

Федеральное государственное бюджетное образовательное
учреждение высшего образования
Санкт–Петербургский государственный университет

Кочедыкова Екатерина Михайловна

Выпускная квалификационная работа
*Разработка приложения для навигации внутри
помещений в среде Unity*

Уровень образования: бакалавриат
Направление 50.03.01 «Искусства и гуманитарные науки»
Основная образовательная программа «Свободные искусства и науки»

Научный руководитель:
к.физ.-мат.н., доцент
Черных Герман Анатольевич

Рецензент:
СПбГУ,
д.ф.-м.н., профессор
Пастон Сергей Александрович

Санкт-Петербург
2021 г.

Содержание

Введение	3
0.1. Актуальность	3
0.2. Цели и задачи	5
Теоретическая часть	7
Глава 1. Обзор литературы	7
1.1. Навигация внутри помещения	7
1.1.1 Интерактивная карта	8
1.1.2 Позиционирование внутри помещения	10
1.1.3 Поиск пути	12
Глава 2. Мобильные приложения и среда разработки Unity .	14
2.1. Сцена	17
2.2. Объекты и иерархия	17
2.3. Пользовательский интерфейс	18
2.4. Unity и XCode	18
Практическая часть	20
Глава 3. Разработка прототипа мобильного приложения . . .	20
Обсуждения	26
Список литературы	27

Введение

Современные реалии требуют содержания информации на цифровом носителе. Компании и организации стараются задействовать интеграцию собственных продуктов и услуг в мобильные устройства и различные Web-сервисы. Подтверждением проникновения Интернета, коммуникационных и информационных технологий в любые сферы социального взаимодействия, в том числе и в создание проектов, посвященных навигации внутри объекта, могут служить, как наличие постоянного спроса на digital-решения в данной области, так и появление мероприятий, связанных с созданием "Умного города".

В данной работе будет рассмотрен процесс создания приложения в интегрированной среде разработки Unity[1]. Итогом работы будет являться прототип приложения, реализующий функции навигации внутри помещения. Второстепенной, но не менее важной целью было добиться как можно более бюджетного варианта разработки, с намерением использовать приложение, не привлекая установку дополнительных устройств по площади помещения.

0.1 Актуальность

Навигация по улицам города при наличии заряженного смартфона и подключения к сети Интернет давно не является проблемой, нет необходимости покупать бумажный путеводитель – повсеместно существует глобальное количество программного обеспечения с сервисом поиска маршрутов – все данные можно получить, используя GNSS (Глобальные навигационные спутниковые системы). В случае с большими открытыми территориями сигнала спутника оказывается достаточно, чтобы определить местоположение мобильного устройства пользователя и проложить маршрут в точку назначения. Проблемы появляются при навигации в закрытом помещении, механизм перестаёт работать из-за отражения и рассеивания спутникового сигнала от стен [2]. Размер объекта также имеет значение: развертка

навигации происходит на небольшой территории, соответственно, расстояние между объектами внутри минимально, и определение местоположения требует более высокой точности. Таким образом, можно заключить о сложности создания решения в сфере indoor-навигации.

Когда пользователь попадает в помещение, его поле зрения ограничено, и нагружено различными объектами вариативной формы и величины, это приводит к легкой потере ориентации в пространстве. Хаотичная нумерация кабинетов, многоэтажность, наличие тупиков, — подобные препятствия возникают при поиске быстрого и оптимального решения задачи поиска маршрута и затрудняют передвижение даже по небольшому зданию, особенно при посещении в первый раз. Интерактивная карта — частое решение для объектов с высоким процентом посещаемости и разнообразным количеством секций. Данное решение используется в торговых центрах, музеях и даже офисах с большим количеством сотрудников[2]. Реализация подобного проекта позволяет не только быстро сориентироваться на местности, сэкономив собственные нервы и время, но также повышает интерес пользователя к объекту: возможность объяснить маршрут до мероприятия или устроить виртуальную экскурсию по дворцу привлечет внимание пользователя, в связи с обеспечением его понятным игровым аспектом приложения.

Одни из популярных продуктов, использующих электронную версию карт здания: "Этажи"[3] и Google Maps[3]. В их числе и одно из известнейших решений – Dubai Mall[4]. Размещение на бумажном носителе крупнейшего торгово-развлекательного центра в мире, насчитывающего несколько миллионов квадратных метров площади, и при этом сохраняющего удобство для использования, представляется невозможной задачей. В этом случае принимается решение о создании интерактивной карты. Подобный способ хранения модели объекта имеет широкий спектр применения: застройщики могут разместить на сайтах информацию о точечных объектах и жилых районах, владельцы торговых центров хранят информацию о торговых точках и их расположении внутри центра, музеи имеют возможность

разместить описание исторического места или памятника культуры.

0.2 Цели и задачи

Одна из первых стадий проектирования приложения требует формулировки четких задач, выполняемых по мере продвижения к финальному этапу разработки. Изначальной целью данного проекта является создание прототипа клиентского приложения кампуса "Факультета свободных искусств и наук", которое должно будет содержать электронную карту факультета и в дальнейшем выполнять такие функции, как:

1. Вывод и получение местоположения пользователя на плане здания
2. Возможность получения местоположения нужной пользователю аудитории
3. Построение оптимального маршрута до нужного пользователю места внутри помещения на плане здания
4. Реализация дополнительных возможностей: дополненная реальность, оптимальный маршрут через открытое пространство (сад, внутренний двор), информация о дворце, расписание, возможные обновления по встречам клубов или конференциям, проводимых внутри факультета и др.

Таким образом, для успешного достижения целей были сформулированы следующие задачи:

1. Изучение синтаксиса языка C# и основных возможностей межплатформенной среды разработки Unity
2. Выбор наиболее оптимального решения определения местоположения пользователя
3. Интеграция плана здания в приложение
4. Реализация алгоритма поиска пути на карте помещения

5. Создание пользовательского интерфейса приложения

В работе будут рассмотрены следующие пункты: В главе 1 будет содержаться обзор статей о применяемых методах навигации внутри помещения, а именно созданию карты, размещения на ней местоположения пользователя и алгоритмах поиска пути. Глава 2 содержит описание подхода разработки кроссплатформенного приложения в среде Unity. Глава 3 посвящена практической компоненте работы и рассказывает о попытках создания приложения для навигации внутри факультета.

Приложение создано с помощью среды разработки Unity – платформа позволяет за относительно короткое время создать прототип приложения, и незамедлительно контролировать результат. Работоспособность приложения проверена на iOS-устройстве, среда также поддерживает экспорт под другие мобильные операционные системы.

