Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Базовая кафедра геоинформационных систем

УТВЕРЖДАЮ
Заведующий кафедрой
_____В.И. Харук
подпись
«____» _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 - Информационные системы и технологии

Разработка игр для мобильных устройств на основе движка Unity и ГИС-технологий

Руководитель _____ доцент, канд. тенх. наук А. С. Савельев
подпись, дата

Выпускник _____ М. А. Сеславин
подпись, дата

Красноярск 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Передвижение и ориентирование на местности, будь то туризм или повседневная жизнь, всегда было важным качеством человека, которое особенно проявлялось при попадании на незнакомую территорию. Автоматизация коснулась и этого процесса. С этой целью активно внедряют геоинформационные системы (ГИС), мобильные и десктопные приложения и иные технологии.

Наряду с туризмом развивалась и игровая индустрия, в последнее время она стала самостоятельным бизнесом с большим оборотом денежных средств. Геоинформационные технологии (ГИТ) не обошли и эту отрасль, активно разрабатываются и используются игры с реальными картами.

Приложения такого типа обычно содержат в себе навигацию и функцию определения местоположения. Вместе с навигацией используют карты, это могут быть Google Maps, Яндекс. Карты, OpenStreetMap (OSM) и др.

Востребованность таких систем и приложений на рынке подтверждает актуальность их разработки и внедрения.

Особенно внимательными при передвижении следует быть водителям транспорта, во избежание пропущенных поворотов и потраченного времени. С этой целью требуется детально изучать улицы и маршруты города. Игра, разрабатываемая в данной работе, ориентирована на просмотр улиц города и на отработку скорости реакции при поворотах. С помощью неё пользователь сможет запомнить проблемные участки и в реальном мире сориентироваться на местности правильным образом.

В рамках данной работы будет описано создание мобильной игры на базе ГИС-технологий с использованием движка Unity для операционной системы (ОС) Android. Данная ОС является самой распространенной и установленной на большинство выпускаемых телефонов и планшетов. Отличительной особенностью такой мобильной системы является высокая степень доступности средств разработки и открытое программное обеспечение (ПО).

Выбрана мобильная разработка, основанием такого выбора послужило то, что именно такие устройства используются в повседневной жизни и находятся всегда под рукой.

Цель работы заключается в изучении возможностей современных движков для разработки игр и ГИС-технологий для разработки мобильных приложений, содержащих реальную карту города и создание игры для платформы Android.

Объектом исследования в работе являются игры с использованием реальной карты местности. Предметом исследования являются методики и алгоритмы создания мобильной игры. Для достижения поставленной цели необходимо решить следующие задачи.

1) Изучение инструментов для разработки игр.
2) Проектирование и разработка мобильной игры.
3) Описание руководства пользователя.

Структура работы обусловлена предметом, целью и задачами исследования. Работа состоит из введения, трёх глав и заключения.

Во введении рассматривается актуальность работы, раскрывается объект и предмет исследования, выявляются цель и задачи работы.

Первая глава рассматривает предметную область, обоснование мобильной разработки под операционную систему Android, обзор инструментов для создания игры и обзор аналогов.

Во второй главе происходит проектирование и разработка мобильной игры, построение различных диаграмм, показывающих функциональные возможности системы, алгоритм работы игры, также представлено описание способа хранения маршрутов в игре.

В третьей главе представлены результаты разработки игры, которые показываются по средствам описания руководства пользователя с описаниями всех функций.

1 Общие сведения о разработке, предметной области и программном обеспечении

Игровая индустрия в настоящее время является одной из самых быстроразвивающихся и популярных отраслей в сфере информационных технологий и развлечений, особенно это касается мобильных игр. Данная сфера имеет большой бюджетный оборот и превратилась в последнее время в сферу бизнеса. Разработкой занимаются как крупные компании, такие как Capcom, Ubisoft, Activision Blizzard, так и небольшие. Большая конкуренция и множество разработанных мобильных игр не исключают возможность создания собственного игрового приложения, так как в настоящее время реализованы не все идеи и не все концепции, тем более игровая индустрия является полем для творчества.

1.1 Выбор платформы для разработки игры для мобильных устройств

Системы и приложения, предназначенные для ориентирования на местности, разрабатываются под компьютерные и мобильные операционные системы. Наиболее востребованными являются приложения для мобильных ОС, к которым относятся Android, iOS, Windows Phone. Востребованность объясняется тем, что именно такие устройства находятся с собой у пользователей в поездке или в дороге. Они предоставляют быстрый доступ к картам, приложениям, веб-ресурсам и др., являются средством коммуникации и используются пользователями повседневно.

По данным Star Counter [1] за 2018 год самой популярной мобильной операционной системой в России и самой популярной ОС в мире является операционная система Android. Результаты данного анализа в виде рейтинга представлены на рисунке 1. По данным мирового рейтинга такая мобильная операционная система используется у 41,24 % населения. В России данную ОС используют 15,88 % населения, на первом месте находится Windows.

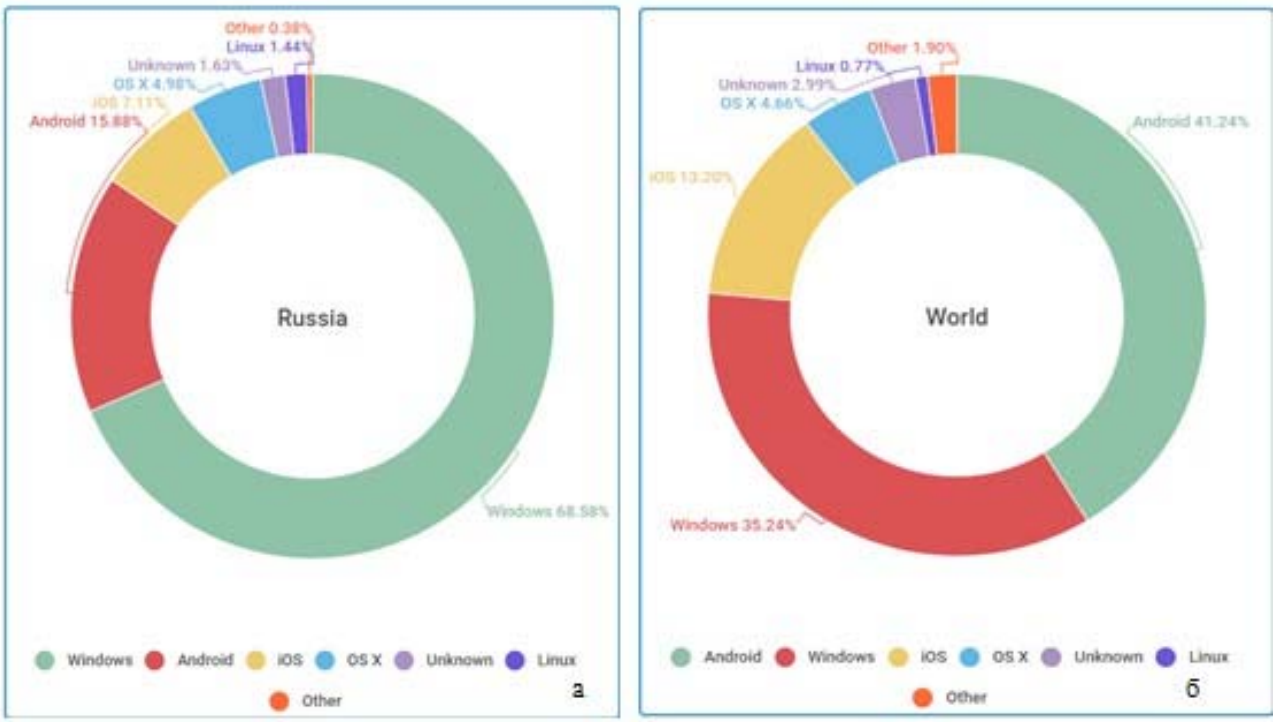

Рисунок 1 – Рейтинг операционных систем: а – операционные системы в России (2018 г.); б – операционные системы в мире (2018 г.)

Операционная система Android построена на ядре Linux и работает с собственной виртуальной машиной Java от Google. Данная ОС появилась в 2008 году [2]. Является операционной системой для телефонов, планшетов, часов, очков, нетбуков, телевизоров и других устройств. ОС позволяет создавать Java - приложения, которые управляются устройством через разработанные Google библиотеки.

Приложениями под операционную систему Android являются программы с байт-кодом для виртуальной машины Dalvik и форматом установочных пакетов .APK. Данные пакеты представляют собой архив с исполняемыми файлами для Android. Каждое приложение для данной ОС компилируется и упаковывается в такой файл, включающий в себя весь код приложения, ресурсы активы, библиотеки, файл манифеста. Файлы такого формата не шифруются, они являются представителем формата архива ZIP. У файлов такого формата есть своя структура, они включают meta-inf, lib (библиотеки для типов процессов, res(ресурсы), assets(ресурсы, используемые AssetManager, AndroidManifest.xml

(файл с описанием приложения), lasses.dex (исполняемый код в формате DEX), resources.arsc (компилированные ресурсы).

У такой операционной системы есть свои преимущества, среди которых открытость исходного кода, установка программ, не требующая подключения к интернету, доступность разных аппаратных платформ, наличие собственного магазина и др.

Подводя итог всему выше сказанному, можно сделать вывод о том, что мобильные устройства используются всеми пользователями в повседневной жизни, отсюда вывод о том, что мобильная разработка является правильным и логичным решением в данной ситуации.

1.2 Обзор существующих игр, симулирующих реальность

В настоящее время стала популярной разработка и использование игр с реальным игровым миром. Такие приложения предоставляют доступ к альтернативной реальности. В качестве платформы используется реальный мир: карты стран, городов, реальность дополняется игровыми элементами. В открытом мире пользователь сам решает, как ему передвигаться. Такие игры имеют ряд преимуществ. Во-первых, они увлекают пользователя, делая процесс игры более интересным и захватывающим. Во всем остальном такие миры воспринимаются более целостно, пользователь ощущает больше свободы.

1.2.1 Generation Streets

Игра позволяет загрузить карту любого города. Карта представляется в 3 D режиме, за основу берутся карты OpenStreetMap. Игровой процесс заключается в том, что пользователь загружает карту местности, расставляет дома, деревья, добавляет текстуру, уличное освещение, опоры линий электропередач и иные визуальные элементы. Данная игра представляет аркадный шутер с видом от третьего лица [3]. Пример того, что получается в итоге, представлен на рисунке 2.

Рисунок 2 – Игра Generation Streets

Загрузки подлежат фактически любые карты мира, исключением является Антарктида и Северный полюс.

Игра относится к типу экшена. Это реализовано с целью приукрашивания процесса исследования местности.

1.2.2 Ingress

Геолокационная онлайн-игра с дополненной реальностью, разработанная Google для мобильных устройств. Выпущена в 2012 году. Работает с реальными картами.

Суть игры заключается в следующем: пользователь, используя своё мобильное устройство, осуществляет передвижение в реальном мире, при этом необходимо осуществить захват порталов. Порталы представляют собой достопримечательности реального мира. На карте изображены очертания дорог и домов, порталы, материи, поля, артефакты и связи. Каждому элементу присвоен свой цвет [4]. За действия в игре игроки получают очки. Окно игры представлено на рисунке 3.

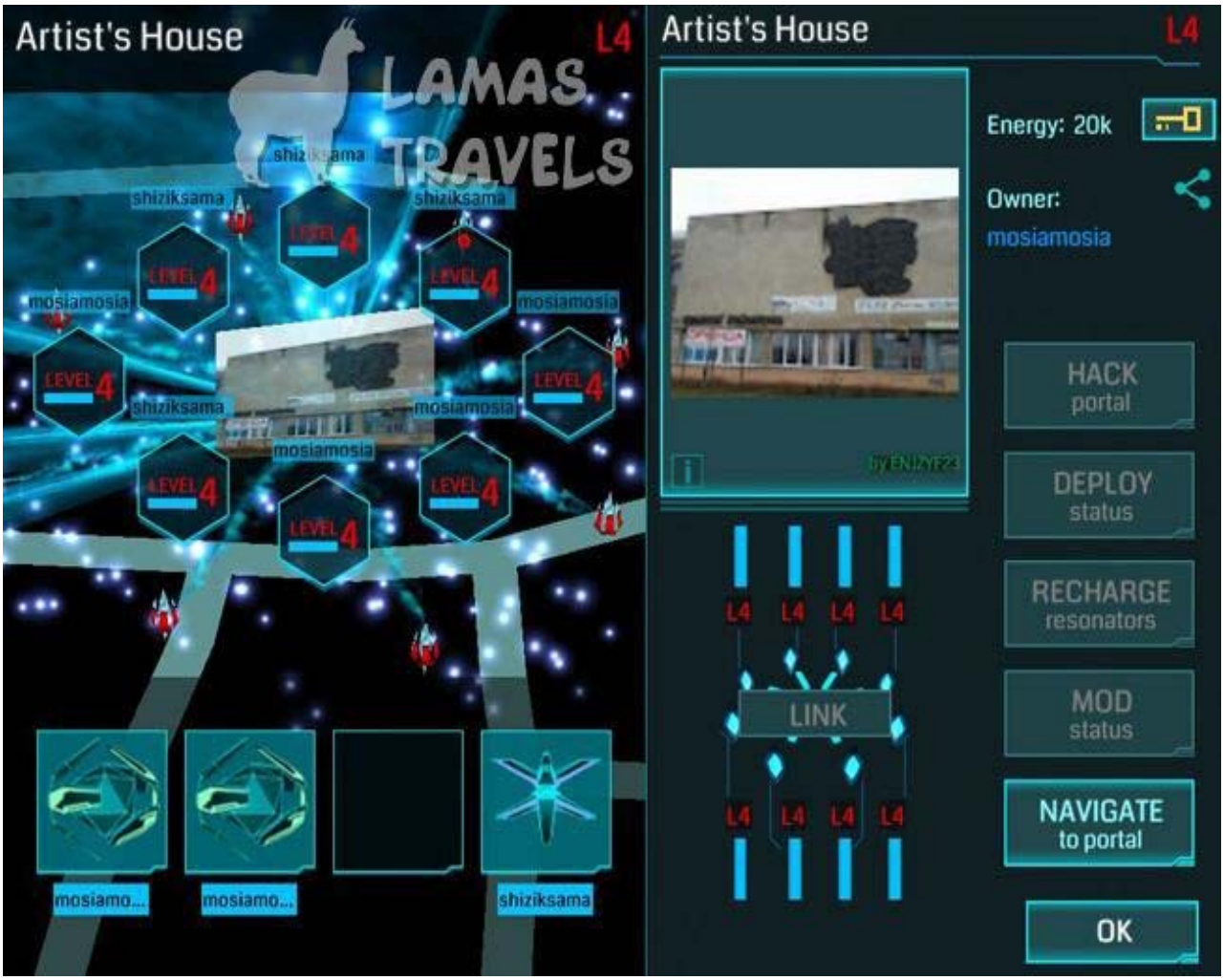

Рисунок 3 – Игра Ingress

1.2.3 Resources

Игра на основе реального мира, которая имеет жесткую привязку к координатам. Для прохождения необходимо перемещаться по реальному миру. Является экономическим симулятором. Суть игры заключается в следующем: пользователю необходимо выйти на улицу, определить своё местоположение и выполнить построение своей первой шахты. Шахты строятся для извлечения ресурсов, которые в последующем используются для переработки продуктов. Всё это выполняет с целью повышения собственного капитала и расширения строений и своих владений. Для поиска ресурсов в игре необходима геолокация а для сбора и переработки ресурсов подключение к интернету. Все эти действия необходимы для того, чтобы заработать имя в игровом мире и попасть в заголовки газет [5]. Окно работы с игрой продемонстрировано на рисунке 4.

