

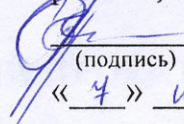
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н. П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная (цикловая) комиссия (кафедра) по
общепрофессиональным
и специальным (информационно-коммуникационным) дисциплинам

УТВЕРЖДАЮ

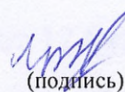
Зам. декана по учебной
работе, преподаватель ВКК


(подпись) В. А. Богатырская
« 4 » июня 2018 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ

на тему Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва)

Автор проекта


(подпись)

25.05.18
(дата)

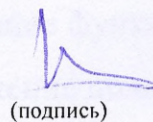
К. А. Архипов

Специальность 090203 Программирование в компьютерных системах

Обозначение дипломного проекта ДП - 02069964 - 090203 - 02 - 16

Руководитель проекта

канд. тех. наук, зав. каф.



(подпись)

25.05.18
(дата)

В. В. Никулин

Нормоконтролер

преподаватель ПКК

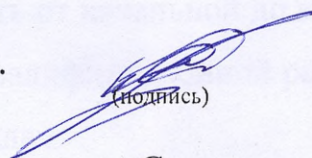

(подпись)

27.05.18
(дата)

М. Ф. Петянкин

Рецензент

канд. тех. наук, доцент. каф.


(подпись)

04.06.18
(дата)

С. Д. Шибайкин

Саранск
2018

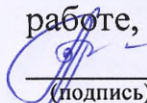
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная (цикловая) комиссия по общепрофессиональным
и специальным (техническим) дисциплинам

УТВЕРЖДАЮ

Зам. декана по учебной
работе, преподаватель ВКК
В. А. Богатырская


(подпись)

«15» марта 2018 г.

ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

Студент Архипов Кирилл Андреевич

1 Тема Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва)

Утверждено приказом № 2878–с от 12.04.2018

2 Срок представления работы к защите 24.05.2018

3 Исходные данные для научного исследования (проектирования)

Система должна быть реализована с удобным и понятным пользователю интерфейсом и выполнять следующие функции:

- определять начальное положение пользователя;
- определять окончательное положение пользователя;
- выполнять построение пути от начальной до конечной точки;
- корректно строить путь от начальной до конечной точки;

4 Содержание выпускной квалификационной работы

4.1 Анализ поставленных задач

4.1.1 Описание предметной области. Характеристика структурных

подразделений МГУ им. Н. П. Огарёва

4.1.2 Выбор инструментов и технологий, необходимых для реализации решаемых задач

4.1.3 Анализ спецификации программного модуля

4.1.4 Механизм обработки исключительных ситуаций

4.2 Проектная часть

4.2.1 Написание алгоритма волновой трассировки

4.2.2 Разработка пользовательского интерфейса

4.2.3 Руководство пользователя

4.3 Тестирование программного модуля

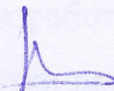
4.3.1 Результаты тестирования

5 Приложения:

Приложение А (обязательное) Технический паспорт двадцать седьмого корпуса

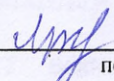
Приложение В (обязательное) Листинг запрограммированного дискретного рабочего поля

Руководитель работы


15.03.18
подпись, дата

В.В. Никулин

Задание принял к исполнению


15.03.18
подпись, дата

К. А. Архипов

РЕФЕРАТ

Пояснительная записка к дипломному проекту содержит 67 страниц, 14 рисунков, 3 задачи, 6 использованных источников, 2 приложения.

ПРОГРАММНЫЙ МОДУЛЬ, АЛГОРИТМ ВОЛНОВОЙ ТРАССИРОВКИ, ДИСКРЕТНОЕ РАБОЧИЕ ПОЛЕ.

Объектом исследования является использование современных информационных технологий в разработке программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D».





Цель работы – Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва).

В процессе работы использовались знания и навыки, полученные при изучении учебных курсов «Теория алгоритмов», «Основы программирования», «Технология разработки программного обеспечения», «Инструментальные средства разработки программного обеспечения».

В результате получен программный модуль навигационной системы, который полностью удовлетворяет всем поставленным требованиям и задачам.

Эффективность – навигация по структурному подразделению университета – двадцать седьмой корпус.

ДП – 02069964 – 090203 – 02 – 16

Изм.	Лист	№ докум.	Подпись	Дата				
Разраб	Архипов			25.05	Дипломный проект на тему «Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва)»	Лит	Лист	Листов
Провер	Никулин			25.05		Д	4	67
Н. Контр	Петянкин			27.05	ФГБОУ ВО «МГУ им. Н.П.Огарёва»			
Утверд	Богатырская			27.05				

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 Анализ поставленных задач	8
1.1 Описание предметной области. Характеристика структурных подразделений МГУ им. Н. П. Огарёва	8
1.2 Анализ плана структурного подразделения	9
1.3 Выбор инструментов и технологий, необходимых для реализации решаемых задач	12
1.4 Анализ спецификации программного модуля	15
1.5 Механизм обработки исключительных ситуаций	16
2 Проектная часть	17
2.1 Написание алгоритма волновой трассировки	17
2.2 Объекты пользовательского интерфейса	19
2.3 Структура сцен приложения	22
2.4 Программирование дискретного рабочего поля	24
2.5 Разработка методов	26
3 Тестирование программного модуля	34
3.1 Результаты тестирования	34
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	37
ПРИЛОЖЕНИЕ А (обязательное) Технический паспорт двадцать седьмого корпуса	38
ПРИЛОЖЕНИЕ Б (обязательное) Листинг запрограммированного дискретного рабочего поля	52

ВВЕДЕНИЕ

Федеральное государственное бюджетное образовательное учреждение высшего образования национальный исследовательский Мордовский государственный университет имени Н. П. Огарева является крупнейшим ВУЗом республики Мордовия. В университете обучается свыше двадцати тысяч студентов. В состав университета входит множество учебных корпусов, каждый со своей собственной структурой. Вследствие этого возникает проблема по нахождению необходимой аудитории или кабинета. Данная проблема актуальна для широкого круга лиц, к нему относятся: студенты младших курсов, не успевшие ещё познакомиться со структурой университета, абитуриенты, пришедшие на консультацию перед поступлением или осматривающие учебные корпуса, гости университета, приходящие по различным причинам. Решением проблемы является создание программного модуля навигационной системы.

Актуальность темы дипломный проект связана со значительным распространением исследуемого явления и заключается в необходимости разработки программного модуля навигационной системы для поиска аудиторий и кабинетов, входящих в структуру университета МГУ им. Н. П. Огарёва.

Объект исследования дипломного проекта – структура университета МГУ им. Н. П. Огарёва.

Предмет исследования дипломного проекта – навигационная система по поиску нужной аудитории или кабинета.

Цель дипломного проекта – создание программного модуля, позволяющего находить и строить путь до нужной аудитории или кабинета университета.

Задачи дипломного проекта:

А) исследовать и рассмотреть структуру организации корпусов университета;

									Лист
									6
Изм.	Лист	№ докум.	Подпись	Дата	ДП – 02069964 – 090203 – 02 – 18				

Б) проанализировать и выбрать наиболее подходящие методы нахождения аудиторий или кабинетов на основе исследования организации структуры университета;

В) создать программный модуль построения пути до необходимой аудитории и кабинета на основе исследования организации структуры университета.

Программный модуль будет интегрирован в навигационную систему университета, что позволит создать быструю и удобную систему по нахождению аудиторий или кабинетов университета.

Методами исследования дипломный проект являются:

- А) анализ;
- Б) абстрагирование;
- В) восхождение от абстрактного к конкретному;
- Г) моделирование;
- Д) описание;
- Е) обобщение.

Результатом разработки дипломного проекта является программный модуль, помогающий находить нужные аудитории или кабинеты, строить пути прохождения к ним с понятным и удобным пользовательским интерфейсом.

1 Анализ поставленных задач

1.1 Описание предметной области. Характеристика структурных подразделений МГУ им. Н. П. Огарёва

Мордовский государственный университет имеет восемнадцать структурных подразделений (семь институтов и одиннадцать факультетов). Ежедневно стены университета посещает множество людей незнакомых с его структурой, к ним относятся: абитуриенты, гости университета, первокурсники, которые ещё не успели познакомиться со структурой корпусов университета.

Каждое структурное подразделение университета включает в себя некое количество этажей, на каждом из которых расположены аудитории и кабинеты. Каждая аудитория и кабинет имеет свой уникальный номер. Аудитории имеют названия в том случае, если являются лабораториями.

Проблема, решаемая дипломным проектом, – нахождение необходимой аудитории или кабинета в том или ином корпусе университета посредством создания программного модуля с удобным и понятным пользовательским интерфейсом.

Разработки в данной сфере уже ведутся [3-4]. Отличительной чертой предлагаемой работы является построение пути из стартовой в финишную точку без использования таких технологий, как GPS, WI-FI и ГЛОНАСС. Отказ от данных технологий осуществлен по следующим причинам.

GPS позволяет определять местоположение пользователя с использованием GPS модуля. Точность определения местоположения составляет от пяти до десяти метров, что в реалиях одного здания слишком большая погрешность, это обусловлено тем, что расстояние между аудиториями и кабинетами меньше погрешности. Связь со спутником GPS внутри здания может быть невозможна ввиду толщины крыши и стен.

ГЛОНАСС является отечественным аналогом GPS, следовательно, возникающие проблемы аналогичны. Погрешность при определении местоположения слишком велика.

WI-FI – технология беспроводной локальной сети с устройствами на основе стандарта IEEE 802.11. На основе WI-FI возможно осуществлять определение местоположение точек в пространстве, используя отпечатки тепловых карт. Способ напрямую зависит от покрытия сети WI-FI и её работоспособности. Не все корпуса университета полностью покрыты сетью WI-FI, также для построения точных тепловых карт необходима WI-FI сеть на основе стационарных точек.

1.2 Анализ плана структурного подразделения

Каждое структурное подразделение университета имеет план, в котором отмечен метраж всех помещений, коридоров, расположение всех лестниц, лифтов, туалетов, переходов.

В рамках разработки приложение охватывает навигацию по двум этажам двадцать седьмого корпуса: по второму и третьему. На рисунке 1 представлен план второго этажа.

В нижней части плана представлен переход в первый корпус, который продолжается дальше и заканчивается вторым переходом в шестнадцатый корпус. В нижнем правом углу представлен женский и мужской туалеты, напротив туалетов располагается проход к лифту.

					ДП – 02069964 – 090203 – 02 – 18	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

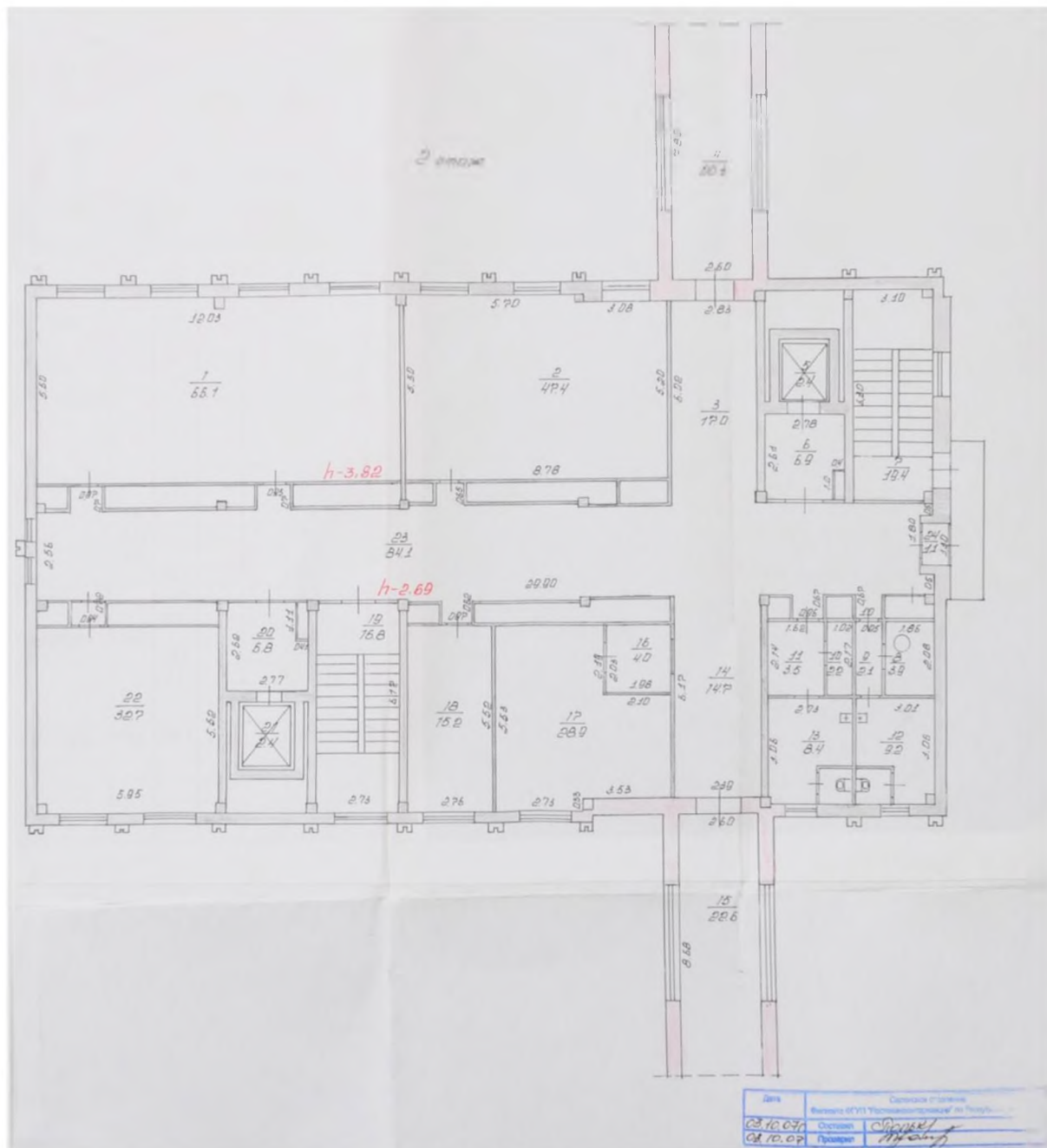


Рисунок 1 – План второго этажа двадцать седьмого корпуса

Весь этаж разделен горизонтальным коридором, по обе стороны которого расположены аудитории. По правую сторону располагаются аудиторию под номерами:

- А) 203;
- Б) 205;
- В) 207.