Теоретическая часть

Глава 1. Обзор литературы

1.1 Навигация внутри помещения

Несмотря на то, что GPS-навигация распространена повсеместно и актуально ее использование для открытых пространств, затруднительно применять данную технологию в закрытых помещениях. Вопрос о том, какую из технологий выбрать как альтернативу в навигации внутри помещений до сих пор остается открытым и, по мнению многих авторов, до сих пор не существует единого стандарта оптимального позиционирования, к тому же с низкой стоимостью. Рейнер Маутс [2] утверждает, что не существует на 2012 год глобального решения, которое бы удовлетворяло точности в 1 метр и оставалось одновременно бюджетным. В исследовании на 2018-й год авторы [5] все еще не уверены в том, что разработчики пришли к единому варианту state-of-the-art.

Здания – небольшая территория с достаточно маленьким расстоянием между объектами, в связи с этим требует достаточной точности в определении местоположения. Помимо задачи определения координаты пользователя в закрытом пространстве, для GNSS-систем затруднительно оценивать точное нахождение над уровнем моря, что означает наличие погрешности и в отношении этажей здания[2]. Соответственно, рассматриваемая навигация более сложная, нежели навигация на огромной площади открытого пространства.

Ошибочно утверждать, что создание приложения для навигации в помещении имеет кардинальные различия в основных принципах проектирования архитектуры. Изменяется масштаб данных и подходы к реализации, в то же время запросы пользователя остаются смежными: вид на проекцию местности, в которой он находится, легкость и точность в определении собственного местоположения на карте, оценка позиционирования относи-

тельно конечной цели, и визуализация маршрута от точки позиционирования до точки назначения. В данной главе будут рассмотрены подходы к реализации представленных пунктов.



Рис. 1: Алгоритм пользователя

1.1.1 Интерактивная карта

В первую очередь, разработчика интересует вопрос визуализации архитектурного сооружения. В данном случае следует учитывать расположение плана на экране устройства. Возникают проблемы в том, как разместить двухмерные чертежи, учитывать ли многослойность плана, задуматься, стоит ли добавлять трехмерную модель здания. Какой вид будет восприниматься пользователем лучше: изометрический или ортогональный. Важен и пункт о том, как организовать данные, представляющие объект. Подобные вопросы появляются при интеграции карты в мобильное приложение.

Создание графической модели здания довольно сложная задача, также требующая выбора технологии, учитывающей геометрию объекта. Многие из методов сходятся в использовании серверной части приложения для хранения данных об объекте. Авторы статьи [6] предлагают использовать BIM (Building information models), однако упоминают, что большое количество сооружений не имеет в базе данных 3D модели, из чего следует, что информация об их планиметрии не содержится в цифровом варианте, их создание все еще требует кропотливой неавтоматизированной работы.

Можно заключить, что картографирование помещений по-прежнему остается сложной задачей. Изобретаются способы для трехмерного моделирования существующих в реальности зданий, многие из них основаны на методе облака точек (point cloud)[7], то есть базы данных из набора вершин в трехмерной системе координат. Отсюда проистекают методы сбора данных: с использованием компьютерного зрения, фотограмметрии, и других, основанных на трехмерном сканировании местности [6]. В случае с Unity Engine, возможно не использовать метод облака точек, создавая трехмерную модель не привязывая к реально существующим координатам объекта. Например, авторы исследования, посвященного ориентированию в закрытом помещении[8] восстанавливают из чертежей прототип модели университета, используя программный пакет ArchiCAD, основанный на описанном выше методе информационного моделирования.

Следующим за картографическим сбором информации, возникает задача о комплексном моделировании объекта, с целью получить цифровой рельеф в векторной графике, что, как можно заключить, не является полностью автоматическим процессом. В том числе появляются такие проблемы, как зависимость объекта от трафика людей, количества денег, запроса, и возможности организовать проект для трехмерного indoor-моделирования. Авторы[6] предлагают вариант добиться этого способом краудсорсинга, настаивая на объединении данных для последующего широко-распространенного использования. Данные решения актуальны и по сей день, самым известным может считаться проект OpenStreetMap [9].

Существует более простое в реализации, но предоставляющее понятное пользователю статичное отображение информации, а именно визуализация двухмерного плана [10]. Данный подход не обязательно требует подключения и использования данных сервера, в то же время использование базы данных приветствуется, например, для процедуры сопоставления плана здания с картой.

Дополненная реальность также нашла применение в приложениях для навигации. Использование компьютерного зрения и камеры продвинулось дальше считывания маркеров для определения координаты. Метод трехмерного сканирования для создания проектов по дополненной реальности реализуется как с помощью ARCore и ARKit, так и с использованием сред разработки на их основе, например, интегрированные с игровым движком Unity, применяются Placemote SDK [11] и Vuforia Engine [12]. В 2020 году разработчиками Vuforia презентовано приложение Vuforia Area Target Creator [13], предоставляющее возможность сбора пространственной информации для создания трехмерной модели, повторяющей отсканированное пространство.

1.1.2 Позиционирование внутри помещения

Позиционирование – определение положения объекта или пользователя, используется в частности для своевременного сообщения о перемещении объекта поиска в другую локацию [2]. В настоящее время нет стандарта, монополизирующего хранение информации о местоположении пользователя. Существуют разные методы для отслеживания его координат, огромный прогресс в технологиях определения местоположения внутри помещения случился в течение последних декад, новые решения продолжают быть востребованными и появляться в реальном времени.

Системы навигации должны быть доступны в сложных условиях, включающих возможность отслеживания реального местоположения пользователя внутри помещения, важно также определение его ориентации в пространстве. Данная функция требует интеграции множественных систем позиционирования, чтобы обеспечить легкую и надежную навигацию [6].

Навигация внутри помещения используется в абсолютно различных условиях, в связи с этим существует огромное количество решений, связанных с отслеживанием передвижения пользователя по зданию, помимо