Рисунок 4 – Игра Resources

1.2.4 Результаты

Все рассмотренные игры основаны на реальных картах. Первая игра является компьютерной разработкой, две другие созданы под мобильные ОС. Игры носят развлекательный характер, они не связаны напрямую с обучением пользователя ориентированию на местности. Игра, разрабатываемая в рамках данной работы, напротив, в первую очередь занимается обучением пользователя. Игра разрабатывается под мобильную операционную систему Android. За основу берется карта местности города Красноярска.

1.3 Инструменты для разработки игры

Данный раздел рассматривает в себе инструменты для создания мобильной игры с использованием реальной карты. В разделе рассмотрены цифровые карты от Google, Яндекс и OpenStreetMap (OSM), платформы для стилевого оформления карт QGIS и Mapbox, среда разработки Unity и язык программирования C#.

1.3.1 Цифровые карты

Навигация по городу и местоположение отображаются на карте. Для этого в качестве подложки используют известные картографические инструменты. Выбор таких карт обширный, это могут быть Яндекс Карты, Google Maps, OpenStreetMap и другие. Рассмотрим упомянутые источники карт.

OpenStreetMap (OSM) является картографическим проектом в некоммерческой сфере, доступ к которому осуществляется через веб. Открытый проект, который создан сообществом участников и запущен в 2004 году [6]. Проект создан общими силами сообщества участников. Карта всего мира. Внешний вид карты представлен на рисунке 5.

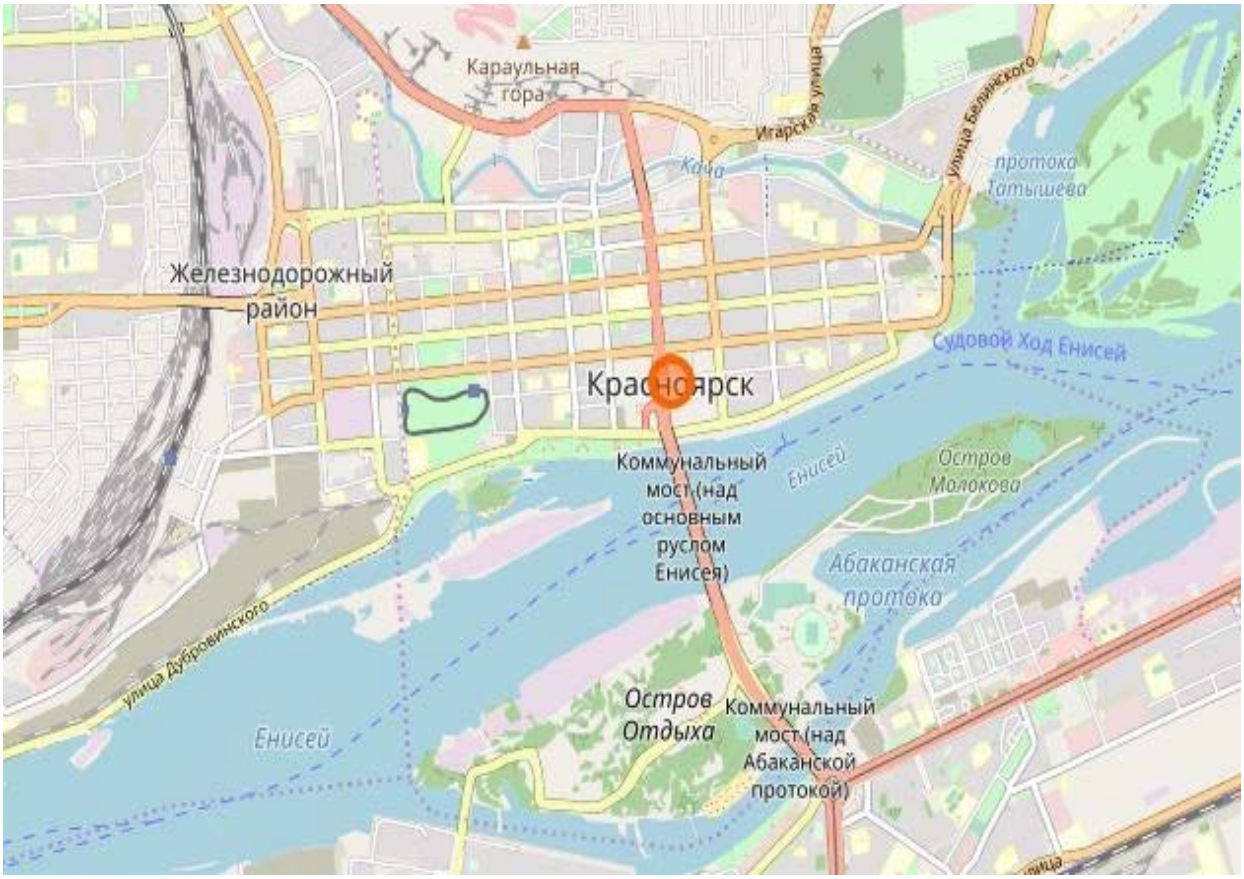

Рисунок 5 – Красноярск на карте OSM

Яндекс. Карты – разработка российского рынка и поисково-информационной картографической службой Яндекса, открыта в 2004 году [7]. Среди функциональных возможностей карты: поиск по местности, построение

маршрутов, панорам улиц, просмотр информации о пробках и местах. Обновление карт происходит раз в две недели.

Google Maps - платный картографический источник, разработанный Google в 2005 году [8]. Представляет собой спутниковые снимки планеты и карту. Работает как через веб-ресурс, так и через мобильное устройство, через приложение.

Для того чтобы использовать готовые карты для игровых приложений необходимо воспользоваться Application Programming Interface (API). API представляет собой набор классов, процедур, функций, структур и констант, которые являются способами взаимодействия одной программы с другой. Он определяет функциональность программы, абстрагируясь от того, как эта функциональность реализована. На базе представленных карт уже имеются созданные игры. Яндекс. Карты использует игра «Владение», Google Карты используются в Jurassic World Alive, OSM в Master City. Яндекс и OSM представляют доступ к API бесплатно. Яндекс накладывает некоторые ограничения для бесплатного использования, проект с использованием их карт должен являться некоммерческим и быть с открытым доступом. Для того, чтобы разработать игру с использованием Google API необходимо заключить контракт с компанией.

1.3.2 Стилевое оформление карт

Для изменения внешнего вида карты существует несколько сторонних систем, рассмотрим две из них: Quantum GIS (QGIS) и Mapbox Studio.

QGIS является свободной и кроссплатформенной геоинформационной системой, которая состоит из двух частей: серверной части и настольного приложения.

GIS работает в Windows и в большинстве платформ Unix (включая Mac OS), поддерживает множество векторных и растровых форматов и баз данных, а также имеет богатый набор встроенных инструментов.

Возможности системы обширны, среди них просмотр данных, исследование данных и компоновка карт, управление данными, анализ данных, публикация карт в сети Интернет, расширение функциональности при помощи сторонних модулей.

Окно работы с системой и с открытой картой Красноярска представлено на рисунке 6.

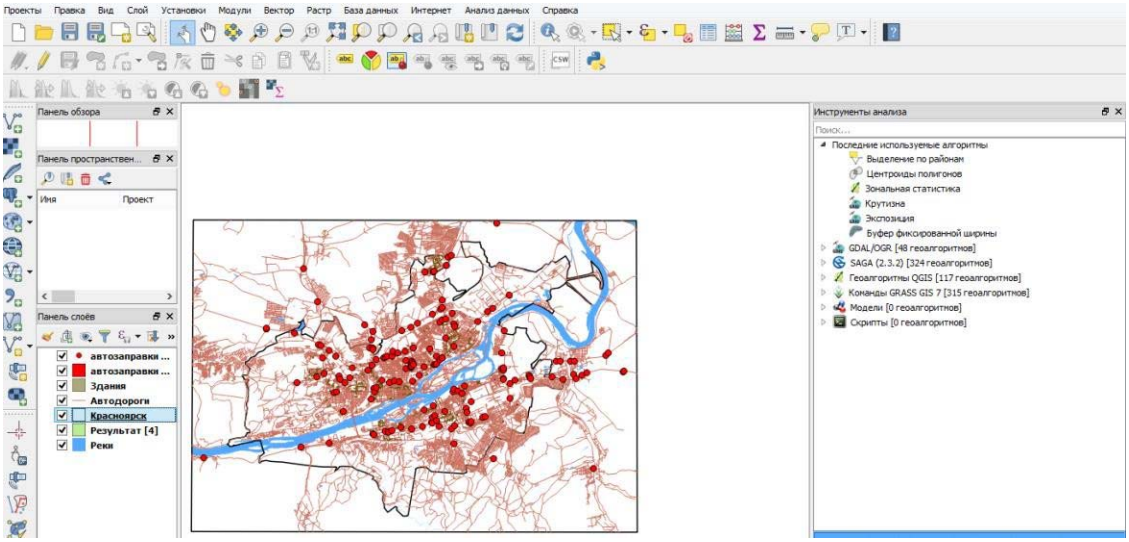

Рисунок 6 – Окно работы с QGIS

Mapbox Studio является системой для создания собственных карт. Осуществляет контроль над дизайном, позволяя создавать красивые пользовательские карты в соответствии с потребностями. Подходит для визуализации данных и создания карт с нуля [9].

Интерфейс работы с программой представлен на рисунке 7.

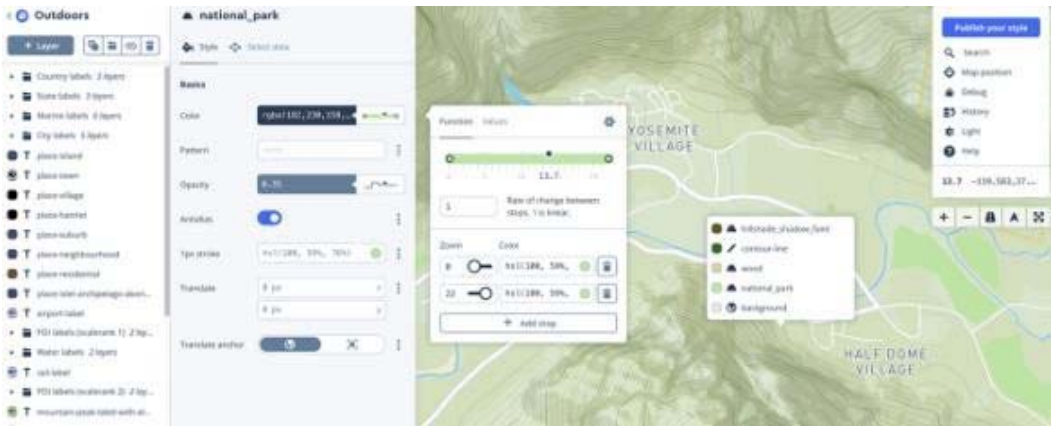

Рисунок 7 – Интерфейс Mapbox Studio

Доступна в веб версии и позволяет кастомизировать следующие вещи: названия географических объектов (изменение шрифта, цвета, прозрачности), цвета дорог, зданий, парков и зеленых насаждений, туннелей, мостов и стран. Возможностей очень много, все меняется в режиме онлайн и сразу виден результат изменений.

В данной работе используется Mapbox, в том числе Mapbox Unity SDK для импорта карт в среду Unity. Mapbox Unity SDK используется для игр, настраивает карты для использования их в дальнейшем в игровом мире [10]. Занимается разблокировкой глобальных данных для создания пользовательских миров, поиска местоположения. Содержит в себе готовые стили, которые можно использовать в проектах. К особенностям SDK относятся:

1. Глобальные данные. Глобальная векторная карта мира, которая включает в себя дороги, дорожные знаки, достопримечательности, здания, гидрографию и иные объекты реального мира.

2. Глобальные данные. Использование спутниковых снимков, глобальная цифровая модель рельефа в растровом формате.

3. Возможность загрузки пользовательских данных.

1.3.3 Среда разработки

Unity является межплатформенной средой разработки игр. Позволяет создавать приложения, которые могут осуществлять работу более чем под двадцатью операционными системами, в том числе под мобильными. Преимуществом данной среды является наличие визуальной среды разработки и модульной системы компонентов.