Изм.	Лист	№ докум.	Подпись	Дата

ДП – 02069964 – 090203 – 02 – 18

Лист

10

По левую сторону расположены: лестница, проход к лифту и аудитории под номерами:

- А) 202;
- Б) 204;
- В) 206.

На рисунке 2 представлен план третьего этажа двадцать седьмого корпуса.

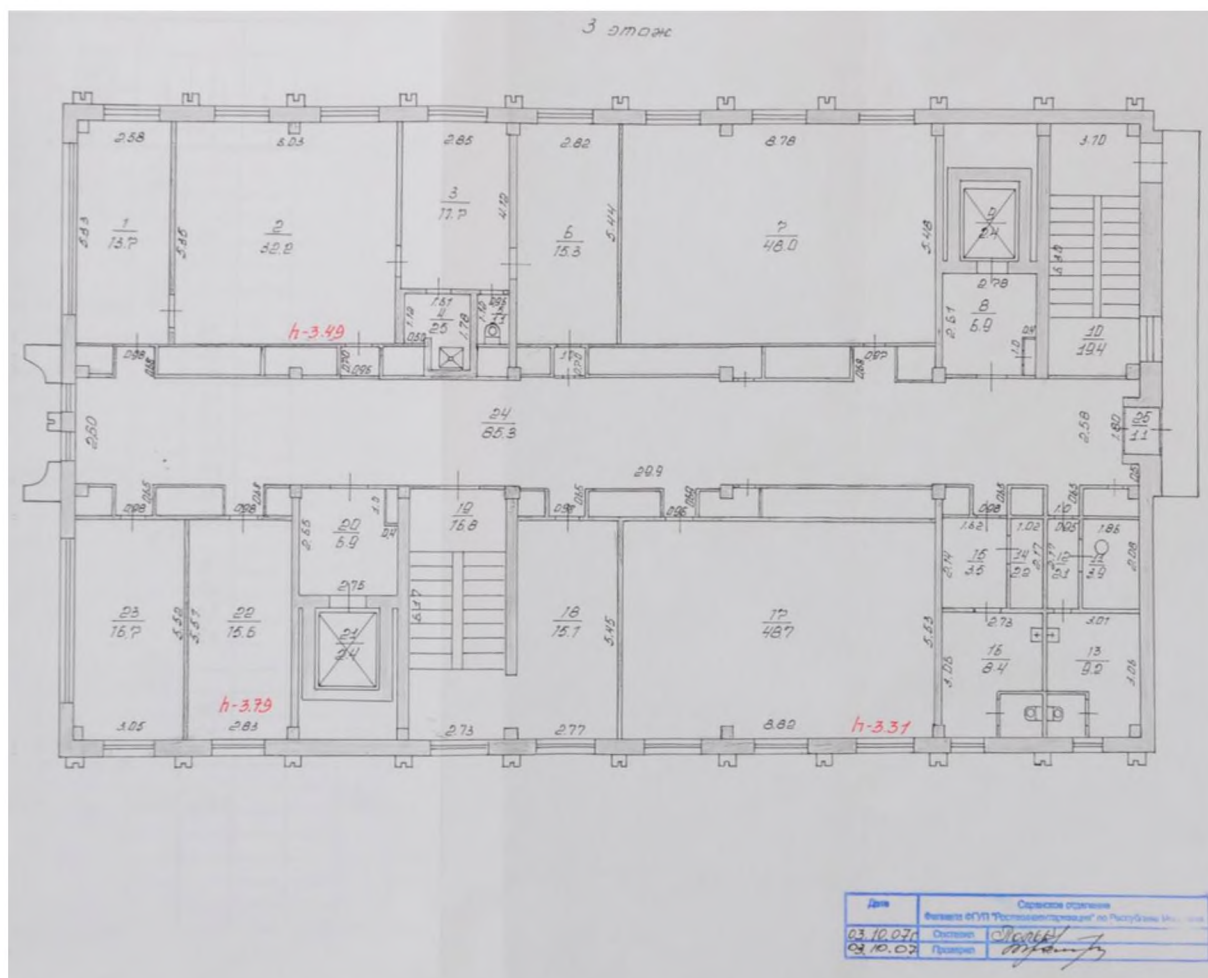


Рисунок 2 – План третьего этажа двадцать седьмого корпуса

Весь этаж пересекает горизонтальный коридор, по сторонам которого расположены: аудитории, кабинеты, лестница, лифты, туалеты.

На правой стороне коридора располагается переход к лифту и аудитория под номером 305, кабинет под номером 302.

С левой стороны от коридора расположены:

- А) мужской туалет;

- Б) женский туалет;
- В) аудитория 306;
- Г) лестница;
- Д) переход к лифту.

Структура второго и третьего этажа схожа: проходы к лифтам и лестнице расположены одинаково, туалеты расположены друг под другом. Каждый из этажей пересекает непрерывный горизонтальный коридор, по обе стороны которого расположены все остальные объекты. Технический паспорт двадцать седьмого корпуса представлен в приложении А.

1.3 Выбор инструментов и технологий, необходимых для реализации решаемых задач

Выбор языка

Среди всех языков программирования для реализации дипломного проекта выбран язык С#. Главной причиной выбора является наличие опыта разработки на данном языке, удобство синтаксиса языка, поддержка всех необходимых технологий.

С# – объектно-ориентированный язык программирования, который относится к семье языков с С-подобным синтаксисом, разработанный компанией Microsoft. Язык поддерживает:

- А) статическую типизацию переменных;
- Б) поддержку полиморфизма;
- В) перегрузку операторов, включая операторы явного и не явного приведения типа данных;
- Г) делегаты;
- Д) атрибуты событий;
- Е) обобщенные типы и методы;

						<i>Лист</i>
					<i>ДП – 02069964 – 090203 – 02 – 18</i>	<i>12</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

Ж) исключения.

C# выбран потому что, обладает понятным и знакомым синтаксисом, имеется наличие поддержки всего необходимого функционала, технологий и опыт разработки на данном языке.

Выбор среды разработки пользовательского интерфейса

Такие технологии, как Unity3D, Cocos2D, Qt, SFML предоставляют возможности по разработке пользовательского интерфейса, по обработке событий, по написанию и встраиванию скриптов. Каждая из технологий межплатформенная, то есть позволяет создавать единое приложение для более чем одной платформы, что позволяет ускорить процесс разработки приложения на различные платформы. Поддерживаются такие платформы, как:

- А) Android;
- Б) IOS;
- В) Windows;
- Г) MacOS.

Cocos2D – кроссплатформенный фреймворк, используемый для разработки интерактивных приложений и игр. Является open-source проектом, позволяющим работать с анимацией, графическим интерфейсом пользователя, физикой объектов, звуками и скриптами. Разработка осуществляется на таких языках программирования, как:

- А) C++;
- Б) LUA;
- В) JavaScript.

Cocos2D не подходит для реализации дипломного проекта, так как не поддерживает язык программирования C#.

Qt – кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++. Qt позволяет разрабатывать единое приложение для большинства современных операционных систем,

имеет все основные классы, необходимые при разработке прикладного программного обеспечения, начиная с разработки графического интерфейса и заканчивая классами по работе с базами данных и XML.

Qt так же, как и Cocos2D не поддерживает C#, следовательно, не подходит для реализации дипломного проекта.

SFML – open-source кроссплатформенная мультимедийная библиотека. Содержит в себе ряд модулей для простого программирования игр и мультимедиа приложений, поддерживает множество языков программирования:

- А) C;
- Б) D;
- В) Java;
- Г) Python;
- Д) Ruby;
- Е) OCaml;
- Ж) .Net;
- З) Go.

Хоть SFML и предоставляет возможность разработки приложений на C# с использованием фреймворка .Net, но этот способ заставляет включать дополнительную технологию и среду разработки CLR, что приводит к однозначному усложнению всего проекта и его реализации.

Unity3D – межплатформенная среда разработки приложений, которая позволяет разрабатывать приложения, работавшие под более чем двадцать различных операционных систем. Основное преимущество Unity3D – наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов.

Unity3D предоставляет полный функционал для разработки приложений, поддерживает разработку на C#, поэтому в качестве технологии реализации дипломного проекта выбрана именно она.

Для написания и редактирования скриптов, написанных на C#, используется интегрированная среда разработки Microsoft Visual Studio 2017. Unity и Microsoft Visual Studio совместимы и работают в связке, что позволяет обеспечить бесшовность при разработке или редактирование программного проекта.

1.4 Анализ спецификации программного модуля

На этапе разработки программного модуля план структурного подразделения университета анализируется. На плане должны быть отмечены аудитории и кабинеты с метражом, все входы и выходы с этажа, в том числе лестница и лифт.

На этапе эксплуатации программного модуля пользователем входными данными являются две точки: стартовая и финишная. Обе точки должны быть либо входом в аудиторию или кабинет, либо лифтом, либо лестницей, либо переходом в другое структурное подразделение университета, либо входом или выходом.

Выходными данными работы программного модуля является построенный путь из стартовой в финишную точку, если такой путь возможен, в противном случае пользователь увидит сообщение о причине невозможности построения пути.

Функционал программного модуля включает в себя:

- А) предоставление пользовательского интерфейса;
- Б) определение стартовой точки;
- В) определение финишной точки;
- Г) построение кратчайшего пути от стартовой точки к финишной точки;

- Д) отображение построенного пути в интерфейсе программного модуля;
- Е) очистка пройденного пути;
- Ж) переход к построению следующего пути;
- З) сообщение пользователю подсказок в виде тестовых полей для корректного ввода данных;
- И) сообщению пользователю об ошибках в исключительных ситуациях.

1.5 Механизм обработки исключительных ситуаций

При работе программного модуля возможны критические и не критические результаты работы.

Критические результаты работы программного модуля – ситуации при которых входные данные слишком велики или невозможны для обработки алгоритмом программного модуля. Если пользователь введет точку, которой нет на плане, или точкой будет несуществующий кабинет, лестница, лифт, переход, то пользователь получит сообщение об ошибке, далее некорректное значение будет стёрто и пользователь получит подсказку о корректном вводе точки.

К некритичным результатам работы программного модуля относятся ситуации, приводящие к корректному завершению программы. В таком случае пользователь получить сообщение об успешном построении пути.

2 Проектная часть

2.1 Написание алгоритма волновой трассировки

Волновой алгоритм трассировки подразумевает использование двумерной клетчатой карты (матрицы). Все клетки карты делятся на стартовую и финишную клетку, а также на подмножества проходимых и непроходимых клеток. Целью алгоритма является построение минимального пути от стартовой до финишной клетки, проходя исключительно по клеткам из проходимого подмножества, исключая клетки из непроходимого подмножества.

Двумерная клетчатая матрица может быть двух видов: окрестность Мура, представленная на рисунке 3, и окрестность фон Неймана, представленная на рисунке 4. С целью упрощения на первом этапе используется окрестность Мура (клетки исходной матрицы квадратные).

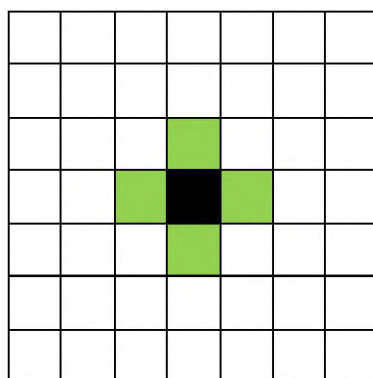


Рисунок 3 – Окрестность Мура

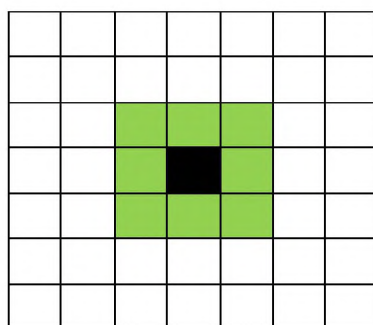


Рисунок 4 – Окрестность фон Неймана

Алгоритм волновой трассировки состоит из трех частей:

А) Инициализация – определение стартовой и финишной клетки, заполнение проходимого и непроходимого подмножества;

Б) Распространение волны – рассмотрение каждой ячейки, начиная от стартовой;

В) Восстановление пути – составление пути из финишной в стартовую клетку, проходя по клеткам с минимальным атрибутом.

Алгоритм волновой трассировки.

А) Скопировать размеченную карту размера $N \times M$ в рабочий массив $DRP(N \times M)$;

Б) Каждому из элементов рабочего массива $DRP(i, j)$ присвоить некоторое значение, в зависимости от свойства соответствующего атрибута ячейки карты, по следующему правилу:

1) Если ячейка размеченной карты, соответствующая элементу рабочего массива $DRP(i, j)$ непроходима, то $DRP(i, j) = -2$;

2) Если ячейка размеченной карты, соответствующая элементу рабочего массива $DRP(i, j)$ проходима, то $DRP(i, j) = -1$;

3) Если ячейка размеченной карты, соответствующая элементу рабочего массива $DRP(i, j)$ «стартовая», то $DRP(i, j) = N \times M$;

4) Если ячейка размеченной карты, соответствующая элементу рабочего массива $DRP(i, j)$ «финишная», то $DRP(i, j) = 0$;

В) Начало этапа распространения волны. Вводится переменная N_i – счетчик итераций, инициализируется 0;

Г) Вводится переменная N_k – константа максимальное количество возможных итераций равна $N \times M$;

Д) Построчно просматривается рабочий массив $DRP(i, j)$;

Е) Если $DRP(i, j) = N_i$, то рассматриваются соседние элементы по следующему правилу (на примере элемента $DRP(i+1, j)$):

1) Если элемент $DRP(i+1, j)$ непроходим, то переход к пункту Л;

2) Если элемент $DRP(i+1, j)$ проходим, то $DRP(i+1, j) = N_i + 1$;

3) Во всех остальных случаях $DRP(i+1, j)$ остаётся без изменений.