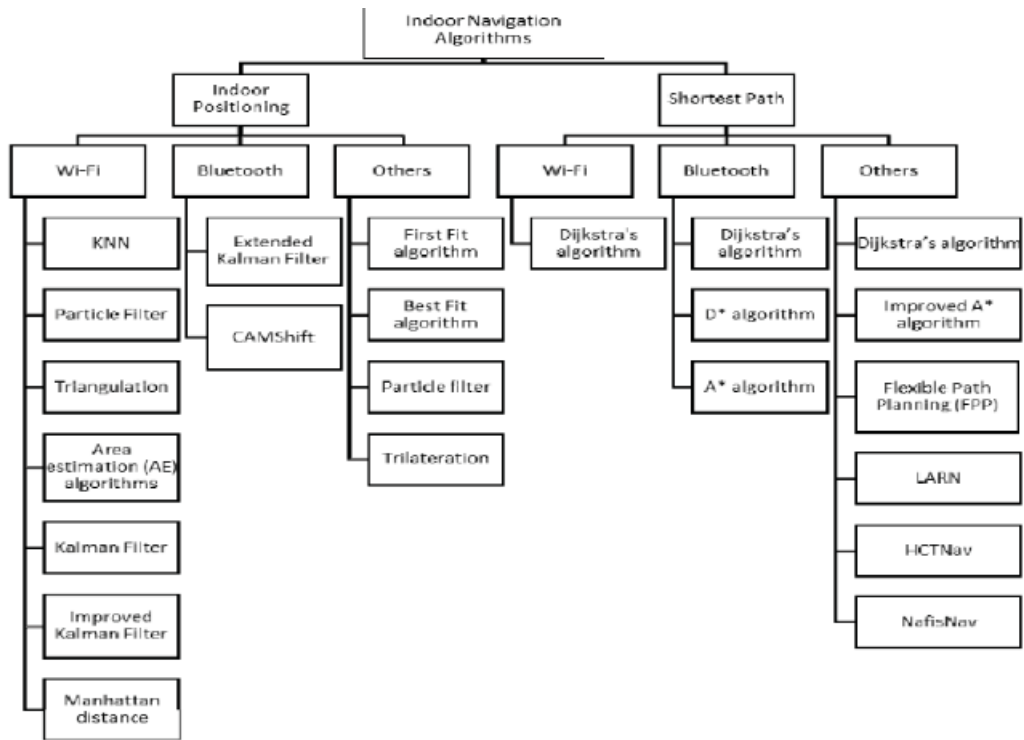


Рис. 2: Существующие алгоритмы навигации и позиционирования [5]

определения точки локации пользователя. Из-за вариативности условий, существует более десятка видов реализации [2], в том числе и беспроводные методы, усиливающие работу GNSS-систем в здании [2], на изображении 2 представлены методы позиционирования и навигации, применяющиеся на 2018 год. В то же время наиболее популярными методами трекинга – отслеживания местоположения пользователя, остаются технологии, использующие Bluetooth, WLAN(беспроводная локальная сеть) и QR-сканирование [5]. Их популярность обусловлена, в том числе низкой стоимостью, а также использованием исключительно смартфона пользователем. Предлагаю остановиться на данных методах и разобрать их подробнее.

Технология WLAN позиционирования или Wi-Fi positioning system основывается на нахождении координат точек беспроводной сети. Главное преимущество метода: архитектура для имплементации в большинстве случаев навигации уже существует [14]. Однако несмотря на распространение Wi-Fi хот-спотов внутри организаций и жилых помещений, реализация точности определения и отслеживания местоположения и переме-

щения пользователя без установки дополнительного оборудования может варьироваться от 1.5 до 2 метров [15], в некоторых случаях точность может достигать 5 метров [16]. В то же время многие пытаются повысить точность сигнала, не прибегая к дополнительной установке точек, и пытаются применять различные алгоритмы, в их числе KNN, fingerprinting или triangulate [16].

Технологии, использующие Bluetooth, имеют название BLE (Bluetooth low energy). Их принцип в установке специальных маячков, называемых beacons, распространяющих радиосигнал. В связи с тем, что точки радиосигнала можно ассоциировать с узлами графа, технология нашла свое применение в системах навигации, основанных на точках пути [17]. Учитывая тот факт, что технология относится к бюджетной, она требует дополнительной установки данных маячков, поэтому в обоих описанных случаях беспроводного трекинга, точность зависит не только от алгоритма, но и от количества установленных устройств [18], что отражается на стоимости производства.

При сканировании визуальных маркеров, используется камера смартфона. Данная технология не требует дополнительных затрат на предустановку устройств. В качестве визуального маркера может использоваться двухмерное изображение, например, QR-код. В случае выбора данного подхода, нужно держать во внимании подвижность пользователя: сами маркеры зависят от освещения, угла сканирования и расстояния между камерой устройства и самим маркером. Сканирование производится технологиями компьютерного зрения, в отличие от безмаркерных технологий затрачивает меньшее количество CPU [19].

1.1.3 Поиск пути

В качестве реализации поиска пути, в навигации используются алгоритмы, основанные на принципах графового поиска. Алгоритм поиска

на графе строит путь, начиная со стартовой точки, и сравнивает узлы со смежными до тех пор, пока не будет найден конечный. Реализация графового поиска в мобильном приложении зависит от задачи. Один из распространенных способов: деление интерактивной карты на тайлы, где узлами выступают прямоугольные фрагменты карты, данный подход также используется, например, в построении маршрута 2GIS[20]. В том случае, если задача требует карты ограниченного пространства, принято использовать путевые точки (waypoints) – узлы графа устанавливаются вручную, сам узел содержит информацию о том, к какому из узлов возможно переместиться.

Навигация пользователя требует выбора алгоритма поиска. Факторы, которые влияют на выбор алгоритма напрямую зависят от графа, имплементированного в приложение. Алгоритм должен удовлетворять следующим условиям: при запуске находится оптимальный путь, он обязан быть кратчайшим до цели. На плане здания могут находиться артефакты, нежелательные пользователем для прохождения: кратчайший путь может проходить через лестницу, лифт или не всегда активные двери. Данный фактор возможно решить, используя взвешенный граф, что значит, установить большие веса для особых ребер. По умолчанию в качестве веса возможно учитывать расстояние между смежными узлами, в этом случае вес каждого из ребер не должен быть отрицательным, что позволяет использовать алгоритм Дейкстры в качестве оптимального решения.

Алгоритм Дейкстры решает проблему поиска кратчайшего пути, используя релаксацию ребер в ориентированном взвешенном графе. В самом алгоритме должно поддерживаться множество ребер S , кратчайшие пути до которых из вершины s уже определены. В общем случае псевдокод будет выглядеть таким образом[21]:

DIJKSTRA(G, w, s)

1. INITIALIZE-SINGLE-SOURCE(G, s)

2. $S \leftarrow$ пустое множество
3. $Q \leftarrow V[G]$
4. while $Q \neq$ пустое множество
5. do $u \leftarrow \text{EXTRACT-MIN}(Q)$
6. $S \leftarrow S \cup u$
7. for each vertex v принадлежащему $\text{Adj}[u]$
8. do $\text{RELAX}(u, v, w)$

Существуют также решения с использованием эвристического алгоритма A^* , в том числе и на двухмерных планах зданий [22], алгоритм является расширенной версией алгоритма Дейкстры. В отличие от последнего, в алгоритме A-Star вводится понятие приблизительной стоимости расстояния до цели.

Поиск пути – динамический процесс, включает в себя сложное человеческое поведение и разнообразные когнитивные навыки. Разработчика интересуют задачи: как понять, какой маршрут оптимален, как выбрать маршрут, контролировать, ориентировать и обозначить точку назначения. Конечная точка должна достигаться пользователем без особых усилий и за короткий промежуток времени[8]. Постоянно должен быть способ верифицировать свое положение в пространстве пользователем, в этом помогают визуальные указания или ориентиры (landmarks) [23].