Редактор среды имеет простой Drag and Drop интерфейс, состоящий из различных окон. Поддерживает два языка разработки: JavaScript, C#.

При разработке проект в Unity делится на сцены, которые представляют собой отдельные файлы, содержащие свои игровые миры с собственными наборами объектов, сценариев и настроек. Сцены могут содержать собственные объекты и пустые объекты. Вместе с этим объекты содержат наборы

компонентов, с которыми взаимодействуют скрипты, также объекты имеют название, тег и слой отображения. Объект на сцене характеризуется координатами местоположения, поворота и размерами изображения по трем осям, для работы с этим используется компонент transform.

Окно работы с Unity представлено на рисунке 8.

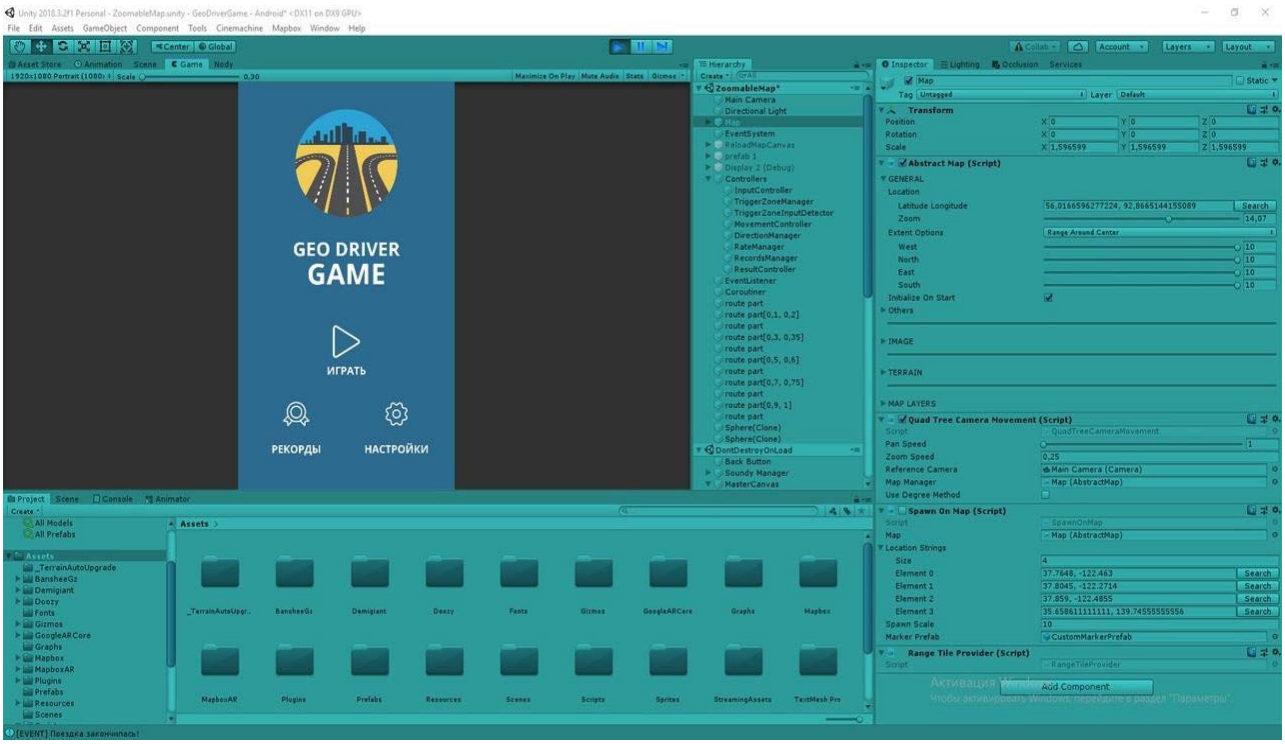

Рисунок 8 – Окно работы Unity

В редакторе имеется система наследования, это когда дочерние объекты повторяют все изменения позиции, поворота, масштабирования родительского объекта. Скрипты прикрепляются к объектам в виде отдельных компонентов.

Движок поддерживает множество популярных форматов. Модели, звуки, текстуры, материалы, скрипты можно запаковывать и передавать другим разработчикам или же размещать на ресурсах для свободного доступа или в специальном интернет-магазине Unity Asset Store, для использования которого необходимо наличие профиля разработчика.

В состав Unity включен Unity Asset Server, который является инструментарием для совместной разработки на базе Unity, являющийся дополнением, добавляющим контроль версий и ряд других серверных решений.

1.3.4 Языки программирования

Так как среда разработки Unity работает с языком программирования C#, то принято решение рассмотреть и использовать именно его. CSharp (C#) является объектно-ориентированным языком программирования, относящимся к семейству языков с с-подобным синтаксисом. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Не поддерживает множественное наследование классов.

Разрабатывался как язык прикладного уровня, стандартизован в ECMA (ECMA-334) и ISO (ISO/IEC 23270).

У данного языка есть несколько реализаций, среди которых реализация C# в виде компилятора csc.exe, присутствие в составе проекта Rotor, в проекте Mono, DotGNU и в DotNetAnywhere.

Язык поддерживает инкапсуляцию, наследование и полиморфизм. Все переменные и методы, вместе с методом Main, представляющий собой точку входа в приложение, инкапсулируются в определения классов. Класс наследуется из одного родительского класса, но может реализовывать любое число интерфейсов. Методы, которые переопределяют виртуальные методы родительского класса, должны содержать ключевое слово override, чтобы исключить случайное переопределение. В языке C# структура похожа на облегченный класс: это тип, распределяемый в стеке, реализующий интерфейсы, но не поддерживающий наследование[13].

Вместе с основными принципами объектно-ориентированного программирования, C# имеет ряд новых языковых конструкций, которые упрощают разработку программных компонентов. Они представлены ниже:

1. Инкапсулированные сигнатуры методов, именуемые делегатами, которые позволяют реализовать типобезопасные уведомления о событиях.

2. Свойства, выполняющие функцию акцессоров для закрытых переменных-членов.

3. Атрибуты, предоставляющие декларативные метаданные о типах во время выполнения.

4. Внутристрочные комментарии для XML-документации.

5. LINQ для создания запросов к различным источникам данных.

Единицей компиляции в данном языке является файл, который может содержать одно или несколько описаний типов. Типы подразделяются на ссылочные, к ним относятся классы, интерфейсы, массивы, делегаты, и типы-значения, к ним относятся перечисления, структуры и элементарные типы.

Данный язык поддерживает автоматическую сборку мусора и имеет обширный набор операций с четырнадцатью уровнями приоритета. Видимость объектов программы регулирует пространство имен, которые могут быть вложенными.

Для компиляции программы написанной на данном языке можно воспользоваться компилятором csc, который входит в состав Microsoft .NET Framework SDK — комплект разработчика для среды Microsoft .NET, запускаемый при помощи командной строки.

В C# сохранены и даже приведены в порядок некоторые традиционные конструкции, такие как перечисления, структуры, многомерные массивы. Механизм передачи параметров в C# хорошо продуман, предусматривая передачу, как по значению, так и по ссылке, отсутствует возможность передачи по ссылке без указателя.

Выбор данного языка объясняется тем, что его используют подавляющее число программистов, он подробно изучается в университетах и имеет библиотеки, которые свободно можно импортировать в собственный проект, в том числе среда Mapbox написана на C#. Выбранный игровой движок Unity поддерживает этот язык программирования для создания мобильной игры для платформы Android.

2 Проектирование и разработка игры

2.1 Диаграмма функционального моделирования

Для проектирования информационных систем и приложений применяют различные методологии, одной из наиболее часто встречающихся являются Structured analysis and designtechnique (SADT), которая является методологией структурного анализа и проектирования. На рисунке 12 изображена контекстная диаграмма процесса, она является диаграммой верхнего уровня и показывает назначение системы и взаимодействие с внешней средой. Вместе с назначением системы показывается и взаимодействие с внешней средой, взаимодействие показывается с использованием четырех типов стрелок:

1. Входные данные (маршруты, имя) – показывают информацию, которая необходима для функционирования системы (изображаются слева от блока функции).

2. Выходные данные (таблица рекордов) – показывают результат работы системы (изображаются справа от блока функции).

3. Управление (выбранный маршрут, алгоритмы) – показывают ограничения, вводимые в систему (изображаются сверху от блока функции).

4. Механизмы (мобильное устройство, пользователь, модули игры) – показывают то, по средствам чего функционирует системы (изображаются снизу от блока функции).

Для функционирования игры необходимо в базу данных добавить маршруты, которые содержат пункт отправления и пункт назначения, а также пользователю необходимо вбить имя пользователя с клавиатуры после того, как будут подсчитаны результаты прохождения маршрута. Игровой процесс ограничивается выбранным маршрутом, именно по нему проходит игра, подсчет итоговых результатов ограничивается алгоритмом расчета, а движение по маршруту ограничивается алгоритмом движения, пользователь может осуществлять движение вперед и выполнять повороты влево и вправо. Результатом работы игры является модифицированная таблица рекордов.

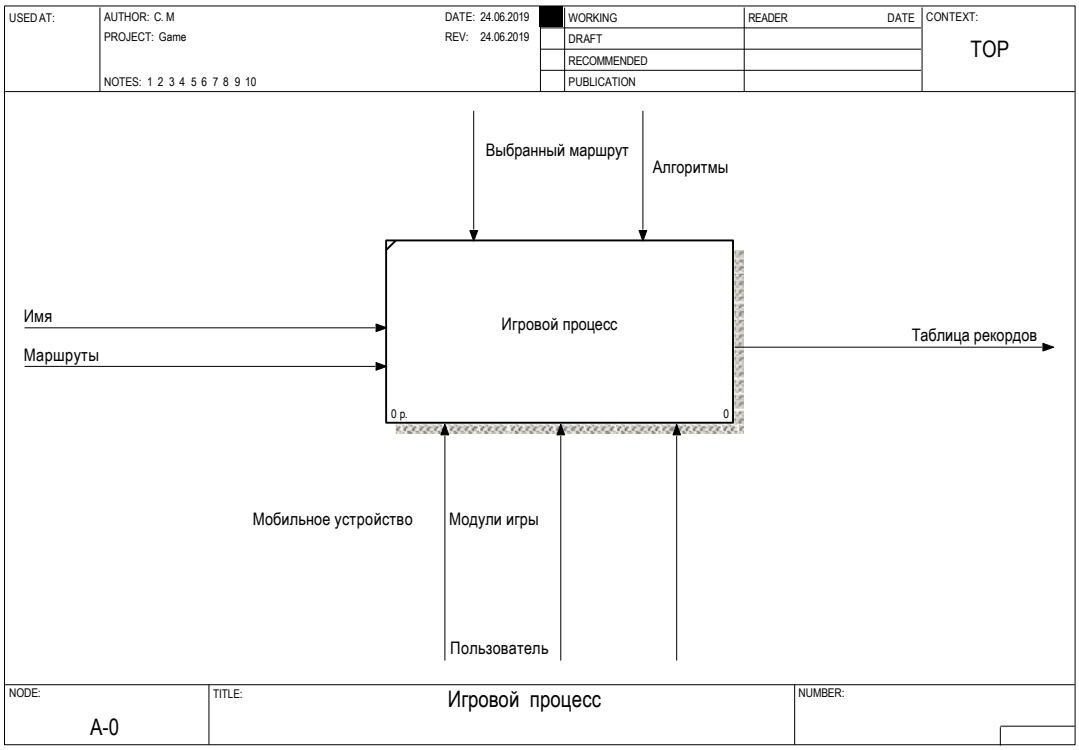

Рисунок 12 – Контекстная диаграмма

После описания основной функции, которая показана в контекстной диаграмме, необходимо выполнить декомпозицию, то есть определить функции, из которых состоит основная, данный процесс показан на рисунке 13.

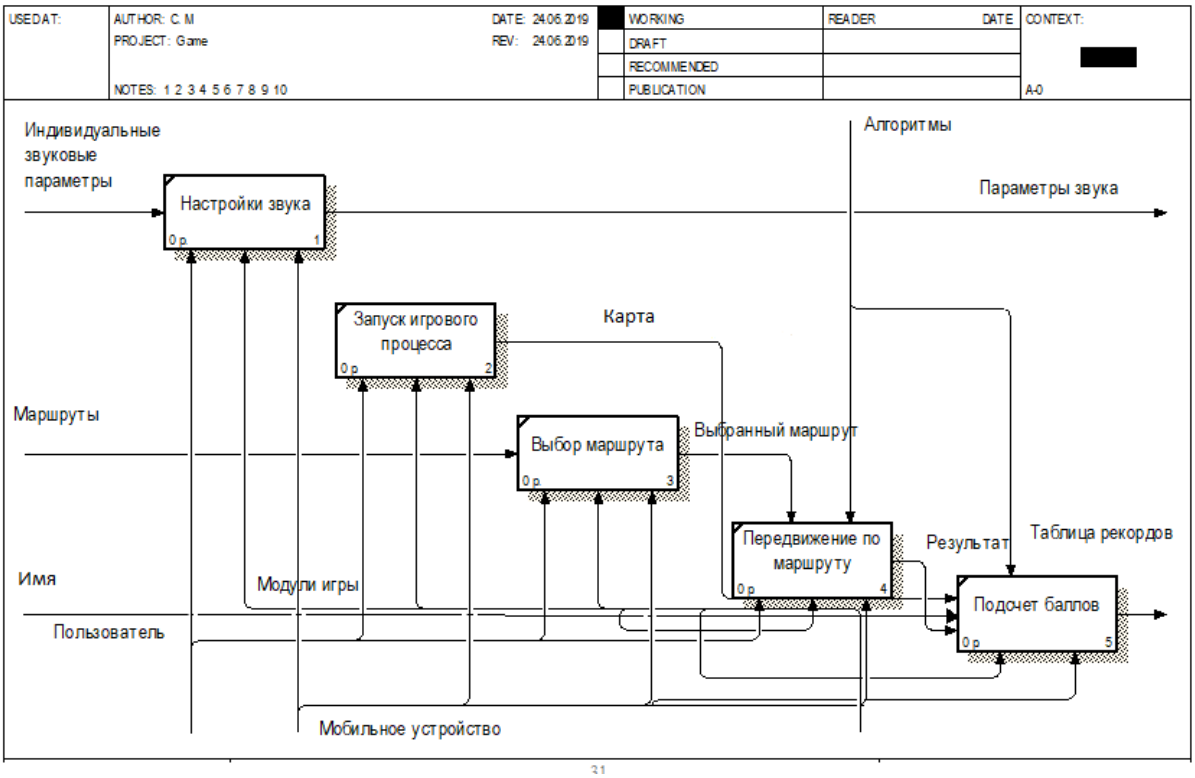

Рисунок 13 – Декомпозиция «Игрового процесса»

Данная диаграмма показывает, что игровой процесс строится с использованием пяти функций. Настройка звука подразумевает выбор индивидуальных настроек пользователем, запуск игрового процесса начинается после запроса пользователя на запуск игры. Выбор маршрута осуществляется из предлагаемых системой, в рамках выбранного маршрута потом осуществляется вся игра. После выбора маршрута проходит процесс передвижения по маршруту, он осуществляется в рамках алгоритма. Самой последней функцией является подсчет баллов, который основывается на результатах игры, после подсчета результата пользователю предлагается ввести своё имя.

