL: <https://github.com/libgdx/gdx-ai/wiki> (дата обращения: 04.06.2022).
15. HyperLap2D – Текст: электронный // Rednblackgames.GitHub. URL <https://rednblackgames.github.io/HyperLap2D-Wiki/> (дата обращения: 04.06.2022).
16. iOS – Текст: электронный // Википедия свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/IOS> (дата обращения: 04.06.2022)
17. LibGDX - фреймворк для разработки игр – Текст: электронный // LibGDX
URL: <http://www.libgdx.ru/2013/08/introduction.html> (дата обращения: 29.05.2022).
18. OpenAL – Текст: электронный // Википедия свободная энциклопедия.
URL: <https://ru.wikipedia.org/wiki/OpenAL> (дата обращения: 04.06.2022).
19. OpenGL – Текст: электронный // Википедия свободная энциклопедия.
URL: <https://ru.wikipedia.org/wiki/OpenGL> (дата обращения: 04.06.2022).
20. Packr – Текст: электронный // GitHub. URL: <https://github.com/libgdx/packr> (дата обращения: 04.06.2022).
21. Unity – что это? – Текст: электронный // Мастер игр. URL: <http://masterigr.ru/unity-что-это/> (дата обращения: 04.06.2022).
22. Unity (игровой движок) – Текст: электронный // Википедия свободная энциклопедия. URL: [https://ru.wikipedia.org/wiki/Unity_\(игровой_движок\)](https://ru.wikipedia.org/wiki/Unity_(игровой_движок)) (дата обращения: 04.06.2022).
23. Unreal Engine – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Unreal_Engine (дата обращения: 04.06.2022).

24. Unreal Engine 4 – Текст: электронный // StackOverflow. URL: <https://ru.stackoverflow.com/tags/unreal-engine-4/info> (дата обращения: 04.06.2022)
25. Гинсбург Д., Пурномо Б. OpenGL ES 3.0. Руководство разработчика / пер. с англ. А. Борескова. – М.: ДМК Пресс, 2015. – 448 с.: ил. ISBN 978-5-97060-256-0 – Текст: непосредственный
26. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы: межгосударственный стандарт: утв. и введ. в действие Постановлением Государственного комитета СССР по стандартам от 24 марта 1989 г. No 661: дата введ. 1990-01-01: переиздание июнь 2009 г. / разработан и внесен Государственным комитетом СССР по стандартам, Министерством приборостроения, средств автоматизации и систем управления СССР. – М.: Стандартинформ, 2009. – Текст : электронный // Электронный фонд правовых и нормативно-технических документов [сайт]. – URL: <https://docs.cntd.ru/document/1200006924> (дата обращения 17.01.2022).
27. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на автоматизированные системы. Техническое задание на создание автоматизированной системы = Information technology. Set of standards for automated systems. Technical directions for automated system making : межгосударственный стандарт : издание официальное : утвержден : Постановлением Государственного комитета СССР по стандартам от 24.03.89 N 661 : взамен ГОСТ 24.201-85 : дата введения 1990-01-01 / разработан и внесен Государственным комитетом СССР по стандартам, Министерством приборостроения, средств автоматизации и систем управления СССР. – Москва : Стандартинформ, 2009 м. – Текст : непосредственный
28. ГОСТ Р 7.0.100 – 2018. Библиографическая запись. Библиографическое описание: общие требования и правила составления : нац. стандарт Рос.

Федерации : изд. офиц. : утв. и введ. в действие Приказом Федер. агентства по техн. регулированию и метрологии от 3 дек. 2018 г. No 1050-ст : введ. впервые : дата введ. 2019-07-01 / разработ. Федер. гос. унитар. предприятием «Информ. телеграф. агентство России (ИТАР-ТАСС)» фил. «Рос. кн. палата», Федер. гос. бюджет. учреждением «Рос. гос. б-ка», Фе-дер. гос. бюджет. учреждением «Рос. нац. б-ка». – Москва : Стандартинформ, 2018. – 128 с. – (Система стандартов по информации, библиотечному и издательскому делу). – Текст : непосредственный.

29. ГОСТ Р 7.0.100 – 2018: новые правила описания статей и книг. Примеры библиографического описания для списка литературы. Памятка. – Текст: электронный // URL: <http://spf-vshni.ru/upload/medialibrary/9df/ПАМЯТКА-i-PRIMERY-bibliograficheskogo-opisaniya-dlya-spiska-literatury.pdf> (дата обращения 21.01.2022).

30. История версий Android – Текст: электронный // Википедия свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/История_версий_Android (дата обращения: 04.06.2022).

31. Подробная история популярной ОС Google Android – Текст: электронный // ferra.ru. URL: <https://www.ferra.ru/review/apps/android-history.htm> (дата обращения: 04.06.2022)

32. Создание игр на языке C#. Обзор Unity – Текст: электронный // Дзен. URL: <https://zen.yandex.ru/media/id/5d33234f31878200acc7007f/sozdanie-igr-na-iazyke-c-obzor-unity-5ddf896f505a381a030263a3> (дата обращения: 04.06.2022).