Глава 2. Мобильные приложения и среда разработки Unity

Одна из задач разработки приложения: понять, какая платформа будет использоваться чаще всего для пользователя. В случае приложения для навигации, появляются решения не только на мобильные устройства,

но и, к примеру, на Google Glasses[24]. Однако существуют относительно бюджетные технологии, которые используются людьми повсеместно, и не только с целью отследить свое положение в пространстве, в том числе смартфоны и другие мобильные устройства остаются глобальной возможностью создать решение, которое будет сразу быстро распространено среди огромного количества пользователей [6].

Разработка мобильного приложения – процесс, включающий в себя множество этапов. Во-первых, требуется выбрать платформу, для которой это приложение будет разрабатываться как приоритет. В связи с этим есть два главных подхода в разработке мобильного приложения – нативный и кроссплатформенный. Нативный подход – разработка на нативном языке платформы, рекомендуемой производителем. К примеру, для Android-устройств рекомендуется Kotlin и Java, для приложений на iOS – SWIFT и Objective-C, а для Windows-Phone нативный подход достигается написанием приложения на C#. Преимущество нативного приложения в максимальной использовании всех возможных функций, что предоставляются операционной системой устройства, как итог, в возможности создания мощного приложения с быстро выполняемым кодом, сложной анимацией и понятным пользователю интерфейсом. Главный недостаток подхода – включает в себя написание отдельного кода для каждой операционной системы, что замедляет время разработки и, соответственно, увеличивает ее стоимость.

Кроссплатформенный подход заключается в написании кода для всех платформ за один раз. Таким образом, основа кроссплатформенного приложения будет одинакова для всех устройств, в то же время разработчики кроссплатформенных фреймворков стараются интегрировать нативные пользовательские интерфейсы в собственную среду для удобства пользования.

Популярные приложения, основанные на методе интерактивной карты, Google Maps и 2GIS, изначально придерживались использования ис-

ключительно смартфона, исключая использование дополнительных устройств пользователем, и применяли дополнение данных изображений снимков со спутника, соотнесением с долготой и широтой и дополнением-схемой дорог. Интерактивные карты оснащены возможностью поиска точки назначения на карте и прокладыванием маршрута. В случае с indoor-навигацией, данные приложения используют кроссплатформенные решения, в частности, WebGL[25].

Приложения, предлагающие решения для навигации внутри помещения, в особенности сфокусированные на создании навигации для одного конкретного объекта, помимо упомянутого выше WebGL, используют среду разработки Unity. Выбор данной среды разработки обусловлен такими пунктами, как:

1. мультиплатформенность, приложения возможно создавать для различных операционных систем
2. низкий порог вхождения, в том числе из-за полной связи игрового движка со средой разработки, благодаря чему возможно тестировать и отлаживать проект внутри редактора.
3. наличие бесплатной лицензии [26]
4. выбор двумерного или трехмерного проекта
5. интуитивное создание дизайна пользовательского интерфейса приложения

В Unity также есть возможность создания веб-приложения с помощью WebGL API, в то же время следует помнить, что код транслируется из C# , официального языка скриптов Unity, в C++, и затем в JavaScript, что может привести к медленной производительности и большому потреблению памяти приложения[27].

Несмотря на изначальную игровую направленность среды, успешны попытки использовать Unity вне игровой деятельности. Поскольку интер-

активное взаимодействие с пользователем интересно вне игровой сферы, случаи использования игрового движка вне игровой разработки получили название геймификации (gamification)[8]. Например, для картографов, архитекторов возможна визуализация данных, создание исследований, поддерживающих интерактив, и, в том числе, визуализация урбанистических планов. В некотором роде, ориентация внутри закрытого пространства становится легче, благодаря развитию интерактивных 3D и изометрических карт[28]. В данной среде разработки используется терминология, которая обусловлена её документацией[29]. Следует рассмотреть основные принципы системы для общего понимания устройства движка.

2.1 Сцена

Сценой в Unity называется среда, на которую разделяется проект. Каждая сцена часто содержит игровой уровень, внутри которого находятся игровые объекты, называемые GameObjects. Вне зависимости от выбора 3D или 2D среды для проекта, Unity разворачивает сцену с объектами в трехмерном пространстве, она загружается, когда запускается приложение.

2.2 Объекты и иерархия

Один из важнейших концептов, выделяющий Unity – каждый объект, находящийся внутри сцены, является игровым объектом. Им может быть как фрагмент карты, так и освещение, стены, указатели, здания, а также невидимые объекты, выполняющие логические операции, такие как GameManager. Получается, что игровой объект может наследовать другой объект, и изменяться в зависимости от родительского объекта. Компонента transform содержит информацию о повороте, координатах местоположения и масштаба каждого игрового объекта. Среда разработки использует DragDrop интерфейс, поэтому скрипты – части кода, написанные для изменения поведения объектов, могут прикрепляться к объектам внутри сцены.

Для каждого из игровых объектов существует локальное и глобальное пространство. В случае отсутствия у игрового объекта родителя, локальное пространство является глобальным само по себе. В случае, если игровой объект является дочерним другого объекта, его изменения будут выполняться в соответствии с ограничениями родительского, в этом случае локальное пространство будет различаться с глобальным.

Игровые объекты относительно удобно изменять и встраивать в сцену, а также изменять их компоненты в инспекторе, однако не всегда удобно вручную настраивать свойства для каждого из объектов. В приложении для навигации у узлов графа должны быть похожие свойства, например, каждый узел должен содержать информацию о количестве связей. В данной ситуации создается префаб(prefab) – особый тип ассетов, который позволяет сохранять значения свойств и компонент игрового объекта. Изменения префаба отражаются на всех префабах, при этом, есть возможность изменять каждый по отдельности.

2.3 Пользовательский интерфейс

В Unity существует возможность создания UI дизайна, она содержится в классе GUI, и представляет интерфейс для настройки графического интерфейса. При создании интерфейса создается элемент Canvas, использующийся для рендеринга экрана, автоматически моделируется обработчик событий EventSystem. Пользовательский интерфейс позволяет использовать функции мобильного приложения внутри игровой среды разработки.

2.4 Unity и XCode

Unity предназначен для мультиплатформенной разработки, поэтому внутри среды возможно указать, на какую платформу ориентирован проект. Данное изменение отслеживается в Настройках во вкладке File>Build

Settings, возможно настроить проект под нужную операционную систему для разработки, включая iOS и Android.

Практическая часть

Глава 3. Разработка прототипа мобильного приложения

В данной главе будут описаны процессы создания приложения по навигации внутри здания факультета "Свободных искусств и наук".