2.2 Объектно-ориентированный анализ и проектирование игры

Вторая методология UML (Unified Modeling Language) используется для объектно-ориентированного анализа и проектирования [14]. UML – открытый стандарт, используемый графические обозначения.

2.2.1 Функциональные возможности системы

Одна из диаграмм UML – диаграмма вариантов использования. Диаграмма вариантов использования используется для того, чтобы показать функциональные возможности системы, а также взаимодействие пользователя и системы. Диаграмма для игры представлена на рисунке 14 и 15, они состоят из вариантов использования и акторов. Вариант использования является действием, которое доступно пользователю или которое должна выполнить система при взаимодействии с ним. Актор — это множество ролей в языке моделирования UML, на рисунке 14 в роле актора выступает пользователь, на рисунке 15 – система. Связь «include», изображенная на диаграмме, означает включение и говорит о том, что для выполнения варианта использования нужно выполнить включенные в этот вариант другие варианты использования [15]. Диаграмма, показывающая взаимодействие пользователя с системой показана на рисунке 14.

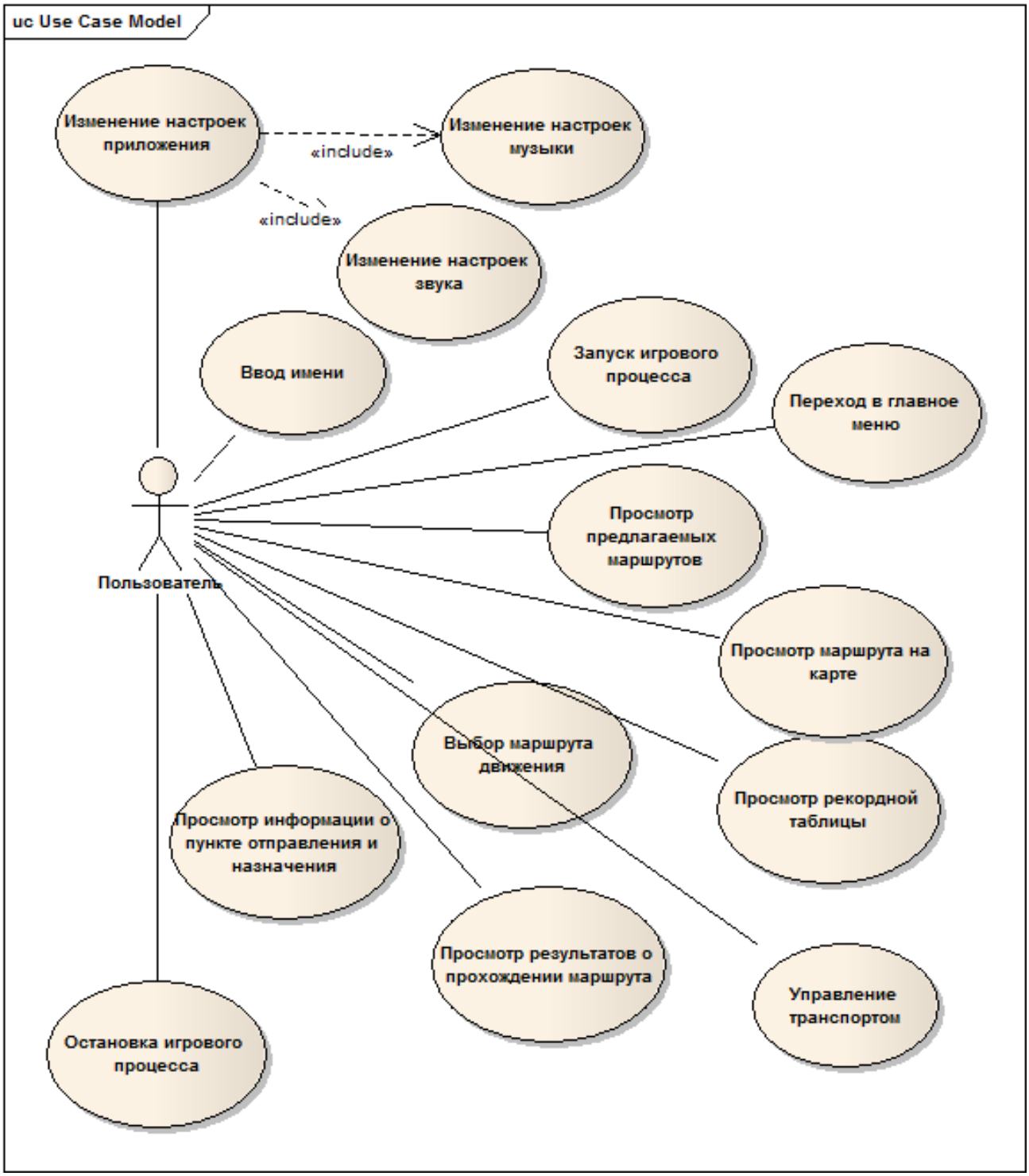

Рисунок 14 – Диаграмма вариантов использования для пользователя

Диаграмма показывает следующие действия пользователя при взаимодействии с системой:

1. Изменение настроек приложения.
2. Просмотр рекордной таблицы.
3. Ввод имени.

4. Запуск игрового процесса.
5. Переход в главное меню.
6. Просмотр предлагаемых приложением маршрутов.
7. Выбор маршрута движения.
8. Просмотр информации о пункте отправления и пункте назначения.
9. Просмотр маршрута на карте.
10. Управление транспортом.
11. Просмотр результатов о прохождении маршрута (в виде процентов).
12. Остановка игрового процесса.

Построим такую же диаграмму для системы, чтобы показать функциональные возможности игры, диаграмма изображена на рисунке 15.

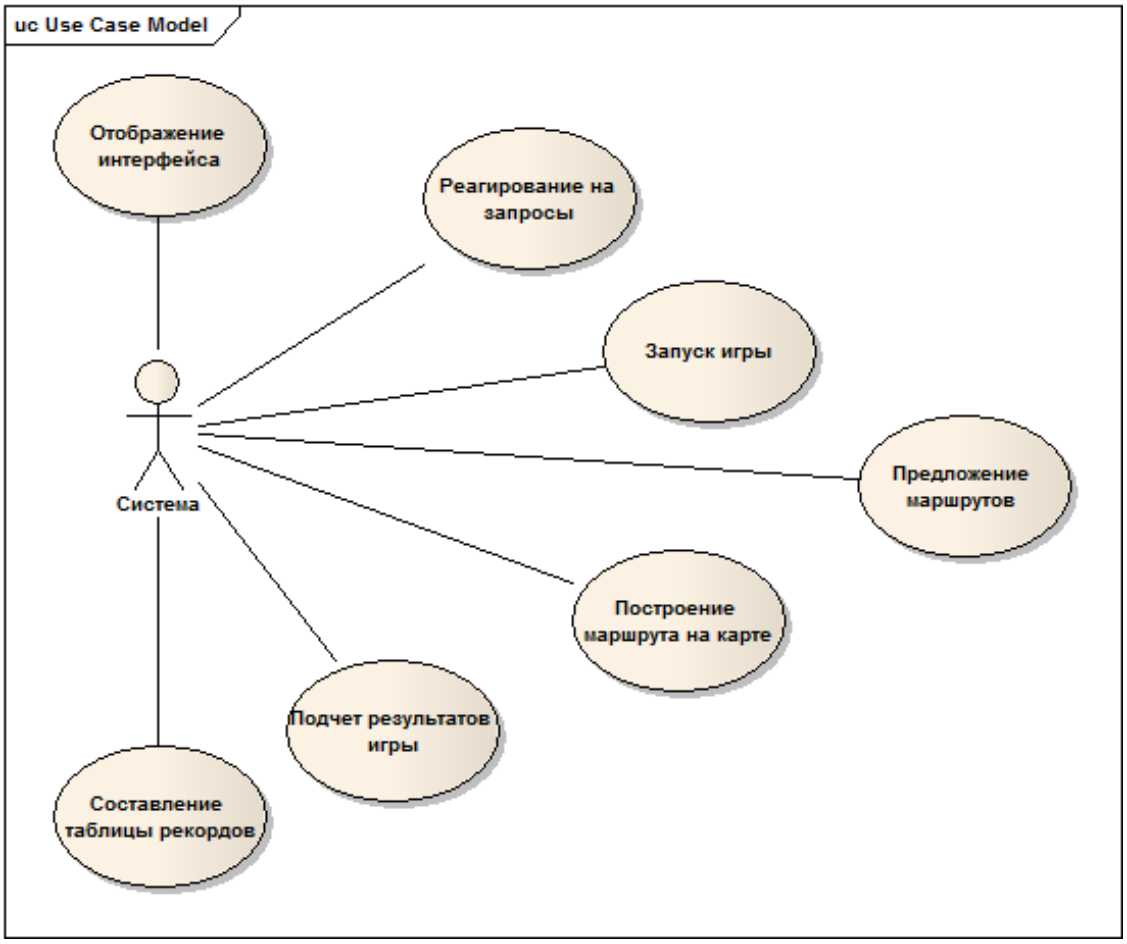

Рисунок 15 – Диаграмма вариантов использования для системы

Диаграмма показывает следующие функциональные возможности системы:

1. Отображение интерфейса.
2. Реагирование на запросы.
3. Запуск игры.
4. Предложение маршрутов.
5. Построение маршрута на карте.
6. Подсчет результатов игры.
7. Составление рекордной таблицы.

2.2.2 Поведение системы

Поведение системы отслеживается по средствам диаграммы деятельности, которая показывает детально алгоритмическую и логическую реализацию операций системы [16]. Вид деятельности принято обозначать в виде прямоугольника скругленными краями, а переходы между ними в виде стрелок. Так же диаграмма имеет начальную и конечную точки. Диаграмма показывает действие пользователя и реакцию системы на произведенное действие, переходы обозначаются стрелками. Начало работы показывается черной точкой с текстом «Начало», завершение работы процесса обозначается светлой точкой с изображением ещё одной черной точкой внутри и текстом «Конец».

Рассмотрим каждый прецедент диаграммы отдельно.

Прецедент 1: Звуковые настройки.

Входными данными являются введенные пользователем индивидуальные настройки звука.

В работе задействованы пользователь и система: система предоставляет интерфейс, пользователь осуществляет выбор параметров звука как для музыки, так и для звука действий. Система отвечает на запросы пользователя, изменяя звуковые настройки на введенные им.

Диаграмма деятельности процесса ввода звуковых настроек предоставлена на рисунке 16.

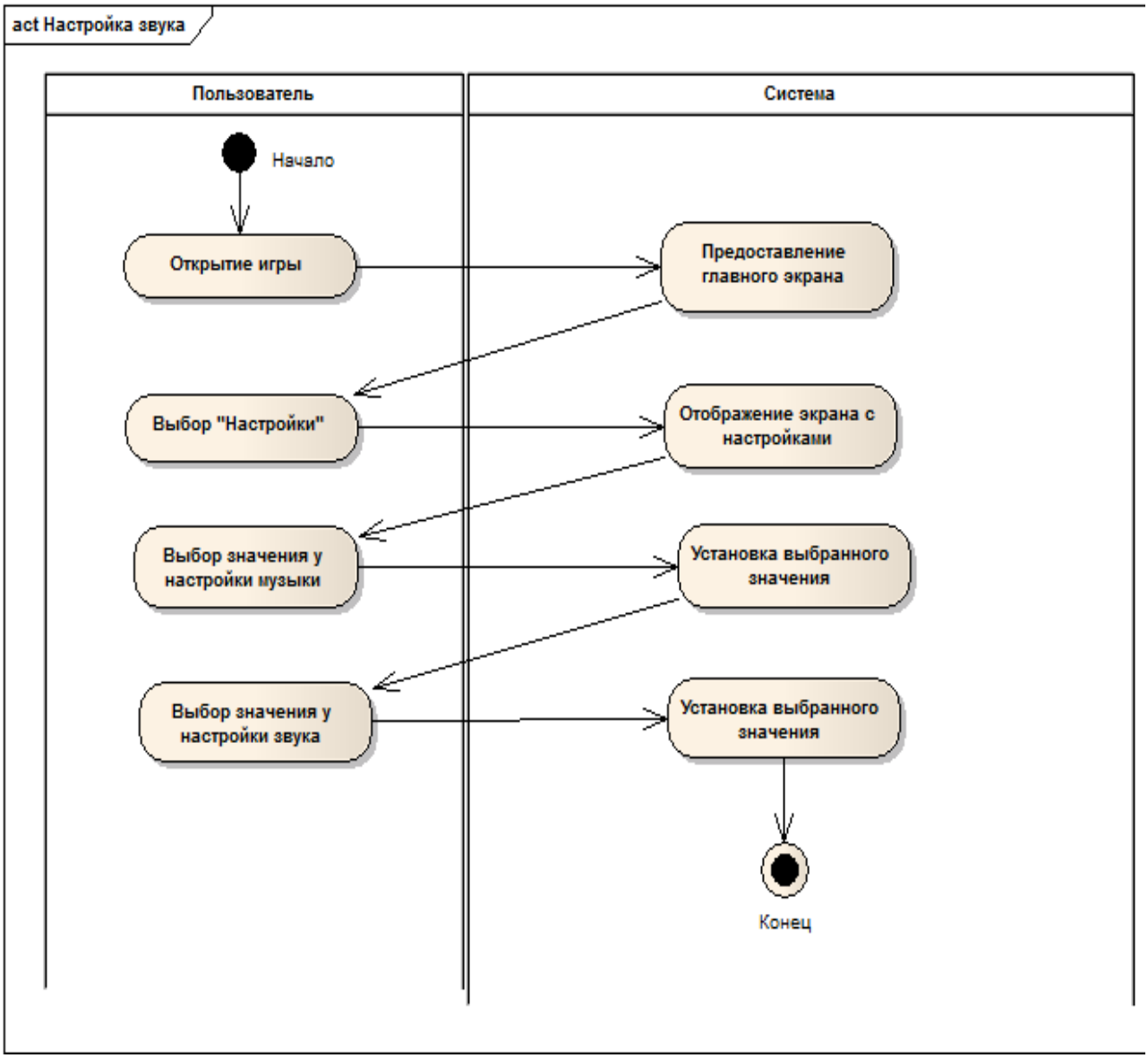

Рисунок 16 – Диаграмма деятельности для процесса «Настройка звука»

Прецедент 2: Запуск игрового процесса.