Ж) Аналогичные действия производят со всеми элементами;

З) После завершения построчного просмотра всех элементов рабочего массива переменная N_i увеличивается на единицу;

И) Если $N_i > N_k$, то путь построить невозможно – завершение работы алгоритма;

К) Переход к пункту Д;

Л) Этап восстановления пути. Вводятся переменные X и Y , равные координатам стартовой точки;

М) В окрестности $DRP(X, Y)$ находится наименьший элемент, координаты элемента заносятся в переменные $X1$ и $Y1$;

Н) Перемещение по карте из клетки с координатами $[X, Y]$ в $[X1, Y1]$;

О) Если $DRP(X1, Y1) = 0$, то переход к пункту П;

П) Выполняется присваивание $X = X1, Y = Y1$.

Сложность волнового алгоритма составляет $O(M*N)$, где N – число клеток дискретного рабочего поля, M – число точек, подлежащих соединению.

2.2 Объекты пользовательского интерфейса

Разработка пользовательского интерфейса осуществляется средствами кроссплатформенной среды разработки Unity3D, которая предоставляет широкий спектр готовых объектов пользовательского интерфейса. Unity3D поддерживает разработку интерфейса, как в режиме 2D, то есть на плоскости, исключая высоту объектов, так и в режиме 3D, то есть каждый объект обладает высотой. Объекты пользовательского интерфейса в Unity3D,

объединены в отдельную группу – UI. Главное окно среды разработки представлено на рисунке 4.

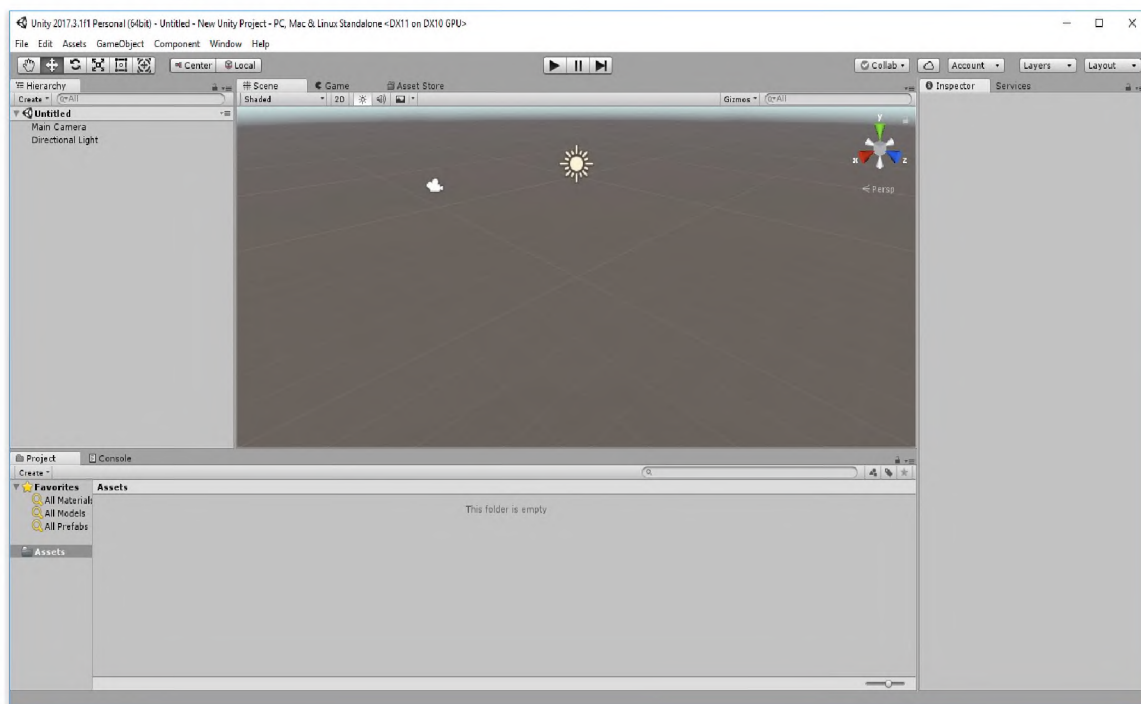
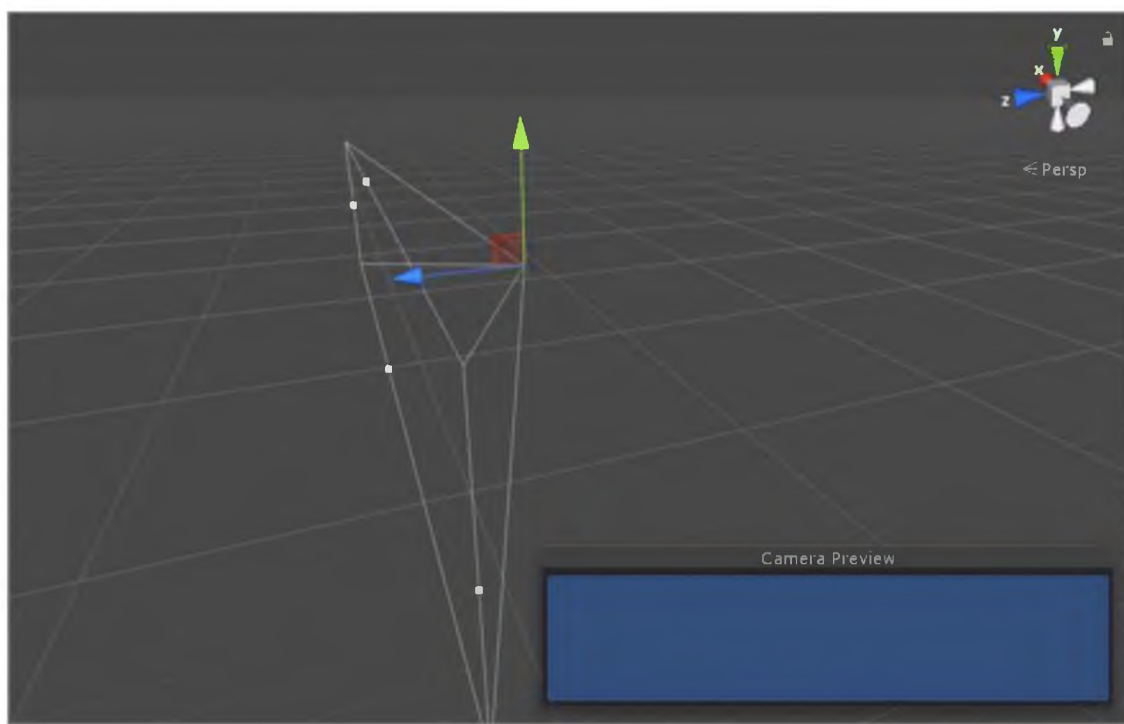


Рисунок 4 – Главное окно среды разработки

Одним из основных объектов Unity3D является Camera. Camera захватывает и отображает пользователю окно приложения. Объект представлен на рисунке 5.



Изм.	Лист	№ докум.	Подпись	Дата

ДП – 02069964 – 090203 – 02 – 18

Лист

20

Рисунок 5 – Объект Camera

Объект Canvas представляет собой абстрактное пространство, в котором производятся настройки и отрисовка всех объектов пользовательского интерфейса, следовательно, работа с пользовательским интерфейсом невозможна без Canvas. Все объекты пользовательского интерфейса обязаны быть привязаны к Canvas, то есть закреплены на нем. Объект представлен на рисунке 6.

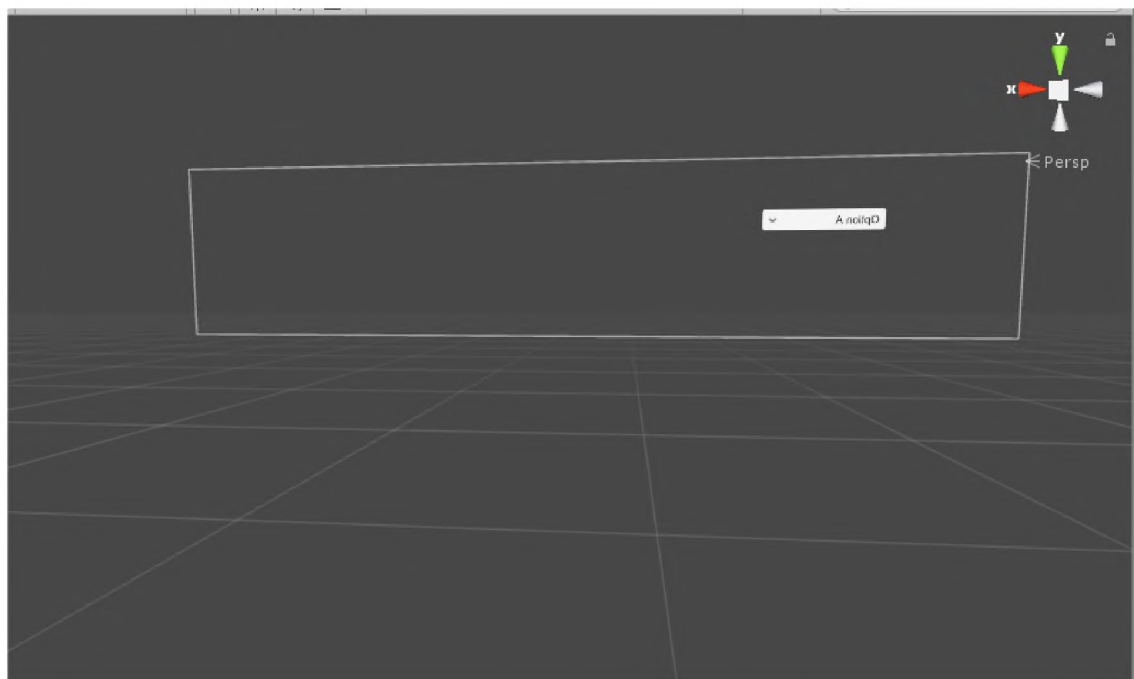


Рисунок 6 – Объект Canvas

Dropdown – объект, представляющий выпадающие меню. Каждый из элементов представляет собой: лестницу, лифт, переход в другой корпус, номер аудитории или кабинета. Объект представлен на рисунке 7.



Рисунок 7 – Объект Dropdown

Объект Text представляет из себя прозрачное поле, в которое может быть записан текст. Объект представлен на рисунке 8.

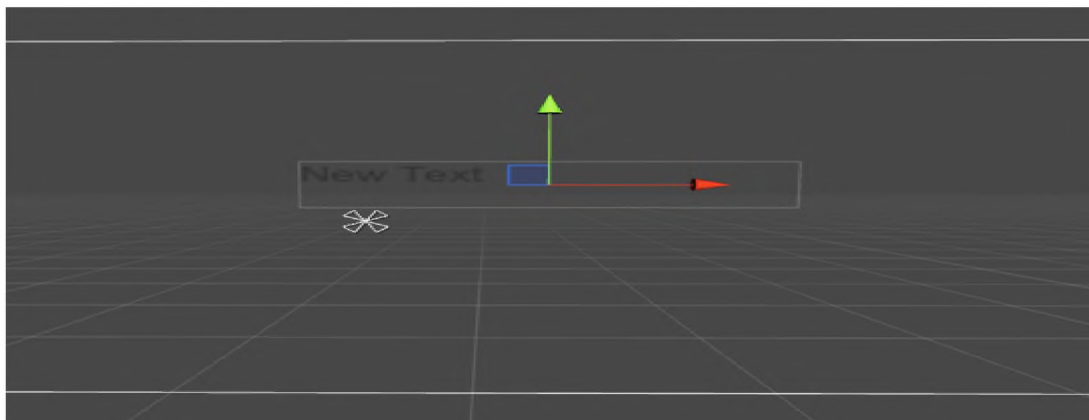


Рисунок 8 – Объект Text

В стандартном наборе объектов пользовательского интерфейса кнопка представлена объектом Button, который обладает всем необходимым функционалом. Объект представлен на рисунке 9.

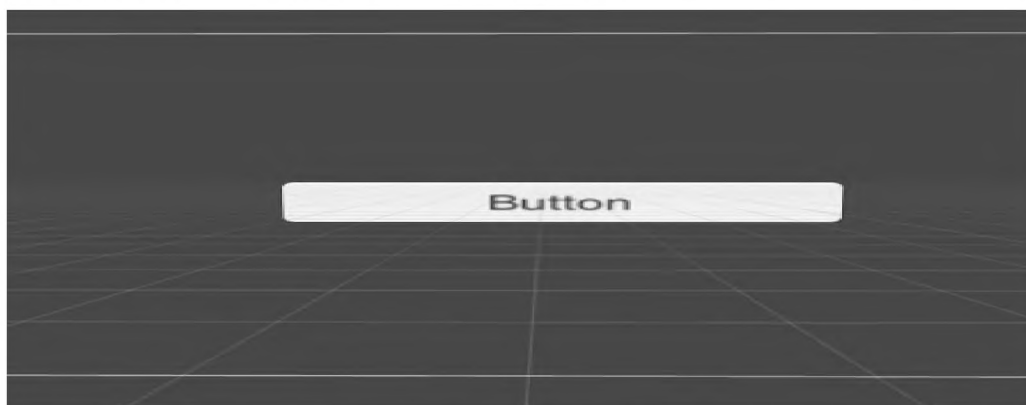


Рисунок 9 – Объект Button

2.3 Структура сцен приложения

При разработке в Unity3D рабочее поле и совокупность всех объектов на нем называется сценой. Всего в приложении насчитывается две сцены.