2 этаж

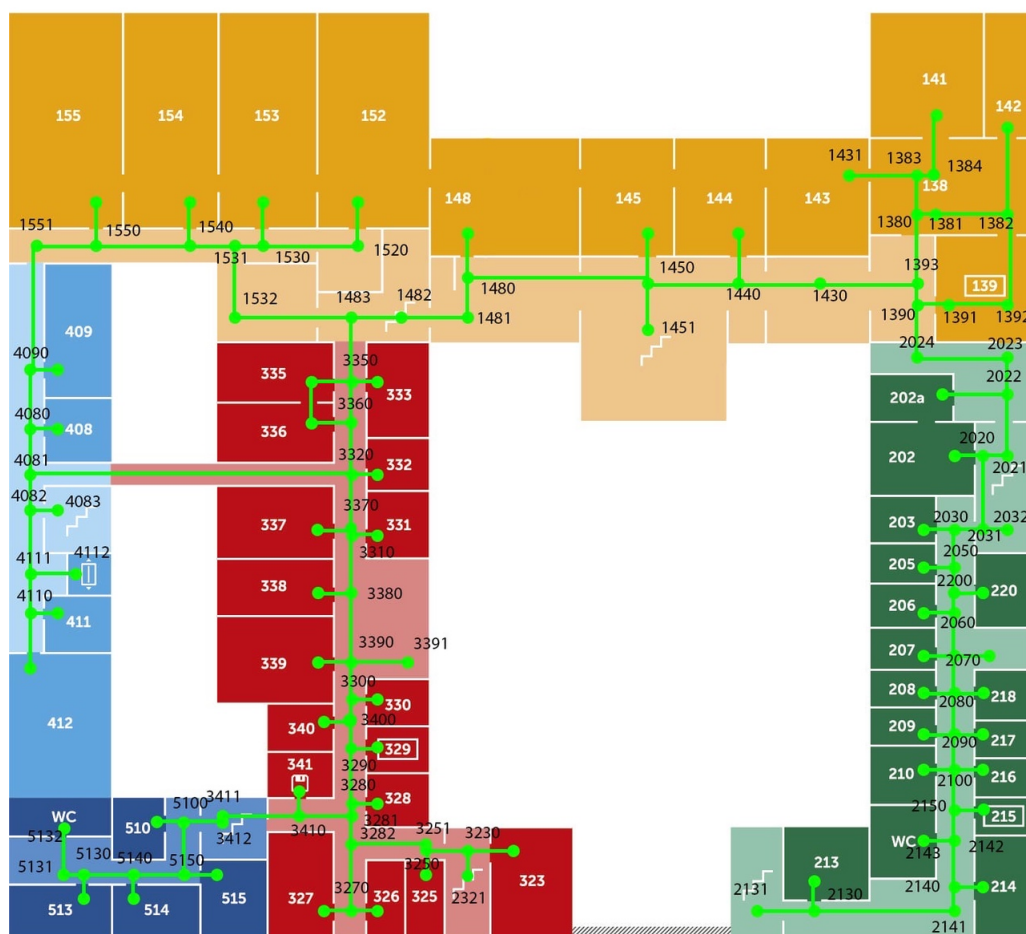


Рис. 3: Проектирование графа для 2 этажа факультета

В первоначальном варианте проекта пользователю не должно требоваться подключение к сети интернет, только предустановленное приложение. В планах позиционирование должно было совершаться при использовании элементов компьютерного зрения: сканирования QR-кода в приложении. Однако благодаря существующей внутри факультета системе

навигации[30], позволяющей человеку довольно быстро сориентироваться и найти номер кабинета вокруг, в этой версии приложения пользователь сам определяет свое местоположение, используя пространственные ориентиры. Для каждого из этажей предусмотрен отдельный граф с доступными перемещениями, данные три графа возможно впоследствии объединить в один для возможности поиска пути между различными этажами.

Далее идет установка Unity и VSCode с официального сайта и создание окружения. Для создания проектов устанавливается лаунчер Unity Hub, внутри которого после авторизации возможно создавать проекты во вкладке Projects. В практической работе использовался Unity версии 2019.4.f1 и XCode версии 12.4. Приложение тестировалось как внутри среды разработки Unity на различных масштабах мобильных устройств, выбираемых во вкладке Game, так и скомпилировано для iOS 14.14.2.

Запуская приложение, важно в меню настроек определить платформу, на которую будет в приоритете разрабатываться приложение и прописать путь к XCode, в случае, если он не прописан по умолчанию. Для того, чтобы проект внезапно не завершился, нужно заполнить поля в меню игрока, указав имя продукта, название компании и Bundle Identifier. Для настройки приложения через XCode нужно подключить мобильное устройство через USB к компьютеру, открыть XCode, на мобильном устройстве в настройках должен появиться режим разработчика.

На начальном этапе еще не было понятно, увенчается ли успехом создание трехмерной или изометрической двухмерной модели здания. Было решено создать 2D проект, поскольку изменить его до трехмерного не представляется сложной задачей. При запуске проекта в отсеке Hierarchy создается автоматически сцена под названием Scene0, внутри которой располагается объект MainCamera, возможно поменять в инспекторе цвет заднего плана. В меню Asset Store устанавливается ассет TextMeshPro для последующей интеграции поля ввода.

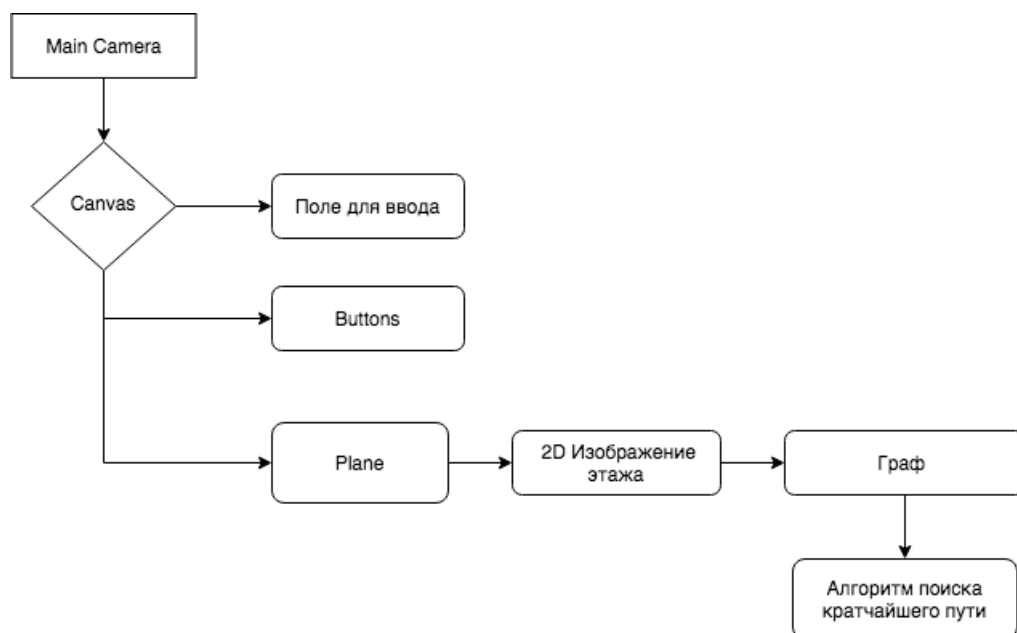


Рис. 4: План иерархии объектов в Unity

Предпринята попытка создания двухмерного интерфейса, результатом стал простой UI-дизайн приложения. При нажатии правой кнопкой мыши выбирается опция UI, позволяющая создавать объекты и трансформировать их по размеру экрана:

1. Объект Canvas, содержащий все элементы пользовательского интерфейса. В нем будут лежать дочерние объекты, описанные ниже
2. Поле ввода (InputField) для возможности ввода номера кабинета пользователем
3. Три кнопки Button, нажимая на каждую из которых, пользователь будет способен выбрать интересующий его план этажа. Внутри объекта Button содержится объект Text с номером этажа
4. Три объекта Panel, цвет фона выбирается прозрачный
5. Каждый из объектов Panel содержит изображение этажа в JPG формате.

Иерархия приложения выглядит согласно изображению 4. Поскольку у кнопок и планов этажа должен быть один стиль, они становятся дочер-

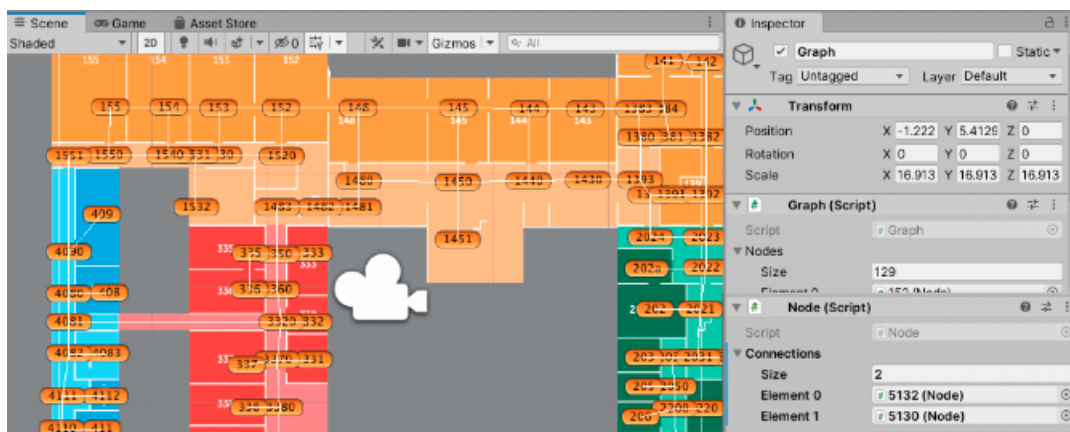


Рис. 5: Граф

ними объектами одной группы. Иерархия позволяет группировать объекты так, чтобы изменения происходили пропорционально другим объектам, что дает возможность автоматически подстраиваться под разрешение экрана.

На следующем этапе был написан скрипт `OpenPanel`, проверяющий, открыто ли пользователем изображение этажа. В случае, если изображения нет на экране, его возможно открыть нажатием на соответствующую кнопку и переключаться между этажами при желании, скрипт прикреплен к каждому объекту `Panel`. По окончании данного этапа получается прототип, в котором возможно посмотреть на пока что не интерактивные карты трех этажей факультета.

Благодаря возможности присваивать объектам дочерние, внутри объекта `Panel`, содержащего изображение этажа возможно создать граф из пустых игровых объектов – точек пути, на котором затем будет реализован алгоритм поиска. Поскольку узел графа должен иметь одинаковые свойства для всех дубликатов, создается префаб с названием `Node`, содержащий информацию о количестве связей и его координаты на карте.

Впоследствии создается количество игровых объектов равное количеству узлов графа, и располагаемое внутри панели, названия узлов соответствуют номерам на рисунке 3, однако остается возможным изменить

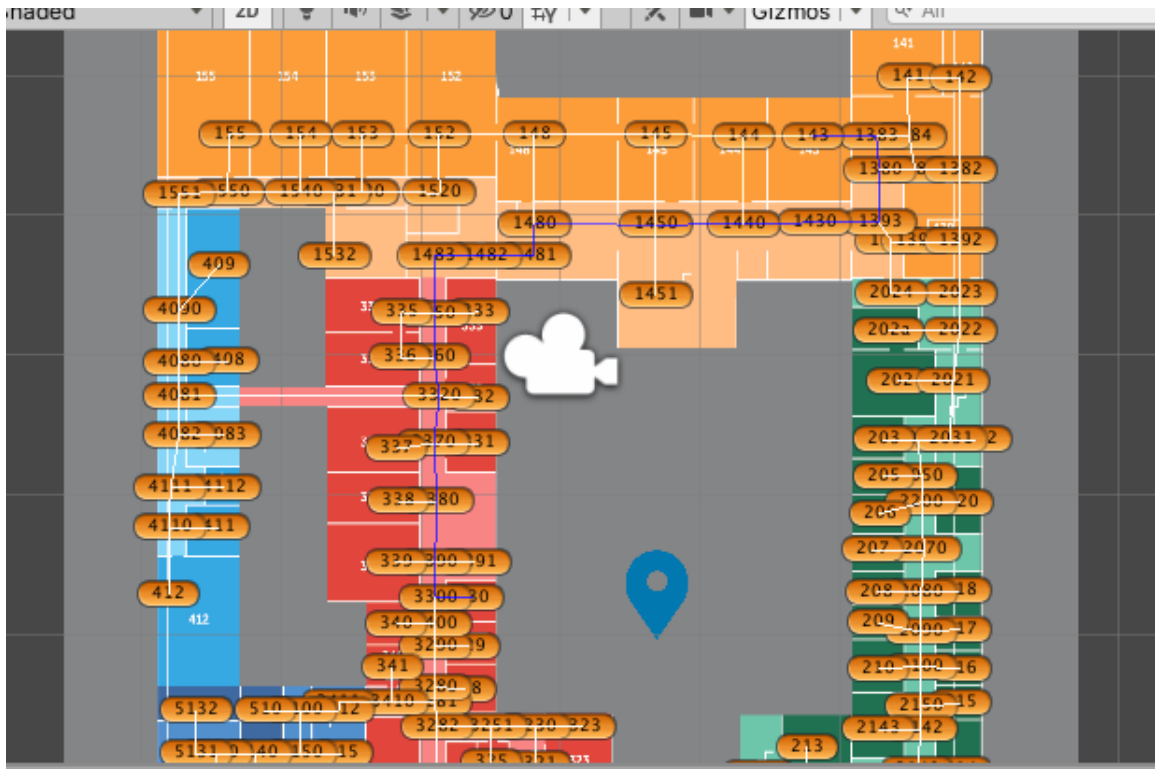


Рис. 6: Реализованный алгоритм поиска

название узла на названия объектов внутри факультета, также узел графа является ориентиром, а значит внедрить информацию о достопримечательностях университета или расписания представляется возможным в будущих версиях приложения. На изображении 5 можно увидеть граф, получившийся в результате, в окне инспектора теперь возможно создавать дочерние объекту Panel игровые объекты, и указывать их связи.