Входными данными является введенные пользователем имя.

В работе задействованы пользователь и система: пользователь осуществляет ввод имени. Система предоставляет маршруты для выбора. Пользователь, видя точку отправления и назначения принимает, либо отклоняет маршрут. Системы реагирует на выбранные пользователем действия, предоставляя следующий маршрут или запуская игру.

Диаграмма деятельности процесса запуска игры представлена на рисунке 17.

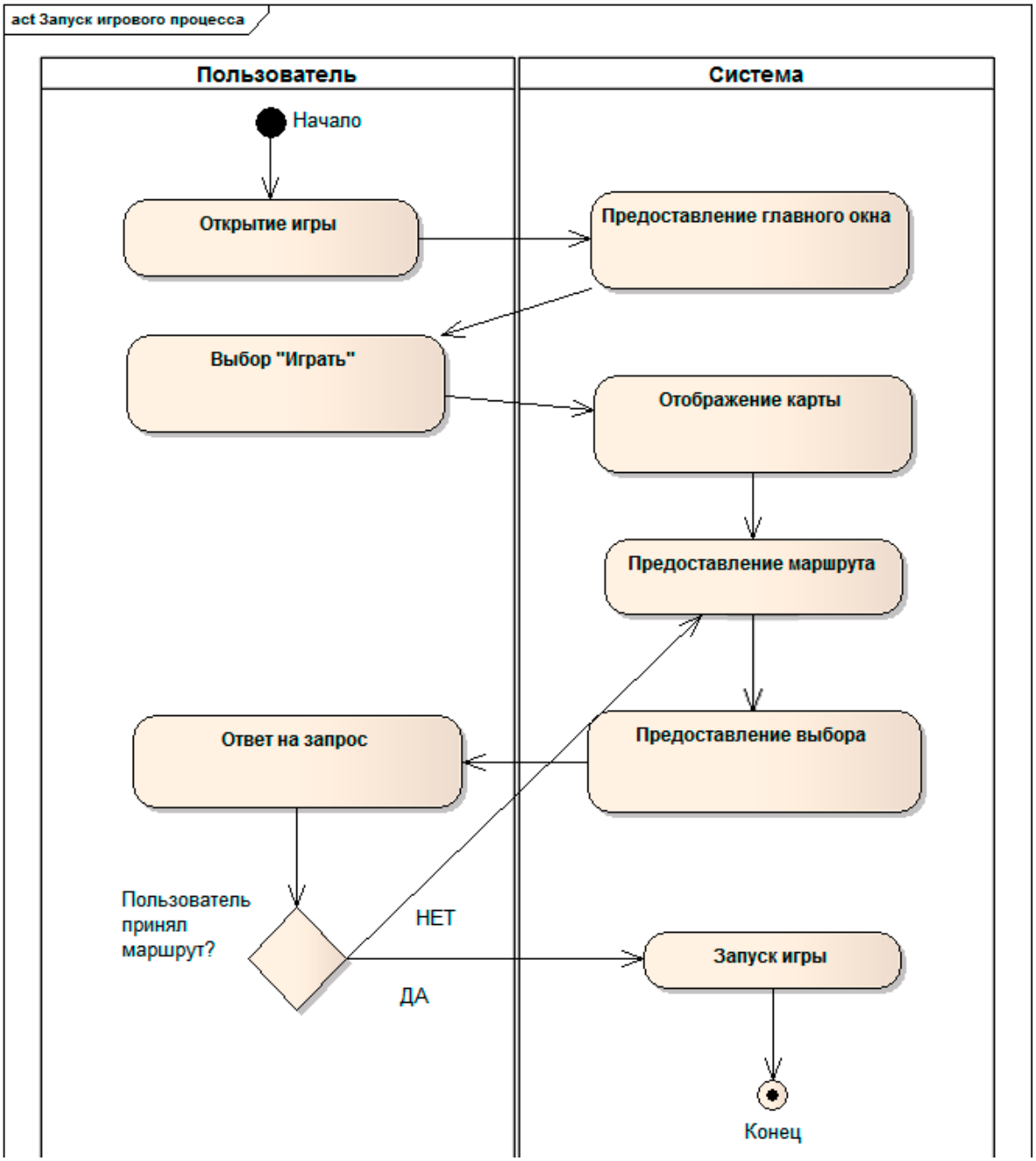

Рисунок 17 – Диаграмма деятельности для «Запуск игрового процесса»

Прецедент 3: Игровой процесс.

Входными данными являются выбранный пользователем маршрут.

В работе задействованы пользователь и система: на начальном этапе пользователь осуществляет выбор маршрута. Система предоставляет отображает маршрут на карте. Пользователь осуществляет движение транспорта по маршруту, система реагирует на это движение.

Диаграмма деятельности процесса запуска игры представлена на рисунке 18.

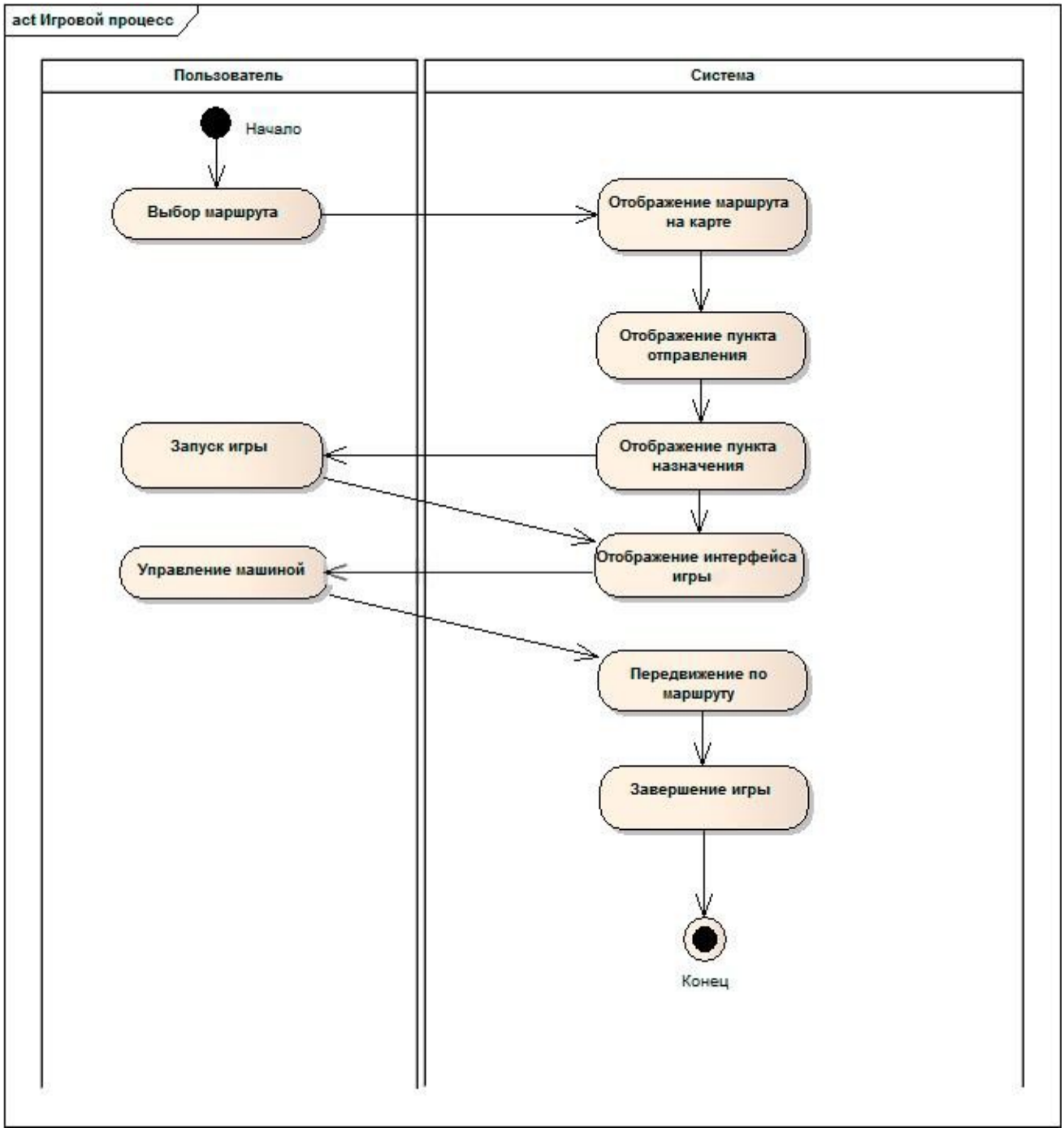

Рисунок 18 – Диаграмма деятельности для «Игровой процесс»

Прецедент 4: Составление таблицы результатов.

Входными данными являются введенное имя пользователя в другом процессе и результаты игры.

В работе задействованы пользователь и система. Пользователь проходит игру по выбранному маршруту и с введенным именем пользователя. Система осуществляет подсчет баллов данной игры и, на основании этой игры и предыдущих, составляет таблицу результатов.

Диаграмма деятельности процесса составления таблицы результатов продемонстрирована на рисунке 19.

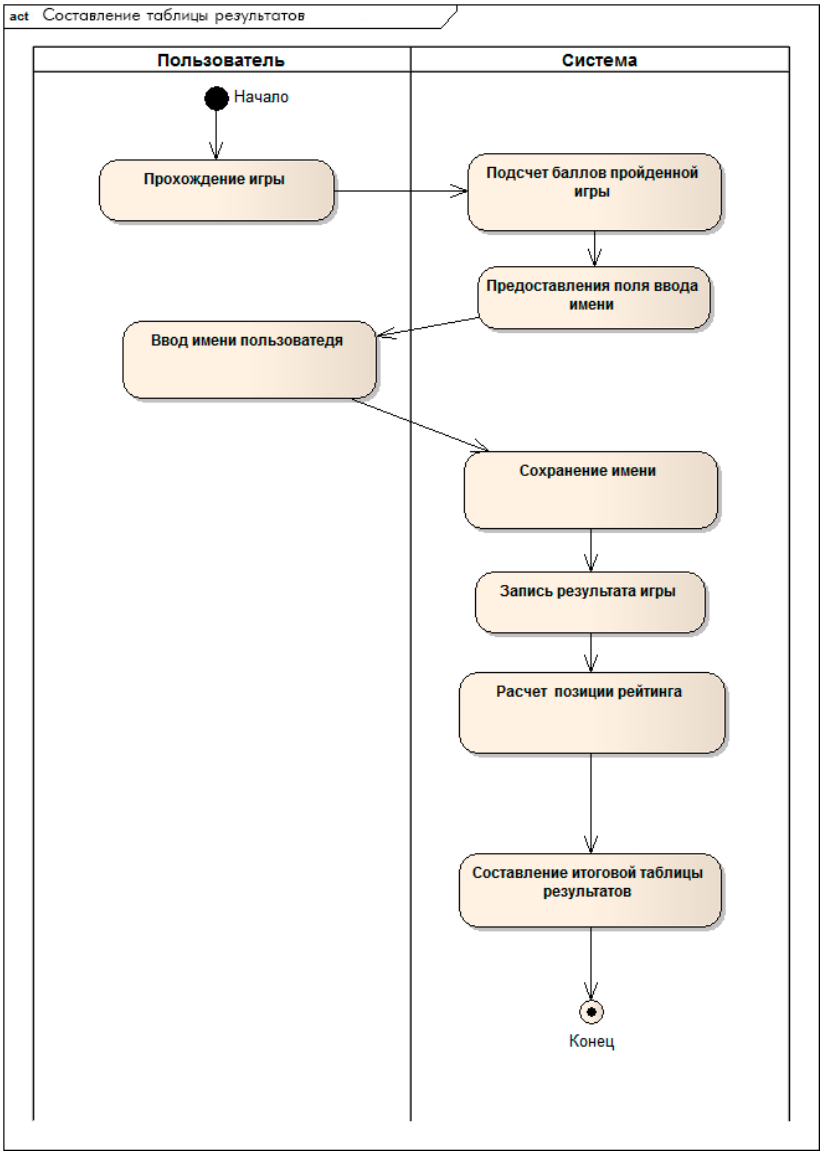

Рисунок 19 – Диаграмма деятельности процесса «Составление таблицы результатов»

Прецедент 5: Просмотр таблицы результатов.