Сцены пользовательского интерфейса разработаны на основе макетов. На макете содержатся объекты определённых типов, все объекты подписаны. Макет первой сцены представлен на рисунке 10.

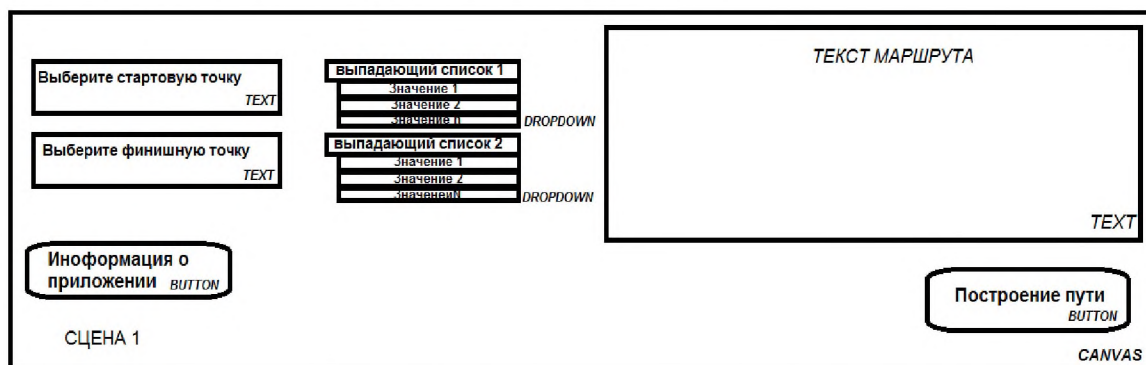


Рисунок 10 – Макет первой сцены

Первая сцена включает в себя три объекта Text, два из которых представляют справку о том, что необходимо выбрать из DropDown, третий объект Text представляет построенный маршрут.

Объекты DropDown предоставляют перечень всех возможных точек, таких как: лестницы, лифты, переходы между корпусами, номера аудиторий и кабинетов.

Первая сцена включает в себя два объекта Button. Первый объект Button – кнопка «Информация о приложении», открывающая вторую сцену, которая предоставляет информацию о приложении. Второй объект Button – кнопка «Построение пути», запускающая скрипт по построению пути из стартовой в финишную точку, если построение такого пути возможно, то он будет представлен на третьем объекте Text, в противном случае пользователь получить сообщение о невозможности построения пути.

На рисунке 11 представлен макет второй сцены.

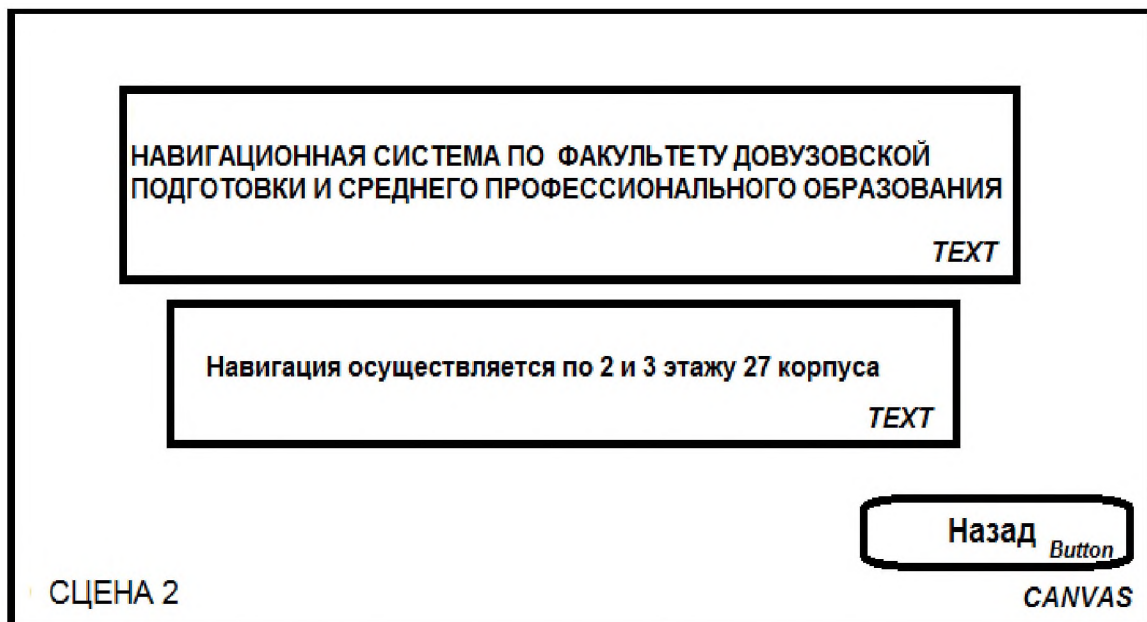


Рисунок 11 – Макет второй сцены

Вторая сцена включает в себя два объекта Text и один объект Button. Первый объект Text хранит информацию о приложении. Второй объект Text хранит информацию об ограничении навигации. Объект Button – кнопка «Назад», которая открывает первую сцену.

2.4 Программирование дискретного рабочего поля

Дискретным рабочим полем в рамках алгоритма волновой трассировки считается двумерное клетчатое поле или матрица. Планы этажей структурных подразделений университета возможно представить в виде дискретного рабочего поля, для этого необходимо масштабировать план структурного подразделения университета таким образом, что каждый квадратный метр площади этажа – это одна ячейка дискретного рабочего поля.

Для представления дискретного рабочего поля в программе был создан класс DRP с тремя полями:

А) первое поле – номер кабинета или аудитории, в случае лестниц, лифтов и т.п. их значением будет являться константа – 255;

Б) второе поле – название кабинета или аудитории;

В) третье поле необходимо для реализации алгоритма волновой трассировки, если клетка относится к проходимому подмножеству, то она принимает значение -1, если клетка относится к непроходимому подмножеству, то она принимает значение -2.

```
public class DRP
{
    private int _Nomer;
    private string _Name;
    private int _Prohod;
}
```

Для обеспечения безопасности и целостности данных каждое из полей имеет тип доступа private, что не позволит обратиться непосредственно внутрь класса к самому полю. Для инициализации значения поля или получения значения необходимо обратиться к свойству set или get.

```
public int Nomer
{
    get { return _Nomer; }
    set { _Nomer = value; }
}

public string Name
{
    get { return _Name; }
    set { _Name = value; }
}

public int Prohod
{
    get { return _Prohod; }
}
```

```
set { _Prohod = value; }  
}
```

После масштабирования плана структурного подразделения получен необходимый размер дискретного рабочего поля – восемь строк и четырнадцать столбцов.

```
DRP[,] drp = new DRP[N, M];
```

N и M – константы, равные восемь и четырнадцать.

Полный листинг программного кода представлен в приложении Б.

2.5 Разработка методов

Всего разработано три метода и обработано три события для реализации алгоритма волновой трассировки.

Первый разработанный метод `InsertStartConcern(NumberFloor)`.

```
private DRP[,] InsertStartConcern(int NumberFloor)  
{  
    DRP[,] d = new DRP[N, M];  
    for(int i = 0; i < N; i++)  
    {  
        for(int j = 0; j < M; j++)  
        {  
            d[i, j] = new DRP();  
            d[i, j].Name = "noname";  
            d[i, j].Nomer = NEPROHOD;  
            d[i, j].Prohod = NEPROHOD;  
            if(i == 4)  
            {  
                d[i, j].Name = "prohod";  
            }  
        }  
    }  
}
```

```

        d[i, j].Nomer = PROHOD;
        d[i, j].Prohod = PROHOD;
    }
    if(NumberFloor == 2)
    {
        if(j == 10)
        {
            d[i, j].Name = "prohod";
            d[i, j].Nomer = PROHOD;
            d[i, j].Prohod = PROHOD;
        }
    }
}
return d;
}

```

Метод определен модификатором доступа private для обеспечения безопасности и целостности данных [1].

Метод возвращает массив типа DRP – пользовательский класс дискретного рабочего поля. На основе анализа планов этажей двадцать седьмого корпуса выявлена закономерность – коридор всегда располагается в четвёртой строке, независимо от этажа. Отсюда следует, что все ячейки матрицы в четвертой строке относятся к проходимому подмножеству, все остальные относятся к непроходимому подмножеству.

Если навигация осуществляется на втором этаже, то к проходимому подмножеству добавляются все ячейки десятого столбца, так как это переход из первого корпуса в двадцать седьмой и из двадцать седьмого корпуса в шестнадцатый. В методе это предусмотрено проверкой на номер этажа.

После того как дискретное рабочее поле заполнено, необходимо перейти ко второму этапу алгоритма – распространение волны. Для этого разработан метод FindWave(drp, Start, Finish).

```

private void FindWave(DRP[,] drp, DRP_PointSaF start,
DRP_PointSaF finish)
{
    bool add = true;
    int step = 0;
    drp[finish.X, finish.Y].Prohod = 0;
    drp[start.X, start.Y].Prohod = PROHOD;
    while(add == true)
    {
        add = false;
        for(int i = 0; i < N; i++)
        {
            for(int j = 0; j < M; j++)
            {
                if(drp[i,j].Prohod == step)
                {
                    if (i - 1 >= 0 && drp[i - 1, j].Prohod != NEPROHOD
&& drp[i - 1, j].Prohod == PROHOD)
                        drp[i - 1, j].Prohod = step + 1;
                    if (j - 1 >= 0 && drp[i, j - 1].Prohod != NEPROHOD
&& drp[i, j - 1].Prohod == PROHOD)
                        drp[i, j - 1].Prohod = step + 1;
                    if (i + 1 < N && drp[i + 1, j].Prohod != NEPROHOD
&& drp[i + 1, j].Prohod == PROHOD)
                        drp[i + 1, j].Prohod = step + 1;
                    if (j + 1 < M && drp[i, j + 1].Prohod != NEPROHOD
&& drp[i, j + 1].Prohod == PROHOD)

```

```

                drp[i, j + 1].Prohod = step + 1;
            }
        }
    }
    step++;
    add = true;
    if(drp[start.X, start.Y].Prohod != PROHOD)
        add = false;
    if (step > N * M)
        add = false;
}
}

```

Метод полностью реализует второй этап алгоритма волновой трассировки – этап распространения волны.

Третий этап алгоритма волновой трассировки – восстановление пути, он реализован группой методов:

А) *OutputResultPath(DRP_TwoFloor, StartPoint, FinishPoint)* – метод, который убирает все ложные пути, появившиеся на этапе распространения волны из дискретного рабочего поля;

Б) *SearchMaxValueDrp(drp, start)* – метод, который находит максимальное значение во всем дискретном рабочем поле, так как на этапе анализа плана двадцать седьмого корпуса было определено, что одна клетка дискретного рабочего поля равна одному квадратному метру, следовательно, максимальный элемент дискретного рабочего поля – метраж построенного пути;

В) *OutputResultTextPath(start, finish, SearchMaxValueDrp(drp, start))* – метод, формирующий окончательный текст маршрута.

```

private void OutputResultPath(DRP[,] drp, DRP_PointSaF start,
DRP_PointSaF finish)
{

```

```

    for (int i = 0; i < N; i++)
    {
        for(int j = 0; j < M; j++)
        {
            if(drp[i,j].Prohod != PROHOD || drp[i, j].Prohod !=
NEPROHOD)
            {
                if (start.Y > finish.Y && (j > start.Y || j < finish.Y)) drp[i,
j].Prohod = NEPROHOD;
                if (finish.Y > start.Y && (j < start.Y || j > finish.Y)) drp[i,
j].Prohod = NEPROHOD;
            }
        }
    }

    OutputResultTextPath(start, finish, SearchMaxValueDrp(drp,
start));
}
private void OutputResultTextPath(DRP_PointSaF start,
DRP_PointSaF finish, int value)
{
    if (start.Y > finish.Y) ResultPath += "Пройдите направо от "
+ TextValueDropdownStart + " ";
    if (finish.Y > start.Y) ResultPath += "Пройдите налево от" +
TextValueDropdownStart + " ";
    ResultPath += value.ToString() + " метров";
    ResultPath += " до " + TextValueDropdownFinish + ".";
    TextPath.text = ResultPath;
}
private int SearchMaxValueDrp(DRP[,] drp, DRP_PointSaF start)

```

```

    {
        int max = drp[0, 0].Prohod;
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < M; j++)
            {
                if(drp[i, j].Prohod != PROHOD && drp[i, j].Prohod !=
NEPROHOD && drp[i,j].Prohod != drp[start.X, start.Y].Prohod)
                {
                    if (drp[i, j].Prohod > max) max = drp[i, j].Prohod;
                }
            }
        }
        return max;
    }

```

Для функционирования приложения обработано три события:

- А) нажатие на кнопку «построение пути»;
- Б) выбор элемента из первого dropdown;
- В) выбор элемента из второго dropdown.

Обработка события нажатия на кнопку создает массив класса DRP, после чего поэтапно вызывает все методы. При обработке события происходит проверка на выбор одинаковых точек в dropdown и, если не выбрана хотя бы одна из точек.

Обработка выбора элемента подразумевает определение стартовой или финишной точек, в зависимости от того в каком dropdown осуществлен выбор.

Для обработки события выбора элемента из dropdown, разработана группа методов с использованием делегатов. Делегат – класс, позволяющий хранить ссылку на метод с определенной сигнатурой.