В качестве алгоритма поиска пути был выбран и имплементирован Дейкстра [31], веса – вычисляемое расстояние до другой вершины. На изображении 6 фиолетовым цветом показан путь от 143 аудитории до 330. Этапы создания графа аналогично проделаны с оставшимися двумя этажами факультета.

В завершение, реализован скрипт, позволяющий пользователю ввести аудиторию отправления и назначения и увидеть путь, для корректного отображения маршрута используется компонента LineRenderer. Далее

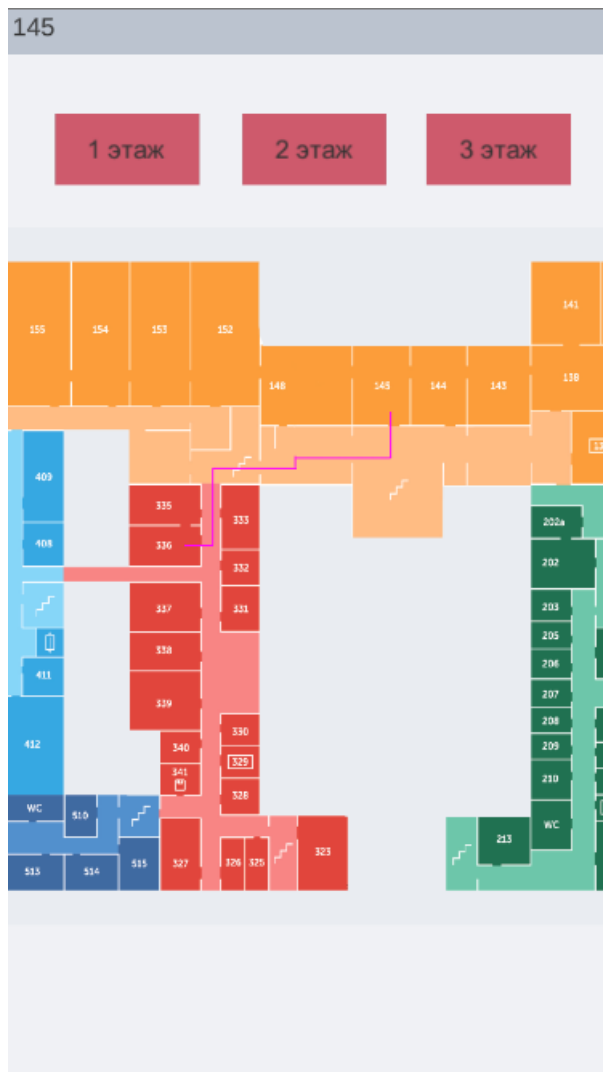


Рис. 7: Finalный вид приложения с поиском пути до 145 кабинета

Приложение компилируется в XCode и возможно протестировать его на мобильном устройстве, finalный код приложения возможно увидеть на GitHub [32].

Обсуждения

Платформа позволила реализовать приложение, предоставляющее пользователю маршрут до точки назначения на интерактивной карте. В результате разработки получилось выполнить основные поставленные задачи.

В последующих версиях, помимо двухмерной карты возможно создать трехмерную модель факультета, также приложение все еще требует проработанного дизайна и более пристального фокуса на удобство пользования. На данном этапе пользователь определяет свое местоположение вручную, в связи с этим следует улучшить позиционирование автоматическим определением точки его локации, упростив пользовательское вмешательство до выбора пункта назначения. Следующим этапом возможно добавление считывания местоположения пользователя по внешнему маркеру, а именно QR-коду для последующего выведения его местоположения и достижения большей интерактивности. В планах, несмотря на отсутствие стабильной эпидемиологической ситуации, добавление ориентиров и поворотных пунктов: следует добавить описания к аудиториям, базы данных приложения для хранения информации об ориентирах, синхронизацию с расписанием, отслеживания заблокированных путей, и т.д.

В дополнение к распознаванию по QR-коду в планах разработать поиск маршрута с помощью AR-технологий: дополненная реальность позволит проводить экскурсии и другие интерактивные мероприятия на площади факультета.

Список литературы

- [1] *Technologies Unity*. Unity, платформа разработки в реальном времени | Платформа для 3D-, 2D-, VR- и AR-контента. — URL: <https://unity.com/ru> (online; accessed: 2021-05-18).
- [2] *Mautz Rainer*. Indoor positioning technologies. — 2012. — P. 1 Band. — Artwork Size: 1 Band Medium: application/pdf Publisher: ETH Zurich. URL: <http://hdl.handle.net/20.500.11850/54888> (online; accessed: 2021-05-01).
- [3] О сервисе - Google Карты. — URL: <https://www.google.com/intl/ru/maps/about/partners/indoormaps/> (online; accessed: 2021-05-19).
- [4] *Group Emaar Malls*. The Dubai Mall | Fashion | Beauty | Lifestyle | Entertainment. — URL: <https://thedubaimall.com/en/map> (online; accessed: 2021-05-18).
- [5] Survey on Algorithms and Techniques for Indoor Navigation Systems. / Ebtesam Alqahtani, Fatimah Alshamrani, Hajra Syed, Fahd Al-Haidari. — 2018. — . — Pp. 1–9.
- [6] Indoor Tracking, Mapping, and Navigation: Algorithms, Technologies, and Applications / Jacky C. K. Chow, Michael Peter, Marco Scaioni, Mohannad Al-Durgham // *Journal of Sensors*. — 2018. — Vol. 2018. — Pp. 1–3. — URL: <https://www.hindawi.com/journals/js/2018/5971752/> (online; accessed: 2021-05-18).
- [7] *Chua Chee Kai, Wong Chee How, Yeong Wai Yee*. Chapter Four - Software and Data Format // Standards, Quality Control, and Measurement Sciences in 3D Printing and Additive Manufacturing / edited by Chee Kai Chua, Chee How Wong, Wai Yee Yeong. — Academic Press, 2017. — . — Pp. 75–94. — URL: <https://www.sciencedirect.com/science/article/pii/B9780128134894000040> (online; accessed: 2021-05-20).