Входными данными является введенный пользователем запрос на открытие таблицы рекордов.

В работе задействованы пользователь и система. Пользователь выполняет запрос на открытие страницы. Система осуществляет переход на нужную страницу.

Диаграмма деятельности процесса просмотра таблицы результатов продемонстрирована на рисунке 20.

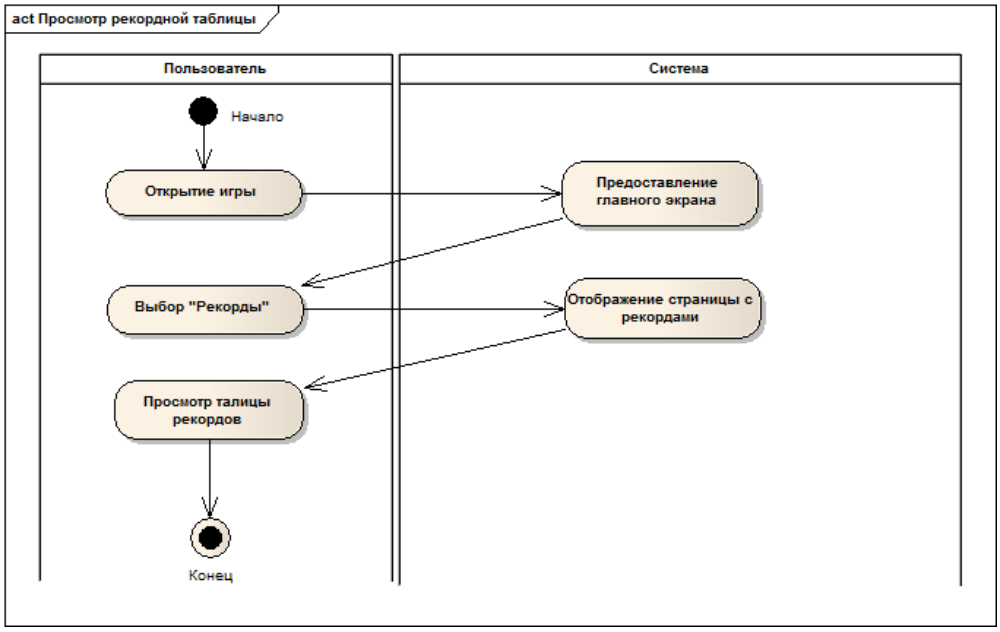

Рисунок 20 – Диаграмма деятельности процесса «Просмотр таблицы результатов»

2.2.3 Программная часть

Программную часть игры можно разделить на три слоя, которые представлены на рисунке 21.

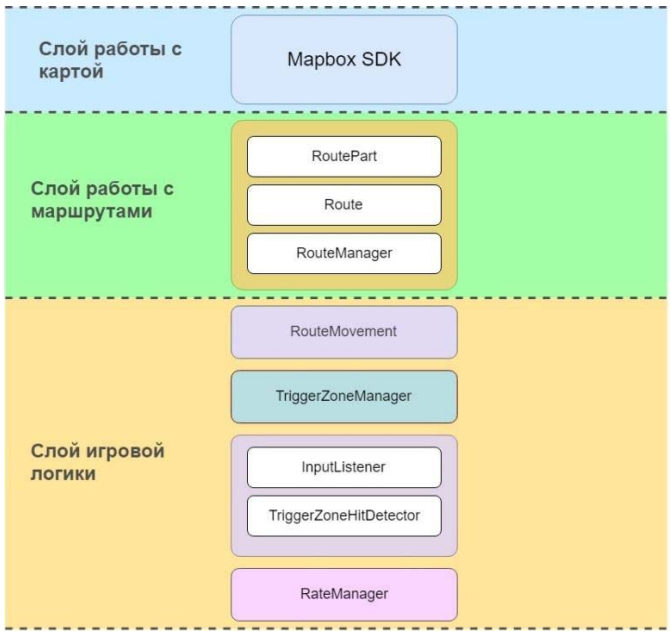

Рисунок 21 – Программная часть игры

1) Слой работы с картой состоит только из одного узла Mapbox SDK и осуществляет работу с сервисом Mapbox, предоставляя Unity через свой SDK возможность работать с картой.

2) Слой игровой логики состоит из 3 узлов и осуществляет работу с данными карты из Mapbox SDK. Этот слой описывает работу с понятием маршрут и управление маршрутами.

Первый узел (RoutePart) представляет собой класс по работе с географическими координатами. Класс занимается переводом географических координат в координаты Unity. Представляет базовую единицу маршрута и используется для отрисовки маршрута. Второй узел (Route) описывает уже саму сущность маршрута. Любой маршрут состоит из 2 частей: базовый маршрут (представлен единственным RoutePart) и списком RoutePart, необходимых для отрисовки активных зон. Третий узел (RouteManager) хранит список маршрутов и осуществляет управление маршрутами: переключение маршрутов и задание маршрута.

3) Слой игровой логики уже непосредственно отвечает за игровой процесс. Слой представлен 5 узлами.

Первый узел (RouteMovement) отвечает за движение по маршруту и является самой важной частью приложения. Вся работа дальнейших узлов завязана на принципах, порождаемых этим узлом. Идея такова, что любой маршрут представляется в виде интервала от 0 до 1. И в таком случае прогресс движения по маршруту будет лежать в этих границах. При таком подходе упрощается вся работа с приложением из-за того, что теперь нет необходимости работать с географическими точками напрямую. Заместо этого вся дальнейшая работа будет происходить на интервалах. Второй узел (TriggerZoneManager) осуществляет создание активных зон на текущем маршруте. Используя принцип интервалов, зоны создаются через сабинтервалы на промежутке от 0 до 1. Так, например, активную зону на первую половину маршрута можно записать как [0, 0.5]. Аналогично создаются и другие зоны. Третий узел (InputListener) осуществляет определение нажатий. Узел фиксирует начало и конец нажатия и

в обоих случаях оповещает об этом через события "Нажатие начато" и "Нажатие закончено". Четвертый узел (TriggerZoneHitDetector) занимается тем, что собирает данные о нажатии. В моменты начала и конца движения в переменные запоминаются значения текущего прогресса движения. После чего происходит событие о том, что данные о нажатии собраны. Пятый узел (RateManager) самый последний узел и занимается оценкой данных о нажатии, полученных через событие предыдущего узла. Идея такова, что данные о нажатии представляют собой интервал. Этот интервал проверяется на пересечение с интервалами, которые представляют собой активные зоны. Считается процент пересечения с интервалами. На основании полученного результата можно судить, на сколько успешна была пройдена текущая активная зона.

Общая структура системы показывается с помощью диаграммы классов, которая показывает их иерархию, кооперацию, атрибуты, методы интерфейс и взаимодействия между ними [17]. Данная диаграмма построена для графического представление статической структуры элементов системы. На диаграмме изображено семь классов:

- RouteMovement,
- RoutePart,
- Route,
- RouteManager,
- TriggerZoneManager,
- InputController,
- TriggerZoneInputDetector

Каждый класс имеет 3 блока:

1. Первый блок раскрывает имя класса.
2. Второй блок раскрывает атрибуты (переменные класса).
3. Третий блок раскрывает методы (функции и процедуры класса).

Диаграмма классов для данной игры представлена на рисунке 22.

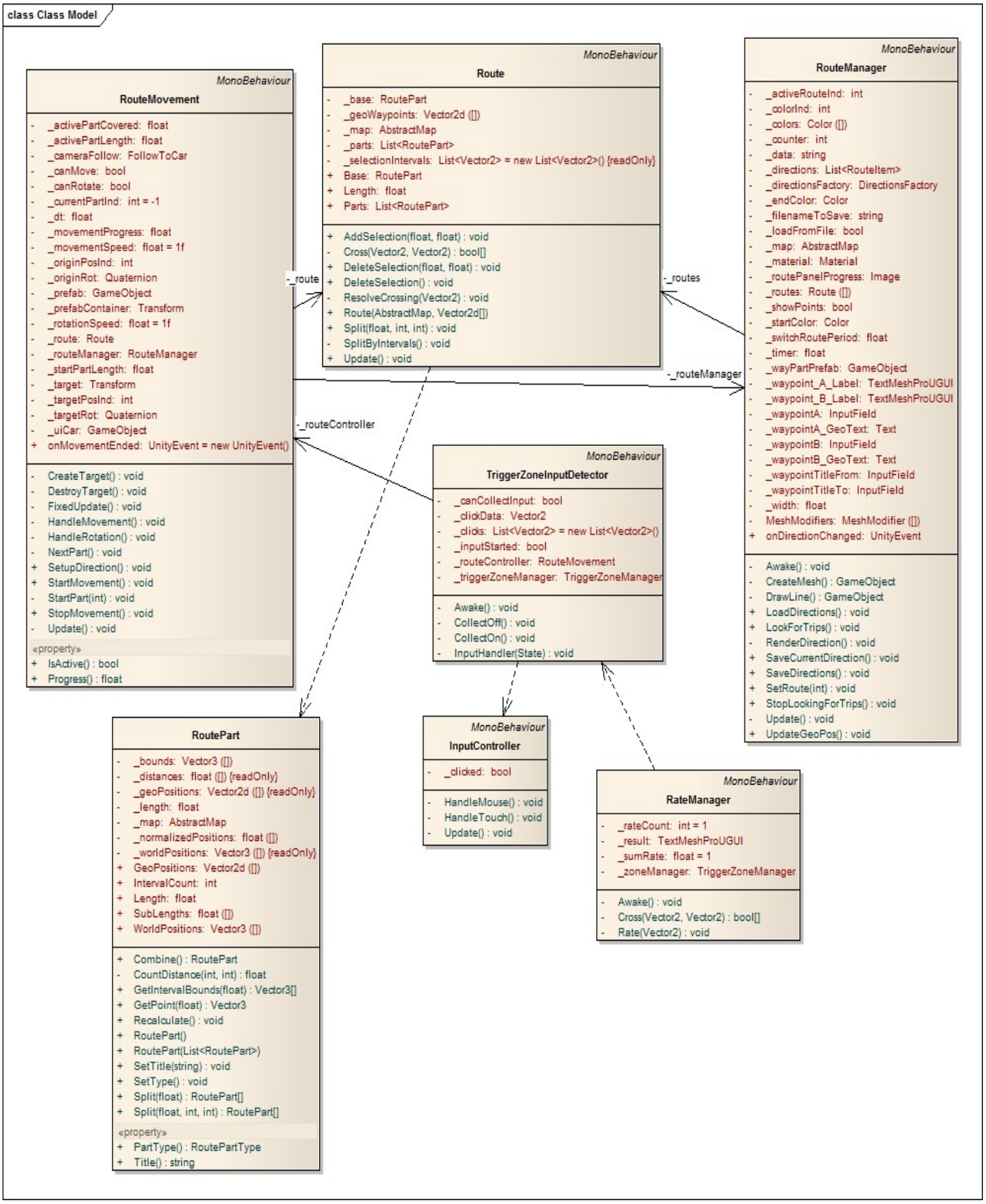

Рисунок 22 – Диаграмма классов

При описании классов, атрибутов и методов используются различные модификаторы доступа, которые позволяют задать допустимую область видимости для полей класса. На диаграмме 22 используется два типа модификатора:

1. Public – публичный класс или член класса, который доступен из любого места в коде и для других сборок (на диаграмме изображается, как знак «плюс»).

2. Private – закрытый тип. Класс или член класса доступен из кода только в том же классе или контексте (на диаграмме изображается, как знак «минус»).

2.3 Хранение маршрутов

В качестве хранения маршрутов используется формат json, который является текстовым форматом обмена данными. На рисунке 23 представлен пример того, как это выглядит. В данном случае представлен 1 маршрут, в игре имеется массив маршрутов. Каждый маршрут представлен массивом с точками с парами координат и двумя строками, которые представляют собой наименования пункта отправления и прибытия.

```
{
    "directions": [
        {
            "points": [
                {
                    "x": 56.02096,
                    "y": 92.89877
                },
                {
                    "x": 56.02125,
                    "y": 92.89966
                }
            ],
            "from": "ул. Белинского, 5",
            "to": "Ж/Д Вокзал Красноярск"
        }
    ]
}
```

Рисунок 23 – Хранение маршрутов

2.4 Работа с картой

Mapbox Unity SDK предоставляет необходимые инструменты для отображения карты в среде разработки Unity. Для этого создается пустой игровой объект и к этому объекту прикрепляется скрипт для работы с картой. На рисунке 24 происходит интеграция карты в Unity.

Рисунок 24 – Интеграция карты

Карта в среде Unity представлена в виде растрового изображения, которое, в свою очередь, разбито на множество частей одинакового размера. Эти части представляют собой квадратные изображения размера 256 на 256 пикселей. Также такие изображения еще называют тайлы (от англ. tile - плитка). В дальнейшем мы будем пользоваться именно таким названием.

Максимально прогружаемое в Unity число тайлов - 400 (по 10 в каждую сторону относительно центра привязки карты, который находится в центре левого берега Красноярска). Карта в Mapbox Unity SDK показана на рисунке 24.