А) `DropDownStartValueChanged(DropdownStart)` – метод, определяющий значение, выбранное значение в `dropdownStart`;

Б) `DropDownFinishValueChanged(DropdownFinish)` – метод, определяющий выбранное значение в `dropdownFinish`;

В) `OprPoint(ValueDropDown)` – метод, позволяющий определить координаты точки в соответствии с дискретным рабочим полем.

```
void DropdownStartValueChanged(Dropdown change)
{
    ValueDropDownStart = change.value;
    TextDropDownStart = change.captionText.text;
    DropdownFinish.onValueChanged.AddListener(delegate{
        DropdownFinishValueChanged(DropdownFinish);
    });
}
void DropdownFinishValueChanged(Dropdown change)
{
    ValueDropDownFinish = change.value;
    TextDropDownFinish = change.captionText.text;
    DRPMenu();
}
private DRP_PointSaF OprPoint(int DropdownValue)
{
    DRP_PointSaF p = new DRP_PointSaF();
    switch (DropdownValue)
    {
        case 1:
            {
                p.X = 5;
                p.Y = 7;
            }
    }
}
```


break;

```
case 2:{p.X = 3;p.Y = 6;}break;case 3:{p.X = 5;p.Y = 2;}break;case 4:{p.X = 3;p.Y = 3;}break;case 5:{p.X = 5;p.Y = 7;}break;case 6:{p.X = 3;p.Y = 0;}break;case 7:{p.X = 5;p.Y = 7;}break;case 8:{p.X = 3;p.Y = 10;}break;case 9:{p.X = 5;p.Y = 7;}break;case 10:{p.X = 5;p.Y = 11;}break;case 11:{p.X = 5;p.Y = 13;}break;case 12:{p.X = 5;p.Y = 11;}break;case 13:{p.X = 5;p.Y = 13;}break;case 14:{p.X = 5;p.Y = 4;}break;case 15:{p.X = 3;p.Y = 12;}break;case 16:{p.X = 5;p.Y = 4;}break;}return p;}
```

Полный листинг программы представлен в приложении Б.

					ДП – 02069964 – 090203 – 02 – 18	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

3 Тестирование программного модуля

3.1 Результаты тестирования

При тестировании программного продукта выявлены следующие ситуации, приводящие к ошибкам в работоспособности:

- А) выбор одинаковых точек для старта и финиша пути;
- Б) не выбор одной из точек или обеих.

В первом случае пользователь получит сообщение, представленное на рисунке 12.

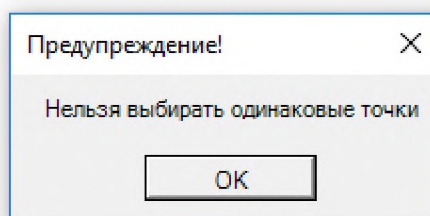


Рисунок 12 – Сообщение о выборе одинаковых точек

Если же пользователь не выберет точку для старта или финиша, то получит соответствующее сообщение.

На рисунке 13 представлено сообщения, если не выбрана стартовая точка, на рисунке 14 представлено сообщение, если не выбрана финишная точка.

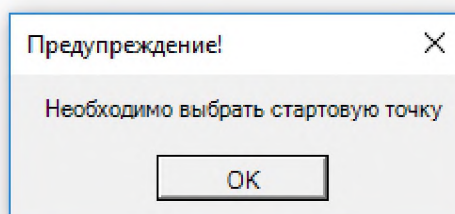


Рисунок 13 – Сообщение о не выборе стартовой точки

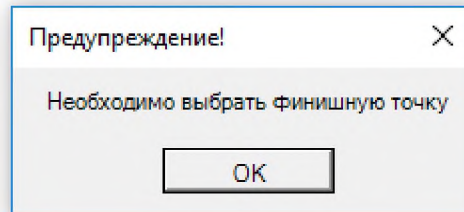


Рисунок 14 – Сообщение о не выборе финишной точки

ЗАКЛЮЧЕНИЕ

В ходе работы над дипломным проектом был разработан программный модуль для навигационной системы прикладной программы.

Модуль разрабатывался на языке программирования C#, интерфейс разрабатывался средствами межплатформенной среды разработки Unity3D.

В соответствии со всеми задачами проведены работы: проанализированы технический паспорт двадцать седьмого корпуса, выбран оптимальный метод построения пути – алгоритм волновой трассировки. На основе анализа технического паспорта двадцать седьмого корпуса разработано дискретное рабочее поле, запрограммированы все этапы алгоритма волновой трассировки.

Проанализированы технологии реализации пользовательских интерфейсов. Создан пользовательский интерфейс средствами Unity3D.

Реализовано программирование дискретного рабочего поля в виде отдельного класса.

В результате получен программный модуль навигационной системы, позволяющий строить путь от одной аудитории или кабинета до другой.

Цель дипломного проекта достигнута полностью.

					ДП – 02069964 – 090203 – 02 – 18	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 Джон Скит. С# для профессионалов: тонкости программирования, 3-е издание, новый перевод — М.: «Вильямс», 2014. – 608 с.

2 Джон Хокинг. Unity в действии. Мультиплатформенная разработка на С#, 1-е издание. – Питер СПб, 2018. – 336 с.

3 Исмаил М. И. А. Navigation inside buildings (using WI-FI points)/ М. И. А. Исмаил, В. В. Никулин // XLIV Огаревские чтения материалы научной конференции: в 3 частях. Отв. за вып. П. В. Сенин. 2016. С. 327-331.

4 Исмаил М. И. А. О разработке приложения для навигации внутри зданий по WI-FI-точкам / Исмаил М. И. А., Никулин В. В. // XLV Огарёвские чтения Материалы научной конференции. В 3-х частях. Отв. за вып. П. В. Сенин. 2017. С. 614-618.

5 Козадаев А. С. Реализация волнового алгоритма для определения кратчайшего маршрута на плоскости при моделировании трасс с препятствиями [Текст] / А. С Козадаев, Е. В. Дубовицкий // Вестник Тамбовского университета. Серия: Естественные и технические науки. – Тамбов, 2010. – С. 1926-1931.

6 Махмуд И. А. И. Навигация внутри зданий (по точкам WI-FI) // И.А.И. Махмуд, В. В. Никулин // Проблемы передачи информации в инфокоммуникационных системах: сб. докладов и тезисов VII Всероссийской научно-практической конференции. Министерство образования и науки РФ; Волгоградский государственный университет. 2016. С. 71-75.

ПРИЛОЖЕНИЕ А

(обязательное)

Технический паспорт двадцать седьмого корпуса

Филиал федерального государственного
унитарного предприятия
"Ростехинвентаризация
- Федеральное БТИ" по Республике Мордовия
Саранское отделение

ТЕХНИЧЕСКИЙ ПАСПОРТ

на часть здания учебного
корпуса №27

по улице Б.Хмельницкого дом № 39а

город (пос.) Саранск

район Ленинский

Инвентарный № 16645

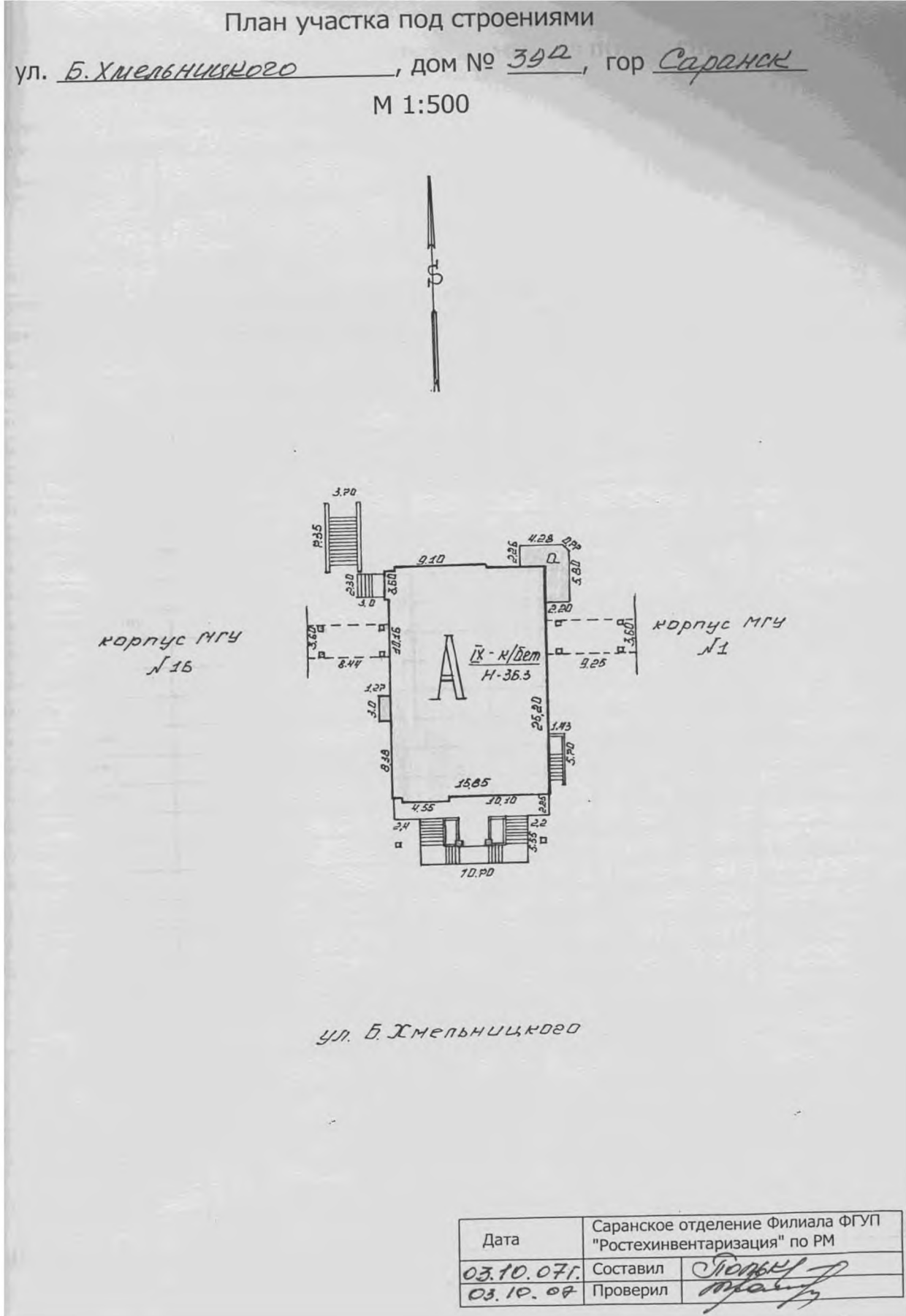
Изм.	Лист	№ докум.	Подпись	Дата

ДП – 02069964 – 090203 – 02 – 18

Лист

38

Продолжение ПРИЛОЖЕНИЯ А



Изм.	Лист	№ докум.	Подпись	Дата	

ДП – 02069964 – 090203 – 02 – 18

Лист

40

Продолжение ПРИЛОЖЕНИЯ А

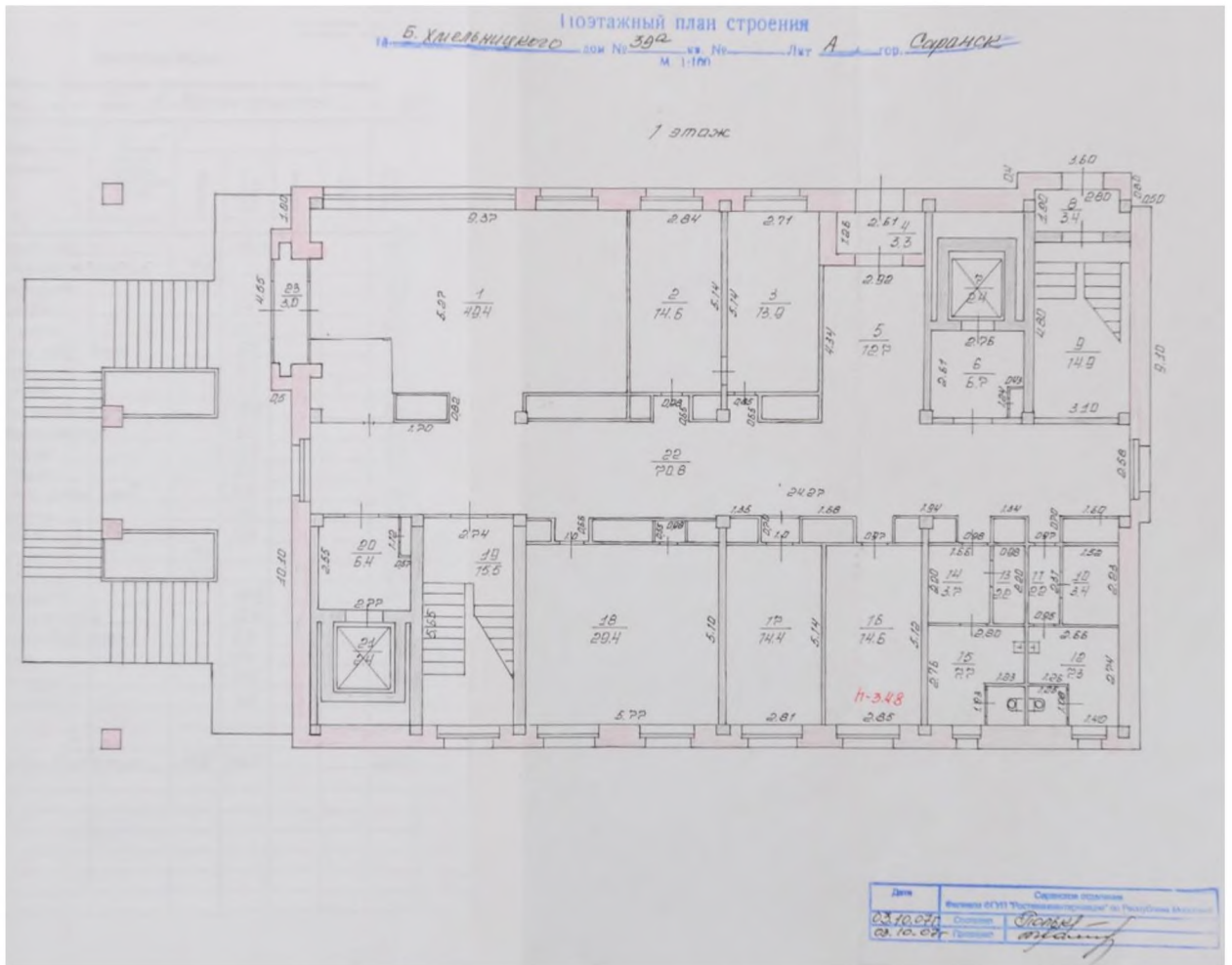
ОПИСАНИЕ КОНСТРУКТИВНЫХ ЭЛЕМЕНТОВ ПОМЕЩЕНИЯ И ОПРЕДЕЛЕНИЕ ИЗНОСА						
Литера <i>А</i>		Год постройки 2001		Этаж 1по5		
Группа капитальности <i>I</i>		Вид внутренней отделки простая				
Наименование конструктивных элементов	Материал и конструкция	Общая площадь кв.м	Ст-ть кв. м.	Восст-ая стоимость в ценах 1969г.	% износа	Действит-ная стоимость в ценах 1969г.
Фундамент	<i>Бетонные блоки</i>	1570,1	158,75	249258	7	231810
Варужн. стены	<i>Керамзито-бетон.,кирп.</i>	Остаточ	балан.	стоимость		21 459 329,48
Перекрытие	<i>Железобетонные плиты</i>					
Крыша	<i>Мягкая кровля</i>					
Полы	<i>Плитка,мрамор,паркет</i>					
Окна	<i>Двойные створные</i>					
Двери	<i>Филенчатые,простые</i>					
Вн. отделка	<i>Шт-ка,окр-ка,обои</i>					
Нар. отделка	<i>Облицовка камнем</i>					
Центр. отопление	+					
Печное отопление						
Водопровод	+					
Электроосвещение	+					
Телефон	+					
Телевидение						
Ванные						
с газов. колонк.						
с дров. колонк.						
с горяч. водосн.						
Горячее вод-ие						
Газоснабжение						
Мусоропровод	+					
Лифты	+					
Канализация	+					
Примечание:						
<< 03 >>	10	2007г.	Исполнил	<i>Столбел</i>	<< >>	
<< 03 >>	10	2007г.	Проверил	<i>Морозов</i>	<< >>	
Работа выполнена	<< >>	200 г	<< >>	200 г	<< >>	200 г
ИСПОЛНИЛ						
ПРОВЕРИЛ						



Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ДП - 02069964 - 090203 - 02 - 18

Продолжение ПРИЛОЖЕНИЯ А



Изм.	Лист	№ докум.	Подпись
			Дата

ДП – 02069964 – 090203 – 02 – 18

Лист

42

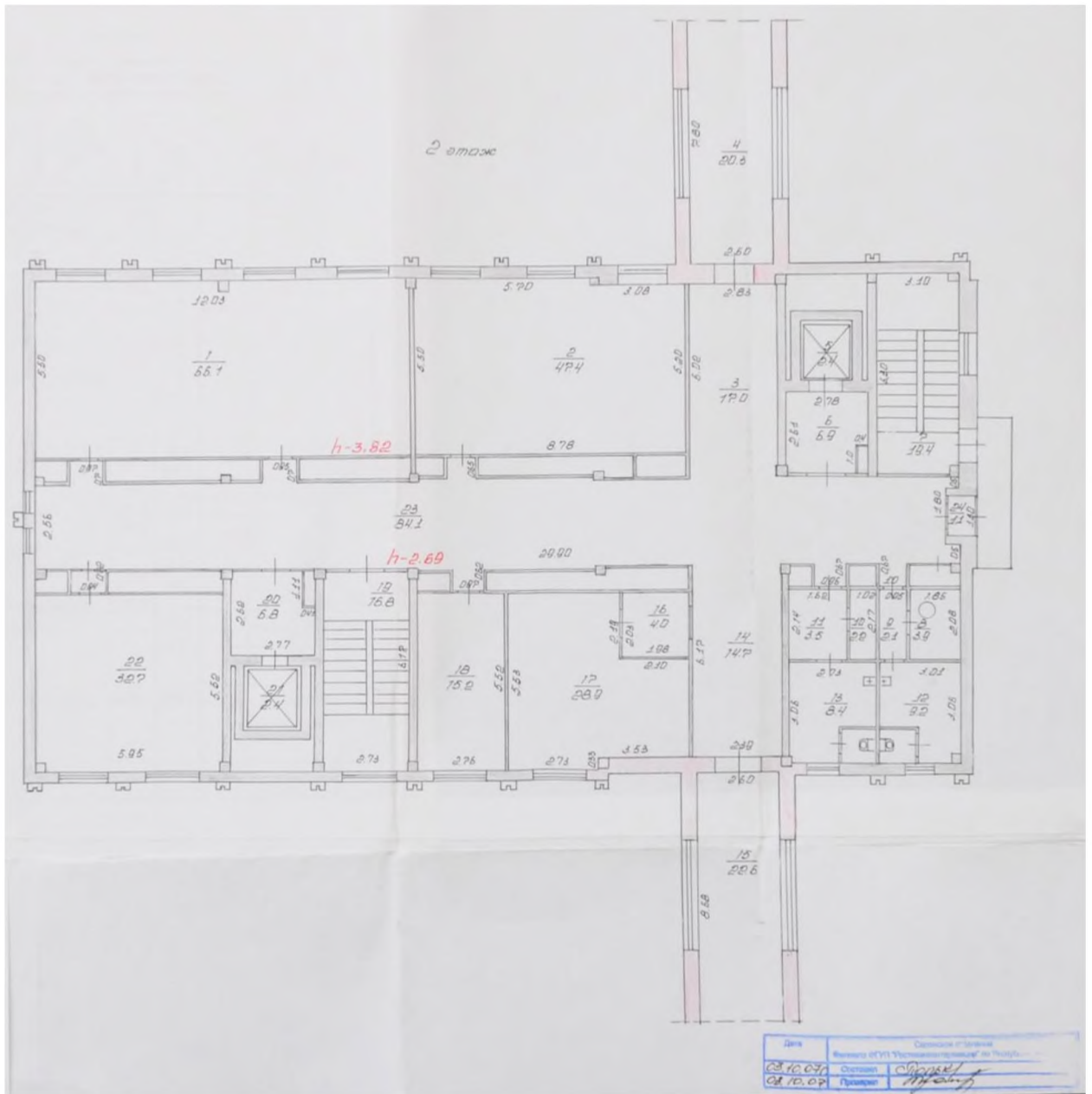
Экспликация

к поэтажному плану строения расположенного в городе (поселке)

Саранск по ул. (пер.) Б. Хмельницкого № 39 а

литера по плану	Этаж	№ помещения по плану	Назначение частей помещения	Формула подсчета площади по внутреннему обмеру					Итого по строению, этажу	Высота
					основная	вспомогательная	основная	вспомогательная		
2	3	4	5	7	8	6	10	11		
А	I	1	вестибюль			49.4			49.4	3.48
		2	сигнализационная		14.6				14.6	"
		3	гардероб		13.9				13.9	"
		4	тамбур			3.3			3.3	"
		5	коридор			12.7			12.7	"
		6	лифтовой КОМ			6.7			6.7	"
		7	лифт			2.4			2.4	"
		8	тамбур			3.4			3.4	"
		9	лест. клетка			14.9			14.9	"
		10	музороповод			3.4			3.4	"
		11	коридор			2.2			2.2	"
		12	ванна			2.3			2.3	"
		13	помещ. убор. ЛНБ.			2.2			2.2	"
		14	коридор			3.7			3.7	"
		15	ванна			2.7			2.7	"
		16	междункт			14.6			14.6	"
		17	кабинет			14.4			14.4	"
		18	гардероб				29.4		29.4	"
		19	лест. клетка				15.5		15.5	"
		20	лифтовой КОМ				6.4		6.4	"
		21	лифт				2.4		2.4	"
		22	коридор				70.8		70.8	"
		23	тамбур				3.0		3.0	"
			Итого по I этажу:		57.5	246.8		304.3		

Продолжение ПРИЛОЖЕНИЯ А



Изм.	Лист	№ докум.	Подпись	Дата

ДП - 02069964 - 090203 - 02 - 18

Лист

44

Продолжение ПРИЛОЖЕНИЯ А

Приложение № 1

На нежилые строения имеющие
встроенные нежилые помещения.

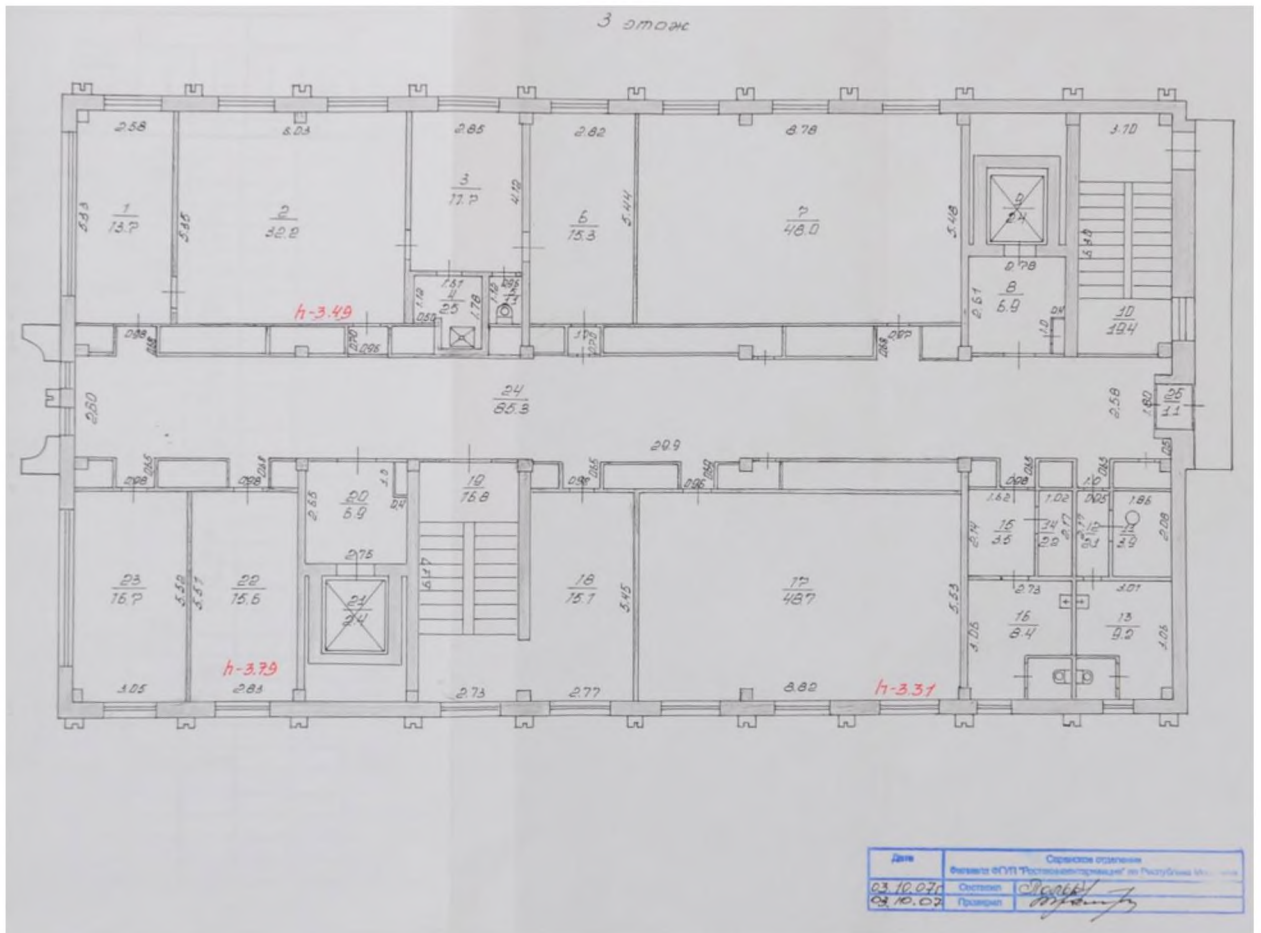
Экспликация

к поэтажному плану строения расположенного в городе (поселке)

Саранске по ул. (пер.) Б. Жемельницкого № 39а

Литера по плану	Этаж	№ помещения по плану	Назначение частей помещения	Формула подсчета площади по внутреннему обмеру	Основная		Вспомогательная		Итого по строению, этажу	Высота	
					основная	вспомогательная	основная	вспомогательная			
2	3	4	5	6	7	8	9	10	11		
А	II	1	УИТАЛЬНИ ЗАЛ		66.1				66.1	3.82	
		2	КНИГОХРАНИЛИЩЕ		47.4				47.4	"	
		3	КОРИДОР			17.0			17.0	2.69	
		4	ПЕРЕХОД			20.3			20.3	"	
		5	ЛИФТ			2.4			2.4	"	
		6	ЛИФТОВОЙ КОЛЛ			6.9			6.9	"	
		7	ЛЕСТ. КЛЕТКА			19.4			19.4	3.82	
		8	МУСОРОПРОВОД			3.9			3.9	"	
		9	КОРИДОР			2.1			2.1	"	
		10	ПОМЕЩ. ЧУБОВ. ЛИНВЕНТ.			2.2			2.2	"	
		11	КОРИДОР			3.5			3.5	"	
		12	САМУДЕЛ			9.2			9.2	"	
		13	САМУДЕЛ			8.4			8.4	"	
		14	КОРИДОР			14.7			14.7	2.69	
		15	ПЕРЕХОД			22.6			22.6	2.69	
		16	КАССА			4.0			4.0	3.82	
		17	БУХГАЛТЕРСЯ			28.9			28.9	"	
		18	КАБИНЕТ			15.2			15.2	"	
		19	ЛЕСТ. КЛЕТКА			16.8			16.8	"	
		20	ЛИФТОВОЙ КОЛЛ			6.8			6.8	"	
		21	ЛИФТ			2.4			2.4	"	
		22	УИТАЛЬНИ ЗАЛ			32.7			32.7	3.82	
		23	КОРИДОР				84.1			84.1	2.69
		24	ТАМБУР				1.1			1.1	"
Итого по этажу:					194.3	243.8		438.1			

Продолжение ПРИЛОЖЕНИЯ А



Изм.	Лист	№ докум.	Подпись
			Дата

ДП – 02069964 – 090203 – 02 – 18

Лист

46

Продолжение ПРИЛОЖЕНИЯ А

Приложение № 1

На жилые строения имеющие
встроенные нежилые помещения.