- [8] *De Leeuw Dries, De Maeyer Philippe, De Cock Laure*. A Gamification-Based Approach on Indoor Wayfinding Research // *ISPRS International Journal of Geo-Information*. — 2020. — . — Vol. 9, no. 7. — P. 423. — Number: 7 Publisher: Multidisciplinary Digital Publishing Institute. URL: <https://www.mdpi.com/2220-9964/9/7/423> (online; accessed: 2020-12-15).
- [9] OpenStreetMap. — URL: <https://www.openstreetmap.org/> (online; accessed: 2021-05-20).
- [10] *Nelen Tim, Weyn Maarten, Klepal Martin*. Indoor Navigation for Complex Environments. — 2010. — .
- [11] Placernote: Build spatial apps. — URL: <https://placernote.com/> (online; accessed: 2021-05-18).
- [12] Vuforia Developer Portal |. — URL: <https://developer.vuforia.com/> (online; accessed: 2021-05-19).
- [13] Spatial Augmented Reality with Vuforia Engine in Unity. — URL: <https://resources.unity.com/unitenow/onlinesessions/spatial-augmented-reality-with-vuforia-engine-in-unity> (online; accessed: 2021-05-19).
- [14] *Wirola Lauri, Laine Tommi A., Syrjärinne Jari*. Mass-market requirements for indoor positioning and indoor navigation // 2010 International Conference on Indoor Positioning and Indoor Navigation. — 2010. — . — Pp. 1–7.
- [15] Indoor positioning: Challenges and solutions for indoor cultural heritage site / Tsvi Kuflik, Joel Lanir, Eyal Dim et al. — 2011. — . — Pp. 375–378.
- [16] Infrastructure-free indoor navigation: a case study / Davide Carboni, Andrea Manchinu, Valentina Marotto at al. // *Journal of Location Based Services*. — 2015. — . — Vol. 9, no. 1. — Pp. 33–54. — URL: <http://www.tandfonline.com/doi/full/10.1080/17489725.2015.1027751> (online; accessed: 2021-05-20).

- [17] WPIN: A waypoint-based indoor navigation system / Edward T.-H. Chu, Shih-Chia Wang, Chao-Chun Chang et al. // Short Paper Proceedings of the Tenth International Conference on Indoor Positioning and Indoor Navigation - Work-in-Progress Papers (IPIN-WiP 2019) co-located with the Tenth International Conference on Indoor Positioning and Indoor Navigation (IPIN 2019), Pisa, Italy, September 30th - October 3rd, 2019 / Ed. by Francesco Potortì, Valérie Renaudin, Kyle O’Keefe, Filippo Palumbo. — Vol. 2498 of *CEUR Workshop Proceedings*. — CEUR-WS.org, 2019. — Pp. 187–194. — URL: <http://ceur-ws.org/Vol-2498/short25.pdf>.
- [18] Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications / Ali Yassin, Youssef Nasser, Mariette Awad et al. // *IEEE Communications Surveys & Tutorials*. — 2017. — Vol. 19, no. 2. — Pp. 1327–1346. — URL: <http://ieeexplore.ieee.org/document/7762095/> (online; accessed: 2021-05-20).
- [19] Performance analysis of visualmarkers for indoor navigation systems / Gaetano C. La Delfa, Salvatore Monteleone, Vincenzo Catania et al. // *Frontiers of Information Technology & Electronic Engineering*. — 2016. — . — Vol. 17, no. 8. — Pp. 730–740. — URL: <http://link.springer.com/10.1631/FITEE.1500324> (online; accessed: 2021-05-20).
- [20] Пешком по тайлам. — URL: <https://habr.com/ru/company/2gis/blog/266753/> (online; accessed: 2021-05-19).
- [21] Introduction to algorithms / edited by Thomas H. Cormen, Thomas H. Cormen. — 2nd ed edition. — Cambridge, Mass: MIT Press, 2001.
- [22] Indoor Navigation Using A* Algorithm / Shahreen Kasim, Loh Yin Xia, Norfaradilla Wahid et al. // Recent Advances on Soft Computing and Data Mining / edited by Tutut Herawan, Rozaida Ghazali, Nazri Mohd Nawawi, Mustafa Mat Deris. — Advances in Intelligent Systems and Computing. — Cham: Springer International Publishing, 2017. — Pp. 598–607.
- [23] *Fellner Irene, Huang Haosheng, Gartner Georg*. “Turn Left after the WC, and Use the Lift to Go to the 2nd Floor”—Generation of Landmark-Based

- Route Instructions for Indoor Navigation // *ISPRS International Journal of Geo-Information*. — 2017. — . — Vol. 6, no. 6. — P. 183. — URL: <http://www.mdpi.com/2220-9964/6/6/183> (online; accessed: 2021-05-01).
- [24] Cross-Platform Indoor Navigation Using Mixed-Reality. — URL: <https://proxy.library.spbu.ru:2345/pqdtglobal/docview/2236378431/28545827E234AF2PQ/1?accountid=28393> (online; accessed: 2020-12-16).
- [25] WebGL - OpenGL ES for the Web. — 2011. — . — Section: API. URL: <https://www.khronos.org/webgl/> (online; accessed: 2021-05-18).
- [26] Legal Information. — URL: <https://unity3d.com/legal> (online; accessed: 2021-05-20).
- [27] *Technologies Unity*. Unity - Manual: Memory in WebGL. — URL: <https://docs.unity3d.com/Manual/webgl-memory.html> (online; accessed: 2021-05-20).
- [28] *Marciniak Jacek Bernard, Janicki Hubert*. Using a game engine to visualize the Main Building of the Warsaw University of Technology in a mobile application // *Polish Cartographical Review*. — 2020. — . — Vol. 52, no. 2. — Pp. 61–73. — Publisher: Sciendo Section: Polish Cartographical Review. URL: <https://content.sciendo.com/view/journals/pcr/52/2/article-p61.xml> (online; accessed: 2020-12-16).
- [29] *Technologies Unity*. Unity - Manual: Unity User Manual 2020.3 (LTS). — URL: <https://docs.unity3d.com/Manual/index.html> (online; accessed: 2021-05-20).
- [30] Система навигации для Факультета свободных искусств и наук. — URL: <http://passion.shishki.co/568013/> (online; accessed: 2021-05-19).
- [31] *Bayat Hasan*. hasanbayatme/unity-dijkstras-pathfinding. — 2021. — . — original-date: 2017-11-15T10:03:10Z. URL: <https://github.com/>

hasanbayatme/unity-dijkstras-pathfinding (online; accessed: 2021-05-20).

[32] Rocknroll-nerd/graph_diplom. — URL: https://github.com/Rocknroll-nerd/graph_diplom (online; accessed: 2021-05-20).