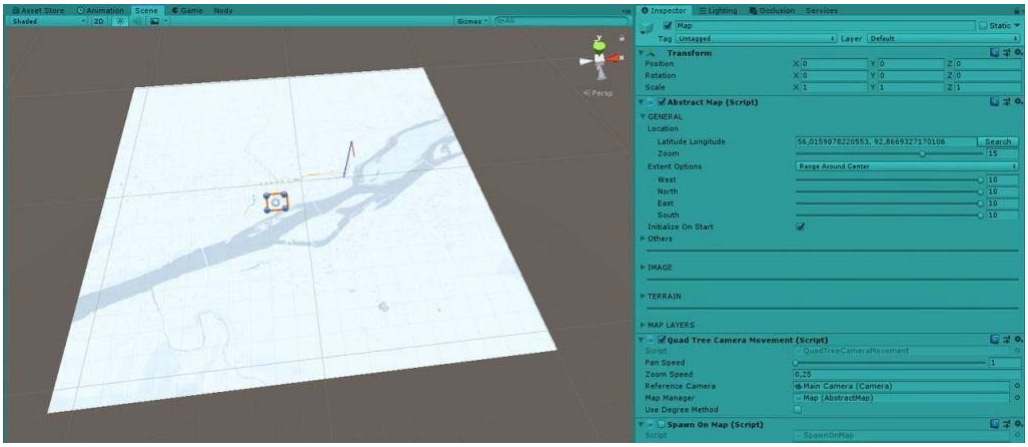

Рисунок 24 – Карта в Mapbox Unity SDK

В Unity мы можем настраивать, сколько тайлов одновременно будет прогружаться. На приведенном рисунке 25 число одновременно прогружаемых

тайлов было уменьшено в 4 раза, с 400 до 100 (по 5 в каждую сторону относительно центра привязки)

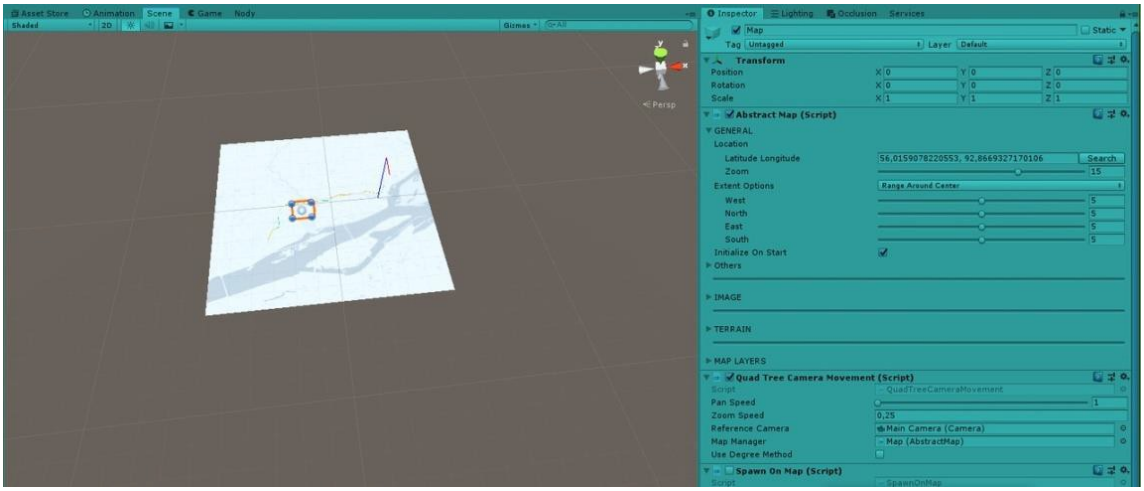

Рисунок 25 – Карта с уменьшенным числом тайлов

Следующим этапом является установка тайлов в границе видимости камеры. Этот процесс показан на рисунке 26.

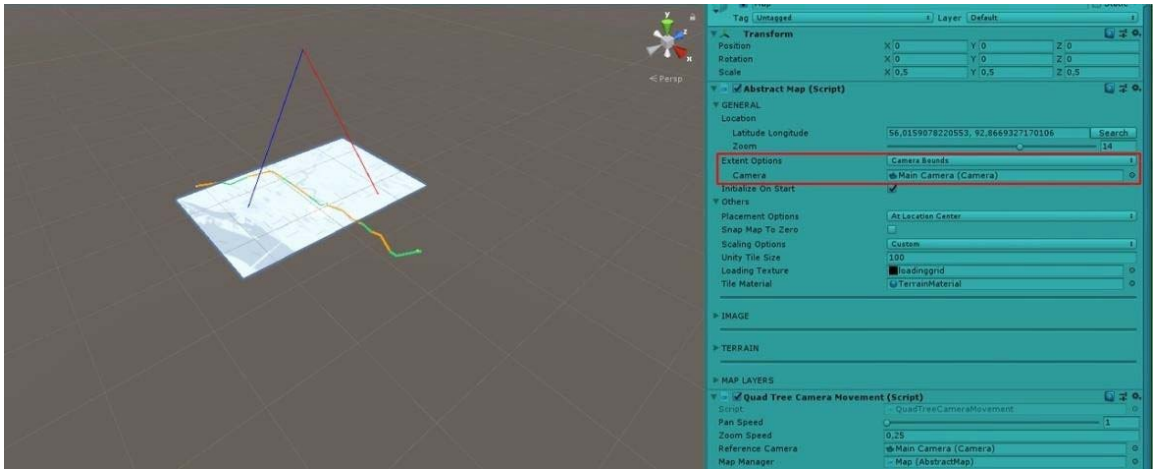

Рисунок 26 – Процесс установки тайлов в границе видимости.

От количества одновременно прогружаемых тайлов зависит сколько приложение будет потреблять ресурсов. Чем большее число одновременно загружается, тем больше ресурсов потребляется приложением и наоборот - чем меньше тайлов, тем меньше ресурсов. Потребляются ресурсы оперативной памяти, графического процессора и центрального процессора.

В игре все, что мы видим на экране устройства или в среде разработки, происходит за счет игровой камеры. Все, что происходит перед камерой, тоже

происходит и на экране. В разрабатываемом приложении карта расположена перед камерой на довольно близком расстоянии для удобства чтения информации с растровых изображений. Из-за небольшого расстояния между камерой и картой камера одновременно может отображать только от 4 до 6 тайлов карты. По этой причине, все тайлы, расположенные за границами видимости камеры, отключены и не прогружаются. Это было сделано в целях экономии ресурсов мобильного устройства.

Тайлы всегда находятся на одном и том же месте, двигается только камера. Камера движется за машиной, представленной в игре. Все тайлы, расположенные за границей видимости камеры не прогружаются и не отображаются. Это можно видеть по двум рисункам. На рисунке 27 отображены 6 одних тайлов, на рисунке 28 уже следующие тайлы.

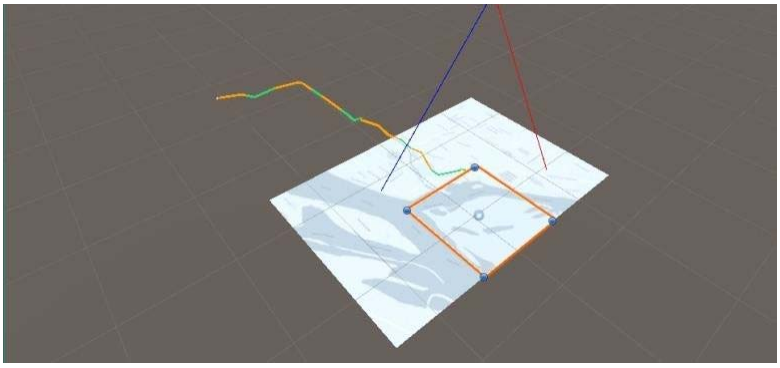

Рисунок 27 – Отображение тайлов

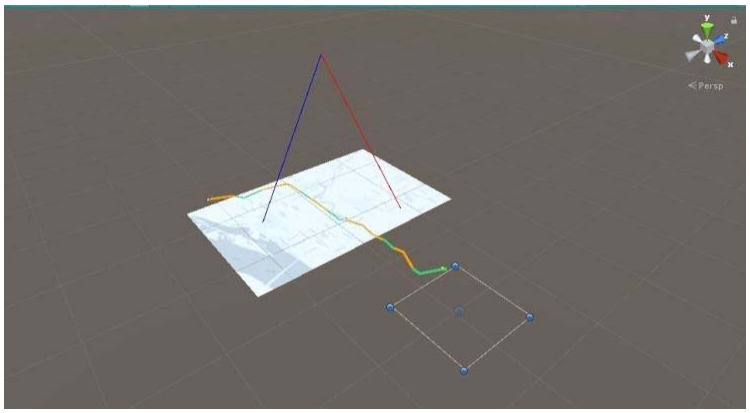

Рисунок 28 – Отображение тайлов

На рисунке 29 продемонстрирована зона видимости камеры.

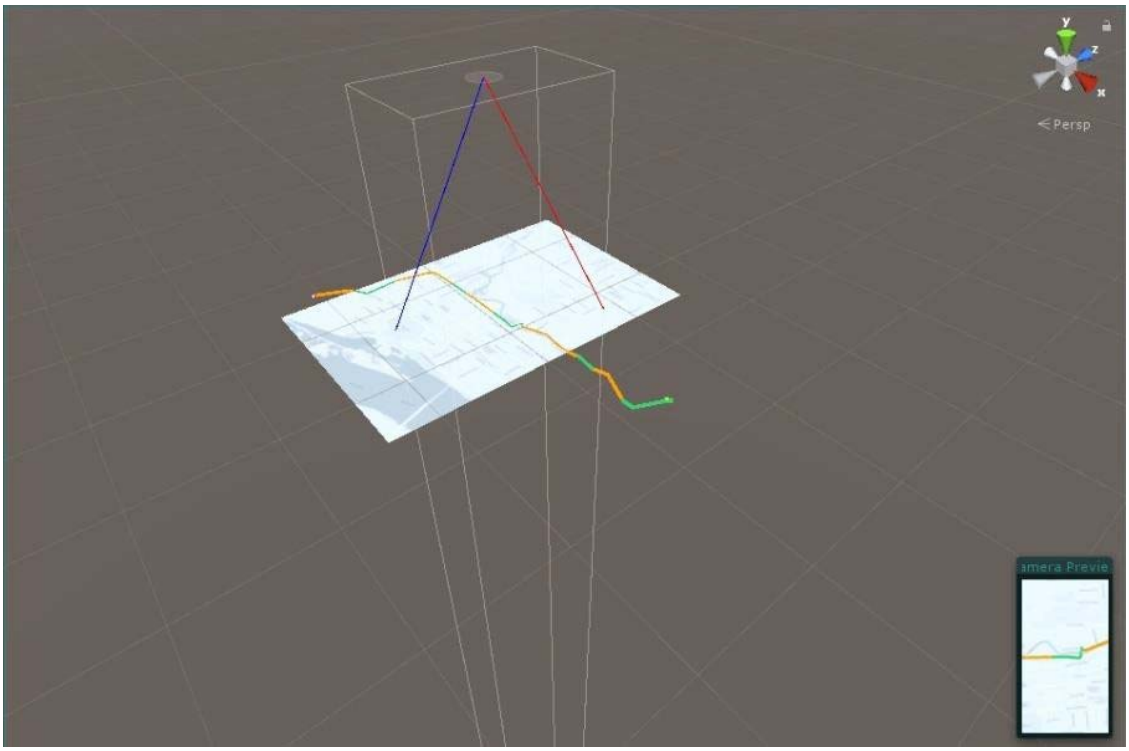

Рисунок 29 – Зона видимости камеры

2.5 Представление маршрута

Каждый маршрут в приложении представлен в виде массива из N точек. Число точек зависит от того, насколько геометрически сложен маршрут. Более прямой маршрут будет использовать меньшее число точек для отображения, а для маршрута с большим числом поворотов потребуется большее число точек. Для того, чтобы отобразить маршрут, между точками маршрута строятся отрезки. Число отрезков зависит от числа точек и всегда будет равно N-1. В Mapbox Unity SDK нет встроенного средства для отображения маршрутов, поэтому данный функционал реализовывался с нуля средствами Unity

Длину каждого отрезка можно рассчитать как расстояние между двумя точками, каждая из которых представлена в координатах Unity. Суммарная же длина маршрута будет представлять собой сумму всех длин отрезков, составляющих маршрут. Длину отрезка математически можно представить следующим образом, как показано в формуле 1.

$$li = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)}, \quad (1)$$

где x_1, y_1 - координаты начала отрезка, x_2, y_2 - координаты конца отрезка.

Длину маршрута можно представить по формуле 2.

$$L = \sum_{i=1}^{N-1} li, \tag{2}$$

где li - длина отрезка маршрута.

Теперь мы переходим к самой значимой части приложения - движение по маршруту. На работе данной части основывается почти вся дальнейшая игровая логика приложения. Само понятие маршрут можно перевести в пространство интервалов и работать с любым из маршрутов в виде интервала [0; 1]. Тогда в любой момент времени прогресс движения по маршруту будет представлять собой число из интервала от 0 до 1. Чем ближе к 0, тем мы ближе к началу маршрута и наоборот - чем ближе к 1, тем ближе к концу маршрута. Узнать этот прогресс движения можно следующим образом: считаем длину отрезков, которые пройдены полностью, и длину участка отрезка, который проходится в настоящий момент. Все это делится на полную длину пути и получается прогресс движения в виде числа в интервале от 0 до 1. С этого момента вся дальнейшая логика приложения построена на работе с этим значением. Все активные зоны представлены в виде интервалов. А визуально отображаются на маршруте зеленым цветом. В момент, когда машина заезжает в активную зону, пользователь нажимает на экран мобильного устройства. Приложение запоминает, какой прогресс движения был у машины. Затем, когда пользователь подходит к концу зоны, отпускает экран, приложение запоминает второе значение прогресса. Оба значения прогресса образуют интервал. Каждый такой интервал нажатия проверяется на пересечение с интервалами зон. И тот интервал зоны, с которым пересекается нужный интервал нажатия, проверяется на то, на сколько в процентном соотношении была пройдена зона.

2.6 Перевод координат

Перевод из географических координат, в координаты Unity представлен на рисунке 30.

```
// World Positions
for (int i = 0; i < _geoPositions.Length; ++i)
    _worldPositions[i] = _map.GeoToWorldPosition(_geoPositions[i]);
```

Рисунок 30 – Перевод координат

Имеется два массива:

Географические координаты – geoPositions.

Координаты Unity – worldPositions.

Эти 2 строчки в цикле переводят каждую географическую координату в координату Unity. Параметр _map – это карта Mapbox, внутри которой есть 2 метода для перевода координат:

1) GeoToWorldPosition перевод из географических координат в координаты Unity.