Экспликация

к поэтажному плану строения расположенного в городе (поселке)

Варанске по ул. (пер.) Б. Железничного № 39а

Этаж по плану	Этаж	№ помещения по плану	Назначение частей помещения	Формула подсчета площади по внутреннему обмеру	площадь		Итого по строению, этажу	Высота	
					основная	вспомогательная			
2	3	4	5	6	7	8	9	10	11
II	1	приемная		13,7			13,7	3,40	
	2	кабинет оидент		32,2			32,2		
	3	комната сторожа		11,7			11,7		
	4	душ. ванна			2,5		2,5		
	5	ванна			1,1		1,1		
	6	библиотека		15,3			15,3		
	7	прилив, элитальн. зал		48,0			48,0		
	8	лифтовой холл			6,9		6,9		
	9	лифт			2,4		2,4		
	10	лест. клетка			19,4		19,4		
	11	мусоропровод			3,9		3,9		
	12	коридор			2,1		2,1		
	13	ванна			9,2		9,2		
	14	помещ. убор. инвент.			2,2		2,2		
	15	коридор			3,5		3,5		
	16	ванна			8,4		8,4		
	17	зал заседаний		48,7			48,7	3,31	
	18	коридор			15,1		15,1	3,79	
	19	лест. клетка			16,8		16,8		
	20	лифтовой холл			6,9		6,9		
	21	лифт			2,4		2,4		
	22	кабинет		15,6			15,6		
	23	кабинет		16,7			16,7		
	24	коридор			85,3		85,3		
	25	тамбур			1,1		1,1		
		Итого по этажу:		201,9	189,2		391,1		

Продолжение ПРИЛОЖЕНИЯ А

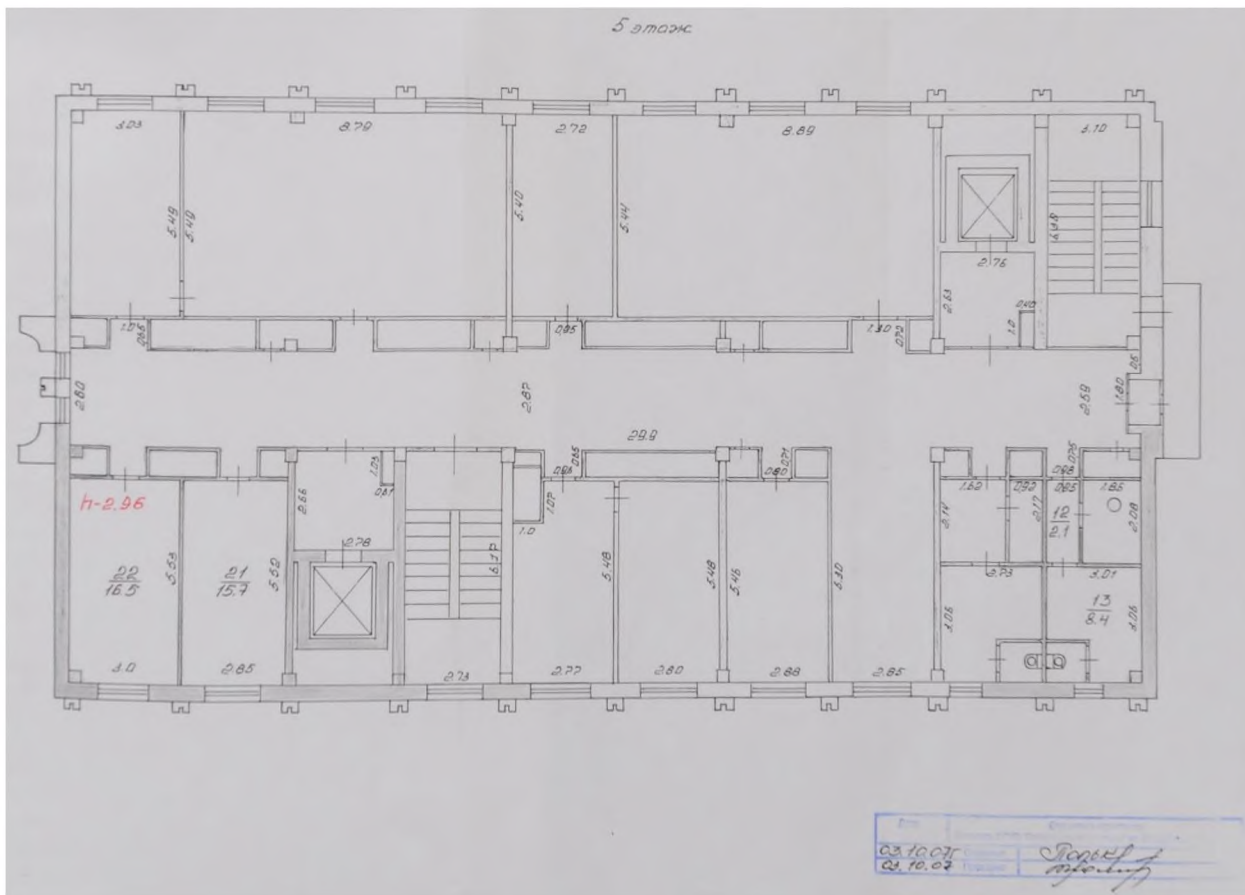
Приложение № 1
На нежилые строения помещений
встроенные нежилые помещения.

ЭКСПЛИКАЦИЯ

к поэтажному плану строения расположенного в городе (населенке) Баранки по ул. (пер.) Б. Земельничкова № 59 Д

1	2	3	4	5	6	7	8	9	10	11
литера по плану	Этаж	№ помещения по плану	Назначение частей помещения	Формула подсчета площади (по внутреннему обмеру)	основная	вспомогательная	основная	вспомогательная	Итого по строению, этажу	Высота
IV	1	1	лабораторная		16.4				16.4	2.93
		2	кадраномерная		48.4				48.4	4
		3	препараточная		15.0				15.0	4
		4	дуговая		47.8				47.8	4
		5	лифтовой код			6.9			6.9	4
		6	лифт			2.4			2.4	4
		7	мет. клемма			1.04			1.04	4
		8	лифтовой код			3.9			3.9	4
		9	коридор			2.1			2.1	4
		10	ванна			9.2			9.2	4
		11	помещение фоб. линд.			2.2			2.2	4
		12	коридор			3.5			3.5	4
		13	ванна			8.4			8.4	4
		14	коридор			17.9			17.9	4
		15	кабинет		15.8				15.8	4
		16	кабинет		15.4				15.4	4
		17	кабинет		15.0				15.0	4
		18	мет. клемма			16.8			16.8	4
		19	лифтовой код			6.9			6.9	4
		20	лифт			2.4			2.4	4
		21	кабинет		15.9				15.9	4
		22	кабинет		16.3				16.3	4
		23	коридор			85.3			85.3	4
		24	мандур			1.1			1.1	4
			лифт по 4 этажу:		205.5	188.4			393.9	

Продолжение ПРИЛОЖЕНИЯ А



Изм.	Лист	№ докум.	Подпись	Дата

ДП – 02069964 – 090203 – 02 – 18

Лист

50

Экспликация

к поэтажному плану строения расположенного в городе (поселке)

Саранске по ул. (вар.) *Б. Хмельницкого* № *392*

Литера по плану	Этаж	№ помещения по плану	Назначение частей помещения	Формула подсчета площади по внутреннему обмеру	основная		вспомогательная		Итого по строению, этажу	Высота
					основная	вспомогательная	основная	вспомогательная		
1	2	3	4	5	6	7	8	9	10	11
А	V	21	кабинет		15.7				15.7	2.96
		22	кабинет		16.5				16.5	
		12	коридор			2.1			2.1	
		13	санузлы			8.4			8.4	
			Итого:			32.2	10.5			
			Всего по части здания:		691.4	878.7			1570.1	

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг запрограммированного дискретного рабочего поля

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;

public class OutputText : MonoBehaviour {
    public Text TextPath;
    public Dropdown DropdownStart;
    public Dropdown DropdownFinish;

    private const int N = 8;
    private const int M = 14;
    private const int NEPROHOD = -2;
    private const int PROHOD = -1;

    private DRP_PointSaF StartPoint = new DRP_PointSaF();
    private DRP_PointSaF FinishPoint = new DRP_PointSaF();
    int ValueDropdownStart;
    int ValueDropdownFinish;

    string TextDropdownStart;
    string TextDropdownFinish;

    string ResultPath;
```

									Лист
									52
Изм.	Лист	№ докум.	Подпись	Дата	ДП – 02069964 – 090203 – 02 – 18				

Продолжение ПРИЛОЖЕНИЯ Б

```
void Start () {
    //все этапы тут
    if (MessError())
    {
        DropdownStart.onValueChanged.AddListener(delegate {
            DropdownStartValueChanged(DropdownStart);
        });
    }
}

public bool MessError()
{
    if(ValueDropdownStart == 0)
    {
        EditorUtility.DisplayDialog("Предупреждение!", "Необходимо выбрать
стартовую точку", "Ок");// заголовок, сообщение, ок, отмена
        return false;
    }
    if(ValueDropdownFinish == 0)
    {
        EditorUtility.DisplayDialog("Предупреждение!", "Необходимо выбрать
финишную точку", "Ок");
        return false;
    }
    if(ValueDropdownFinish == ValueDropdownStart)
    {
        EditorUtility.DisplayDialog("Предупреждение!", "Нельзя выбирать
одинаковые точки", "Ок");
        return false;
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    return true;
}
void DropdownStartValueChanged(Dropdown change)
{
    ValueDropdownStart = change.value;
    TextDropdownStart = change.captionText.text;
    DropdownFinish.onValueChanged.AddListener(delegate{
        DropdownFinishValueChanged(DropdownFinish);
    });
}
void DropdownFinishValueChanged(Dropdown change)
{
    ValueDropdownFinish = change.value;
    TextDropdownFinish = change.captionText.text;
    DRPMenu();
}
private DRP[,] InsertStartConcern(int NumberFloor)
{
    DRP[,] d = new DRP[N, M];
    for(int i = 0; i < N; i++)
    {
        for(int j = 0; j < M; j++)
        {
            d[i, j] = new DRP();
            d[i, j].Name = "noname";
            d[i, j].Nomer = NEPROHOD;
            d[i, j].Prohod = NEPROHOD;
            if(i == 4)
            {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        d[i, j].Name = "prohod";
        d[i, j].Nomer = PROHOD;
        d[i, j].Prohod = PROHOD;
    }
    if(NumberFloor == 2)
    {
        if(j == 10)
        {
            d[i, j].Name = "prohod";
            d[i, j].Nomer = PROHOD;
            d[i, j].Prohod = PROHOD;
        }
    }
}
return d;
}
```

```
private void OutputDRP(DRP[,] d)
{
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < M; j++)
        {
            TextPath.text += d[i, j].Prohod;
        }
        TextPath.text += "\n";
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
private DRP_PointSaF OprPoint(int DropdownValue)
{
    DRP_PointSaF p = new DRP_PointSaF();
    switch (DropdownValue)
    {
        case 1:
            {
                p.X = 5;
                p.Y = 7;
            }
            break;
        case 2:
            {
                p.X = 3;
                p.Y = 6;
            }
            break;
        case 3:
            {
                p.X = 5;
                p.Y = 2;
            }
            break;
        case 4:
            {
                p.X = 3;
                p.Y = 3;
            }
            break;
    }
}
```


Продолжение ПРИЛОЖЕНИЯ Б

case 5:

```
{  
    p.X = 5;  
    p.Y = 7;  
}  
break;
```

case 6:

```
{  
    p.X = 3;  
    p.Y = 0;  
}  
break;
```

case 7:

```
{  
    p.X = 5;  
    p.Y = 7;  
}  
break;
```

case 8:

```
{  
    p.X = 3;  
    p.Y = 10;  
}  
break;
```

case 9:

```
{  
    p.X = 5;  
    p.Y = 7;  
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
break;
case 10:
{
    p.X = 5;
    p.Y = 11;
}
break;
case 11:
{
    p.X = 5;
    p.Y = 13;
}
break;
case 12:
{
    p.X = 5;
    p.Y = 11;
}
break;
case 13:
{
    p.X = 5;
    p.Y = 13;
}
break;
case 14:
{
    p.X = 5;
    p.Y = 4;
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    }
    break;
case 15:
    {
        p.X = 3;
        p.Y = 12;
    }
    break;
case 16:
    {
        p.X = 5;
        p.Y = 4;
    }
    break;
}
return p;
}

private int OprNumberFloor(string TextValue)
{
    int number;
    try
    {
        number = Convert.ToInt32(TextValue);
        number /= 100;
    }
    catch
    {
```

Продолжение ПРИЛОЖЕНИЯ Б

```

        if (TextView.Equals("Мужской туалет 2 этаж") ||
        TextView.Equals("Женский туалет 2 этаж"))
            return 2;
        if (TextView.Equals("Мужской туалет 3 этаж") ||
        TextView.Equals("Женский туалет 3 этаж"))
            return 3;
        if (TextView.Equals("Лестница"))
            return -3;
        if (TextView.Equals("Лифт 1") || TextView.Equals("Лифт 2"))
            return -4;
        else
            return -5;
    }
    return number;
}

private int ChangeFloor(int NumberFloorStartPoint, int
NumberFloorFinishPoint, DRP[,] DrpTwoFloor, DRP[,] DrpThirdFloor,
DRP_PointSaF StartPoint, DRP_PointSaF FinishPoint)
{
    DRP[,] drp = new DRP[N, M];
    int number;
    if (NumberFloorStartPoint != NumberFloorFinishPoint)
    {
        ResultPath = "пройдите до лестницы, ";
        StartPoint.X = 5;
        StartPoint.Y = 4;
    }
    if (NumberFloorFinishPoint == 2)
    {

```

Продолжение ПРИЛОЖЕНИЯ Б