2) WorldToGeoPosition – перевод из Unity в географические координаты.

В данной главе рассмотрено проектирование и разработка мобильной игры. Для проектирования выбраны две методологи: SADT и UML. С помощью первой по средствам контекстной диаграммы и декомпозиции показана функциональная модель. С помощью второй методологии, с использованием диаграмм вариантов использования, деятельности и классов показаны функциональные возможности системы, алгоритм работы и последовательность действий, а также структура системы. Описаны слои программной части игры и процессы организации, представления, расчета длин и хранения маршрутов.

3 Описание работы игры

Результатом разработки является игра «GEO DRIVE GAME», работающая на базе ГИС-технологий, а именно, с использованием реальной карты города Красноярск. Приложение является мобильным и разработано под мобильную операционную систему Android. Главный экран приложения представлен на рисунке 24.

Рисунок 24 – Главный экран игры

На главном экране представлены 3 функциональные кнопки: «Рекорды», «Настройки» и «Играть». При нажатии на первую показывается статистика игр в следующем виде: на первом месте расположен участник с наибольшим количеством баллов, далее участники расположены по убыванию результатов прохождения игры. Страница с рекордами представлена на рисунке 25. Главный экран открывается при запуске игры с мобильного устройства, подключение к глобальной сети Интернет не требуется. Вместе с функциональными кнопками на экране отображается название игры «GEO DRIVER GAME» и логотип.

Рисунок 25 – Страница с рекордами

При отображении рекордов используются следующие поля:

–для отображения места в рекордной таблице,

имя – для отображения игрока,

% – для отображения результатов по пройденному маршруту, результат предоставляется в виде процентов.

Вторая кнопка «Настройки» позволяет пользователю управлять звуковыми настройками игры, включать и отключать музыку и звуковое сопровождение процессов. Для того, чтобы поставить индивидуальные настройки пользователю необходимо поставить или убрать галочку напротив интересующего пункта отображаемого интерфейса. Данный экран представлен на рисунке 26.

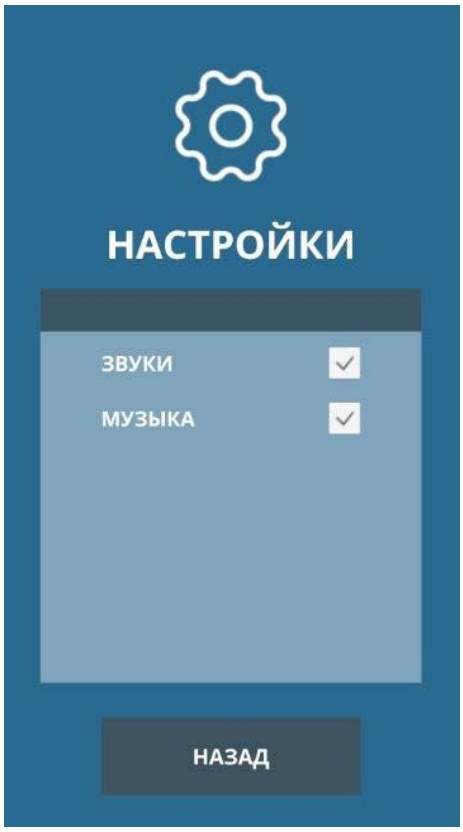

Рисунок 26 – Настройки игры

С двух рассмотренный экранов есть возможность вернуться на главный экран приложения при помощи использования кнопки «Назад». Третья функциональная кнопка главного экрана «Играть» запускает игровой процесс и открывает следующий интерфейс, представленный на рисунке 27. Пользователю случайным образом предлагаются маршруты для игры. Отображается пункт отправления и пункт назначения. Пользователь либо принимает маршрут и начинает игру, либо отклоняет его, после отклонения пользователю предлагается новый маршрут. Отображение маршрутов отображается на фоне реальной карты города Красноярска, именно на этой карте в дальнейшем начинается игра. На карте присутствуют улицы, реки с их наименованиями и здания, парки. С экрана игры можно осуществить выход на главный экран, для этого пользователю требуется выполнить переход по иконке «Домой». Для запуска игры пользователю необходимо выполнить переход по кнопке «Начать». А для выбора другого маршрута необходимо выполнить запрос по кнопке «Отклонить».

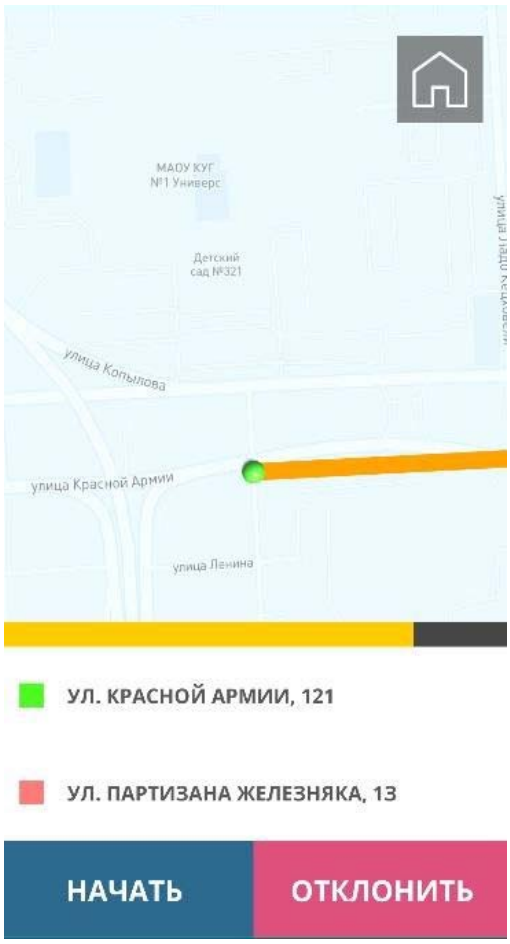

Рисунок 27 – Запуск игрового процесса

После выбора маршрута и запроса по кнопке «Начать» запускается игра. Началом игры является пункт отправления, концом игры является пункт назначения. Пользователю необходимо пройти весь маршрут от начала и до конца, выполняя повороты и различные маневры. Также существует возможность преждевременной остановки игры, с использованием кнопки «Завершить». Перемещение по карте осуществляется с использованием трех кнопок «вперед», «вправо», «влево».

После прохождения всего маршрута пользователь в верхней части экрана видит свои итоговые результаты, которые предоставляются в процентном виде. Проценты показывают насколько точно пользователь проехал выбранный им маршрут. На протяжении всей игры пользователь также видит результаты, но они являются не итоговыми, а промежуточными. Результаты данного действия показаны на рисунке 28.

Рисунок 28 – Результаты игры

После того, как пользователем завершена игра происходит подсчет рейтинга, рейтинг выводится на экран и приложение предлагает пользователю ввести имя. Интерфейс ввода имени показан на рисунке 29.

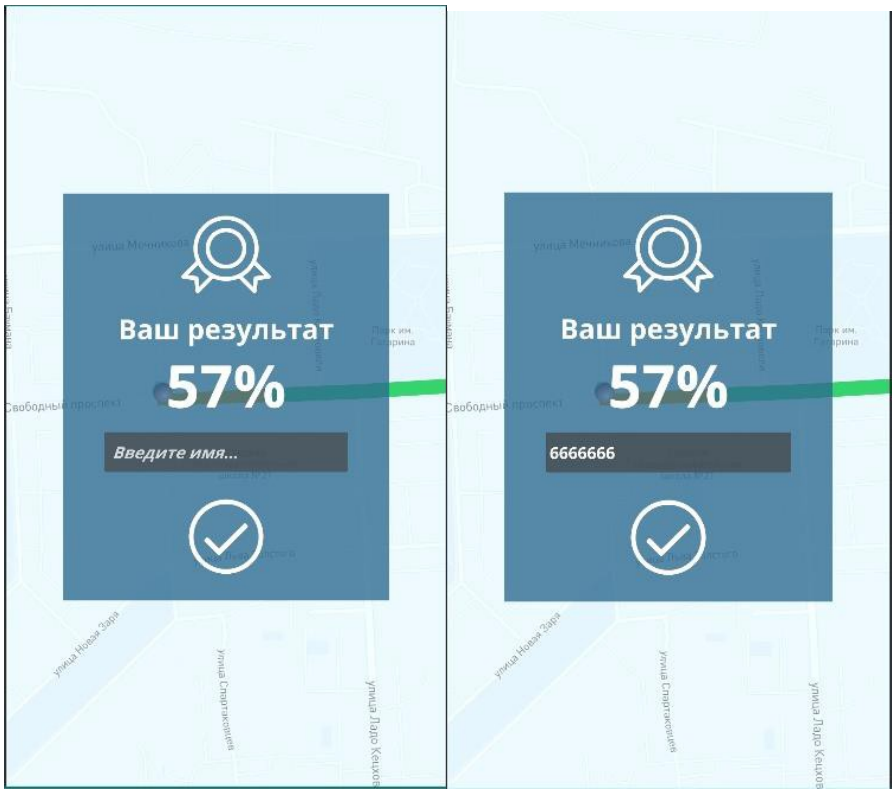

Рисунок 29 – Ввод имени пользователя

Для отладки игры на этапе всей разработки использовался эмулятор из среды разработки Android Studio. Android Studio является платформой для создания мобильных приложений на базе ОС Android.Эмулятор позволяет проверить и протестировать приложение на компьютере. Главным преимуществом эмуляторов является то, что они дают быстрый доступ к любой версии ОС, в данном случае к Android. Используемый эмулятор продемонстрирован на рисунке 30.

Рисунок 30 – Работа игры на эмуляторе Android Studio

Данный эмулятор может имитировать работу смартфона, планшета, часов Wear OS и устройств Android TV. Также он содержит в себе конфигурации популярных типов устройств, работает быстрее, чем реальное устройство, подключенное по USB.С его помощью можно имитировать звонки и сообщения, указывать местоположение, тестировать скорости мобильного интернета, повороты экрана, запускать приложения. Каждый экземпляр эмулятора использует Android Virtual Device (ADV), который представляет собой конфигурацию, определяющую характеристики устройства, которое нужно эмулировать. ADV содержит следующие компоненты: профиль устройства, образ системы, область хранения и внешний вид устройства.

ЗАКЛЮЧЕНИЕ

Игровая индустрия, став самостоятельной отраслью, достигла пика популярности в настоящее время. Для развлекательных целей разработчики внедряют в свои продукты различные технологии, в том числе используют и геоинформационные. Внедрение реальных карт делает игровой процесс интересней для пользователя.

Для разработки игры была выбрана популярная мобильная операционная система Android. В качестве языка разработки использовался язык программирования C#, а также работающая с ним среда Unity. В качестве подложки для игры использовалась видоизмененная карта местности города Красноярска OSM. Для стилевого оформления карты использовалась среда Mapbox Studio. Данные технологии позволили создать мобильное приложение с использованием реальной карты города, тестирование которого на каждом этапе разработки проверялось на эмуляторе Android Studio.

Если говорить о дальнейшем развитии игры, то полученные результаты прохождения маршрутов можно использовать для выявления трудных участков на дорогах.

СПИСОК СОКРАЩЕНИЙ

ГИС – Геоинформационные системы

ГИТ – Геоинформационные технологии

ПО – Программное обеспечение

ОС – Операционная система

AVD – Android Virtual Device

API – Application Programming Interface

OSM – OpenStreetMap

SADT – Structured analysis and design technique

QGS – Quantum GIS

UML – Unified Modeling Language

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. StarCouter [Электронный ресурс]. – Режим доступа: https://starcounter.com/

2. Марсикано К. UML. Android. Программирование для профессионалов/ К. Марсикано, К. Стюарт, Б. Филлипс – Питер: Символ-Плюс, 2017. –688 с.

3. Generation Streets [Электронный ресурс]. – Режим доступа: https://gamedev.ru/projects/forum/?id=238486

4. Ingress [Электронный ресурс]. – Режим доступа: https://www.ingress.com/

5. Resources [Электронный ресурс]. – Режим доступа: https://4pda.ru/forum/index.php?showtopic=693794

6. OpenStreetMap [Электронный ресурс]. – Режим доступа: https://www.openstreetmap.org/copyright/ru

7. Яндекс. Карты [Электронный ресурс]. – Режим доступа: https://bestmaps.ru/yandex-maps?k=google/hybrid

8. Google Maps [Электронный ресурс]. – Режим доступа: https://bestmaps.ru/google-maps?k=google/hybrid

9. Mapbox Studio [Электронный ресурс]. – Режим доступа: https://www.mapbox.com/unity/

10. Mapbox Unity SDK [Электронный ресурс]. – Режим доступа: https://www.mapbox.com/unity/

11. Шилдт Г, Java. Полное руководство / Г. Шилдт– Москва: Вильямс, 2017. – 1373 с.

12. Хелберг А, Язык программирования C#. КлассикаComputerScience / А. Хелдберг, М. Торгерсен, С. Вилтамут– Санкт-Петербург: Питер, 2011. –784с.

13. Дэвид Марка А, Макгоуэн К, Методология структурного анализа и проектирования SADT/ А. Дэвид Марка, К. Макгоуэн– М.: МетаТехнология, 1993. – 243 с.

14. Фаулер М, UML. Основы. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер – Питер: Символ-Плюс, 2006. – 192 с.

15. Рамбо Дж, UML. Руководство пользователя / Дж. Рамбо, Г. Буч, А. Джейкосон – Питер: ДМК пресс, 2004.– 248с

16. Диаграммы UML [Электронный ресурс]. – Режим доступа: https://grapholite.ru/uml/

17. Теория и практика UML. Диаграмма деятельности [Электронный ресурс]. – Режим доступа: http://it-gost.ru/articles/view_articles/96

18. Стандарт организации «Общие требования к построению, изложению и оформлению документов учебной деятельности» [Электронный ресурс]. – Режим доступа: http://about.sfu-kras.ru/node/8127.