```
FindWave(DrpTwoFloor, StartPoint, FinishPoint);
    number = 2;
}
else
{
    FindWave(DrpTwoFloor, StartPoint, FinishPoint);
    number = 3;
}
//проверка, что переход происходит по одному этажу
//если нет, то в ResultPat += "пройдите до лестницы или лифта"
//после сделать стартом лестницу или лифт нужного этажа и строить
путь от неё
//если по одному этажу, то обычный поиск
//после работы этой функции в парсинг ответа, другая функция
return number;
}
private void FindWave(DRP[,] drp, DRP_PointSaF start, DRP_PointSaF finish)
{
    bool add = true;
    int step = 0;
    drp[finish.X, finish.Y].Prohod = 0;
    drp[start.X, start.Y].Prohod = PROHOD;
    while(add == true)
    {
        add = false;

        for(int i = 0; i < N; i++)
        {
            for(int j = 0; j < M; j++)
```

Продолжение ПРИЛОЖЕНИЯ Б

```
{
    if(drp[i,j].Prohod == step)
    {
        if (i - 1 >= 0 && drp[i - 1, j].Prohod != NEPROHOD && drp[i - 1,
j].Prohod == PROHOD)
            drp[i - 1, j].Prohod = step + 1;
        if (j - 1 >= 0 && drp[i, j - 1].Prohod != NEPROHOD && drp[i, j -
1].Prohod == PROHOD)
            drp[i, j - 1].Prohod = step + 1;
        if (i + 1 < N && drp[i + 1, j].Prohod != NEPROHOD && drp[i + 1,
j].Prohod == PROHOD)
            drp[i + 1, j].Prohod = step + 1;
        if (j + 1 < M && drp[i, j + 1].Prohod != NEPROHOD && drp[i, j +
1].Prohod == PROHOD)
            drp[i, j + 1].Prohod = step + 1;
    }
}
step++;
add = true;
if(drp[start.X, start.Y].Prohod != PROHOD)
    add = false;
if (step > N * M)
    add = false;
}
}
private void ScissoringDrp(DRP[,] drp, int LeftBorder, int RighthBorder)
{
    for (int i = 0; i < N; i++)
```

Продолжение ПРИЛОЖЕНИЯ Б

```
{
    for (int j = 0; j < M; j++)
    {
        if (LeftBorder > RighthBorder)
        {
            if ((i > LeftBorder) && (i < RighthBorder))
            {
                drp[i, j].Prohod = NEPROHOD;
            }
        }
        else
        {
            if((i > LeftBorder))
            {
                drp[i, j].Prohod = NEPROHOD;
            }
        }
    }
}

private void SummMetr(DRP[,] drp)
{
    int metr = 0;
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < M; j++)
        {
            if(drp[i,j] != PROHOD && drp[i,j] != NEPROHOD && drp[i,j] > metr)
            {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
                metr = drp[i,j];
            }
        }
    }
    ResultPath += metr.ToString() + " метров";
}

private void OutputResultPath(DRP[,] drp, DRP_PointSaF start, DRP_PointSaF
finish)
{
    ScissoringDrp(drp, start.X, finish.X);
    SummMetr(drp);
    TextPath.text = ResultPath;
}

public void DRPMenu()
{
    DRP[,] DRP_TwoFloor;
    DRP[,] DRP_ThirdFloor;
    DRP_TwoFloor = new DRP[N, M];
    DRP_ThirdFloor = new DRP[N, M];
    DRP_TwoFloor = InsertStartConcern(2);
    DRP_ThirdFloor = InsertStartConcern(3);
    StartPoint = OprPoint(ValueDropdownStart);
    FinishPoint = OprPoint(ValueDropdownFinish);
    int NumberFloorStartPoint = OprNumberFloor(TextDropdownStart);
    int NumberFloorFinishPoint = OprNumberFloor(TextDropdownFinish);
    int WorkDrpFloor = ChangeFloor(NumberFloorStartPoint,
NumberFloorFinishPoint, DRP_TwoFloor, DRP_ThirdFloor, StartPoint ,
FinishPoint);
    if(WorkDrpFloor == 2)
```


Продолжение ПРИЛОЖЕНИЯ Б

```
{
    OutputResultPath(DRP_TwoFloor, StartPoint, FinishPoint);
}
else
{
    OutputResultPath(DRP_ThirdFloor, StartPoint, FinishPoint);
}
}
}
public class DRP_PointSaF
{
    private int _X;
    private int _Y;

    public int X
    {
        get { return _X; }
        set { _X = value; }
    }
    public int Y
    {
        get { return _Y; }
        set { _Y = value; }
    }
}

public class DRP
{
    private int _Nomer;
```

Продолжение ПРИЛОЖЕНИЯ Б

```
private string _Name;
private int _Prohod;

public int Nomer
{
    get { return _Nomer; }
    set { _Nomer = value; }
}

public string Name
{
    get { return _Name; }
    set { _Name = value; }
}

public int Prohod
{
    get { return _Prohod; }
    set { _Prohod = value; }
}
}

public class QuitApp : MonoBehaviour {

    // Use this for initialization
    void Start () {
    }

    // Update is called once per frame
    void Update () {
```

Окончание ПРИЛОЖЕНИЯ Б

```
        if (Input.GetKey("escape"))
            Application.Quit();

    }
}

public class LoadSecondScence : MonoBehaviour {
    public void ChangeScene(string SceneName)
    {
        SceneManager.LoadScene(SceneName);
    }
}

public class ChangeCanvas : MonoBehaviour {
    public Canvas CanvasChangePoint;
    private GameObject CanvasSecond;
    public void DisplayCanvas(bool sw)
    {
        CanvasChangePoint.gameObject.SetActive(sw);
    }
    public void ClearText(Text text)
    {
        text.text = "";
    }
}
```

РЕЦЕНЗИЯ

на выпускную квалификационную работу

Архипова Кирилла Андреевича

Студента группы 409 специальности

«Программирование в компьютерных системах»

на тему: «Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва)»

Проект посвящен актуальной теме разработки программного модуля навигационной системы прикладной программы на примере МГУ им. Н. П. Огарёва, поскольку в МГУ им. Н. П. Огарёва существует проблема с ориентацией в структуре здания университета и структурных подразделений.

Дипломный проект содержит три главы. В первой главе анализируются поставленные задачи дипломного проекта. Во второй главе описано создание программного модуля, с анализом и написанием алгоритма, описана разработка пользовательского интерфейса, программирование дискретного рабочего поля. В третьей главе представлены результаты тестирования программного модуля.

Считаю, что к положительным моментам работы относятся большое количество поясняющих рисунков, подробное описание процесса проектирования и интерфейса пользователя. Удобные в работе окна ввода и вывода данных организованы таким образом, что позволяют легко и быстро освоить предлагаемую программу.

В качестве недостатка можно отметить отсутствие возможности работы разработанной программы во всех структурных подразделениях университета в виду того, что анализ структуры всех структурных подразделений не был произведен, так как объем работ требует больше времени.

Несмотря на указанное замечание, считаю, что выпускная квалификационная работа Архипова К.А. соответствует требованиям, предъявляемым к работам студентов, обучающихся по специальности

«Программирование в компьютерных системах», может быть допущена к защите и заслуживает отличной оценки.

Рецензент: канд. тех. наук,
доцент кафедры ИКТ и СС
ФГБОУ ВО «НИ МГУ им. Н.П.Огарёва»



С. Д. Шибайкин



Федеральное государственное автономное
образовательное учреждение высшего образования
**«Волгоградский
государственный университет»**
*Кафедра телекоммуникационных
систем*

05.06.2018 г. № б/н

Справка

Выдана Архипову Кириллу Андреевичу, Никулину Владимиру Валерьевичу, Макарову Вячеславу Эдуардовичу в том, что их доклад заслушан на конференции, а статья «О реализации алгоритма волновой трассировки» принята к опубликованию в сборнике материалов IX Всероссийской научно-практической конференции «Проблемы передачи информации в инфокоммуникационных системах», ФГАОУ ВО «Волгоградский государственный университет», г. Волгоград, 25 мая 2018 г.

Зав. кафедрой ТКС

Е.С. Семенов

ОТЧЕТ

о результатах проверки работы обучающегося

на наличие заимствований

Ф.И.О. автора работы Архипов Кирилл Андреевич

Тема работы Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва)

Руководитель Никулин Владимир Валерьевич

Представленная работа прошла проверку на наличие заимствований в системе «Антиплагиат.ВУЗ»

Результат автоматической проверки: оригинальность 95,90%

цитирования 0,56%

заимствования 3,54%

Результат анализа полного отчета на наличие заимствований:

правоверные заимствования: 3,54%

да/нет, количество (%), обоснованность

корректные цитирования: 0,56%

да/нет, количество (%), обоснованность

неправомерные заимствования: нет

да/нет, количество (%), обоснованность

признаки обхода системы: нет

(да/нет, описание)

Общее заключение об итоговой оригинальности работы и возможности ее допуска к защите: допустить к защите.

Руководитель: канд. тех. наук,
зав. каф. ИКТ и СС
ФГБОУ ВО «НИ МГУ им. Н.П.Огарёва»




В.В.Никулин



СПРАВКА

о результатах проверки текстового документа на наличие заимствований

Проверка выполнена в системе
Антиплагиат.ВУЗ

Автор работы	Архипов Кирилл Андреевич
Факультет, кафедра, номер группы	ФДП и СПО, 409 гр
Тип работы	Выпускная квалификационная работа
Название работы	Архипов ТЕКСТ
Название файла	Архипов ТЕКСТ.docx
Процент заимствования	3,54%
Процент цитирования	0,56%
Процент оригинальности	95,90%
Дата проверки	13:10:51 05 июня 2018г.
Модули поиска	Модуль выделения библиографических записей; Сводная коллекция ЭБС; Коллекция РГБ; Цитирования; Переводные заимствования; Коллекция Гарант; Модуль поиска Интернет; Модуль поиска "МГУ им. Н. П. Огарева"; Модуль поиска общеупотребительных выражений; Кольцо Вузов
Работу проверил	НИКУЛИН ВЛАДИМИР ВАЛЕРЬЕВИЧ ФИО проверяющего
Дата подписи	 Подпись проверяющего

Чтобы убедиться
в подлинности справки,
используйте QR-код, который
содержит ссылку на отчет.



Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.

ОТЗЫВ РУКОВОДИТЕЛЯ ДИПЛОМНОГО ПРОЕКТА

Тема работы «Разработка программного модуля навигационной системы прикладной программы в системе разработки приложений «Unity3D» (на примере МГУ им. Н. П. Огарёва)».

Автор (студент) Архипов Кирилл Андреевич

Факультет довузовской подготовки и среднего профессионального образования.

Выпускающая предметная цикловая комиссия (кафедра) общепрофессиональных специальных (информационно-коммуникационных) дисциплин.

Специальность 09.02.03 Программирование в компьютерных системах.

Руководитель В. В. Никулин, канд. тех. наук, заведующий кафедры ИКТ и СС ФГБОУ ВО «НИ МГУ им. Н.П.Огарёва».

Оценка соответствия требованиям ФГОС СПО подготовленности автора выпускной работы

Требования к профессиональной подготовке	Соответствует	В основном соответствует	Не соответствует
Уметь корректно формулировать и ставить задачи (проблемы) своей деятельности при выполнении дипломной работы, анализировать, диагностировать причины появления проблем, их актуальность	*		
Устанавливать приоритеты и методы решения поставленных задач (проблем)	*		
Уметь использовать информацию – правильно оценить и обобщить степень изученности объекта исследования (в соответствии со специализацией)	*		
Владеть методами сбора, хранения и обработки (редактирования) информации, применяемой в сфере профессиональной деятельности (по специализации)	*		
Владеть современными методами анализа и интерпретации полученной информации, оценивать их возможности при решении поставленных задач (по специализации)	*		
Уметь рационально планировать время выполнения работы, определять грамотную последовательность и объем операций и решений при выполнении поставленной задачи	*		
Уметь объективно оценивать полученные результаты расчетов, вычислений, используя для сравнения данные других направлений	*		
Уметь анализировать полученные результаты интерпретации данных (в рамках специализации)	*		
Знать методы системного анализа	*		
Уметь осуществлять деятельность в кооперации с коллегами, находить компромиссы при совместной деятельности	*		
Уметь делать самостоятельные обоснованные и достоверные выводы из проделанной работы	*		

Уметь пользоваться научной литературой профессиональной направленности	*		
--	---	--	--

Отмеченные достоинства: Рассматриваемые в дипломном проекте вопросы актуальны в связи с тем, что существует проблема в навигации для круга лиц: студенты первокурсники или студенты незнакомые со структурой университета, гости университета, абитуриенты, сталкиваются с проблемой по нахождению нужной аудитории или кабинета, что приводит к потере времени и сил. В первой главе анализируются поставленные задачи дипломного проекта. Во второй главе описано создание программного модуля, с анализом и написанием алгоритма, описана разработка пользовательского интерфейса, программирование дискретного рабочего поля. В третьей главе представлены результаты тестирования программного модуля.



Выполненный дипломный проект представляет собой приложение реализованный с удобным и интуитивно понятным графическим интерфейсом пользователя, полностью готовый к применению. В процессе реализации данного проекта проведен анализ предметной области, осуществлен сбор необходимой информации; обеспечено корректное отображение построенного пути; произведено тестирование полученной системы.

Одним из преимуществ данного приложения является его понятный интерфейс. Это позволяет пользователю без труда находить нужную информацию.

Отмеченные недостатки:

Существенных недостатков в работе не выявлено.

Заключение: оценивая в целом дипломный проект Архипова Кирилла Андреевича, следует отметить, что она отвечает основным требованиям и может быть допущена к защите. Оригинальность текста составляет 95,90%. Автор заслуживает присвоения квалификации техник-программист.

Руководитель  « 25 »  2018 г.
(подпись)