

1. Установка модулей для разработки Android приложений.

Open Unity Hub → Preferences → Installs → Gear → Add modules → Install Android Build Support.

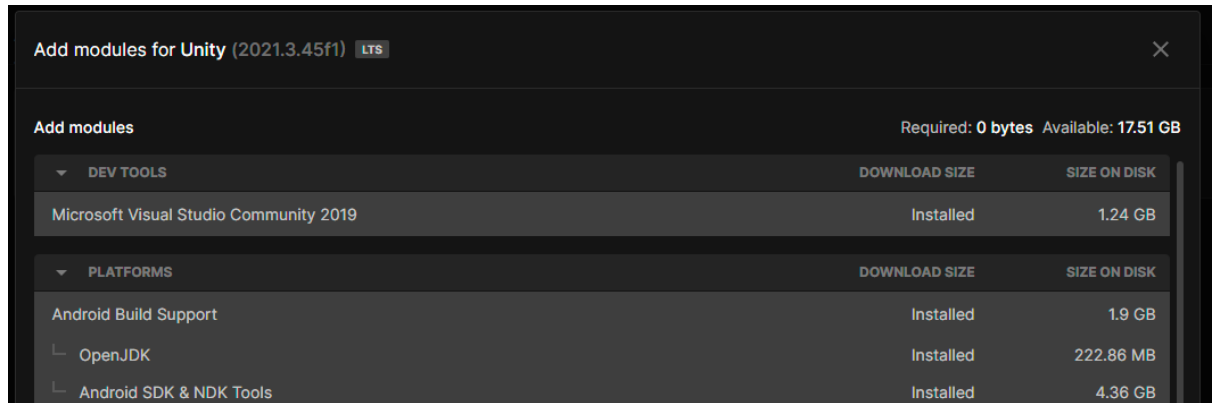
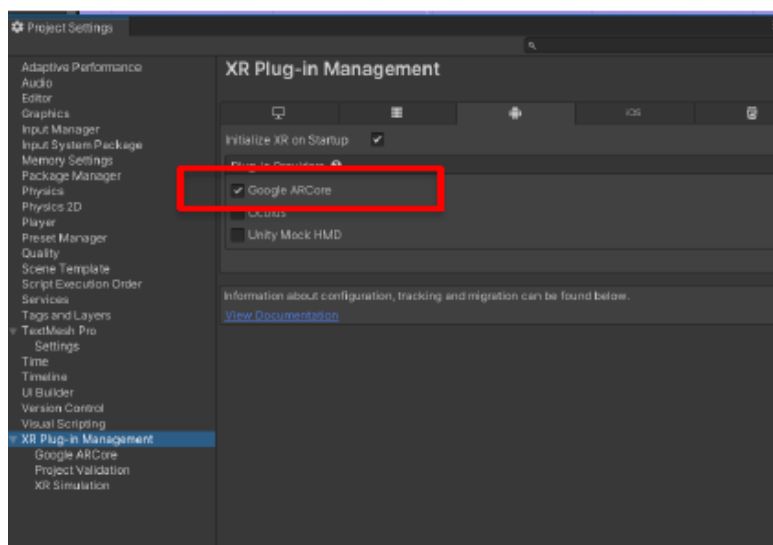


Рисунок 6 Android Build Support.

2. Необходимая установка Google AR core support и AR foundation package для начала разработки AR сцены.



Go to edit → project settings → XR plug-in Management to tick the GoogleARCore.

Рисунок 7 – GoogleARCore plugin install.

Window → package manager → Unity Registry to download the AR Foundation

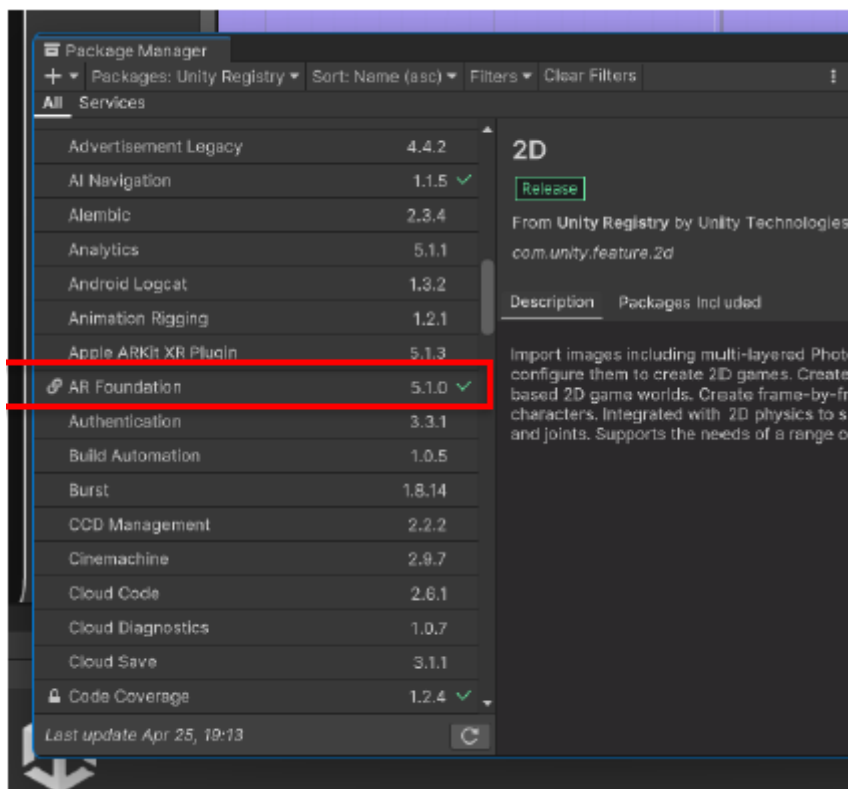


Рисунок 8 – AR Foundation support.

3. Настройки игрока:

1. Удалите Vulkan API.
2. Включите dynamic batching.
3. Установите минимальный уровень API на Android 7.
4. Установите бэкенд скриптов на IL2CPP и целевую архитектуру на ARM64.

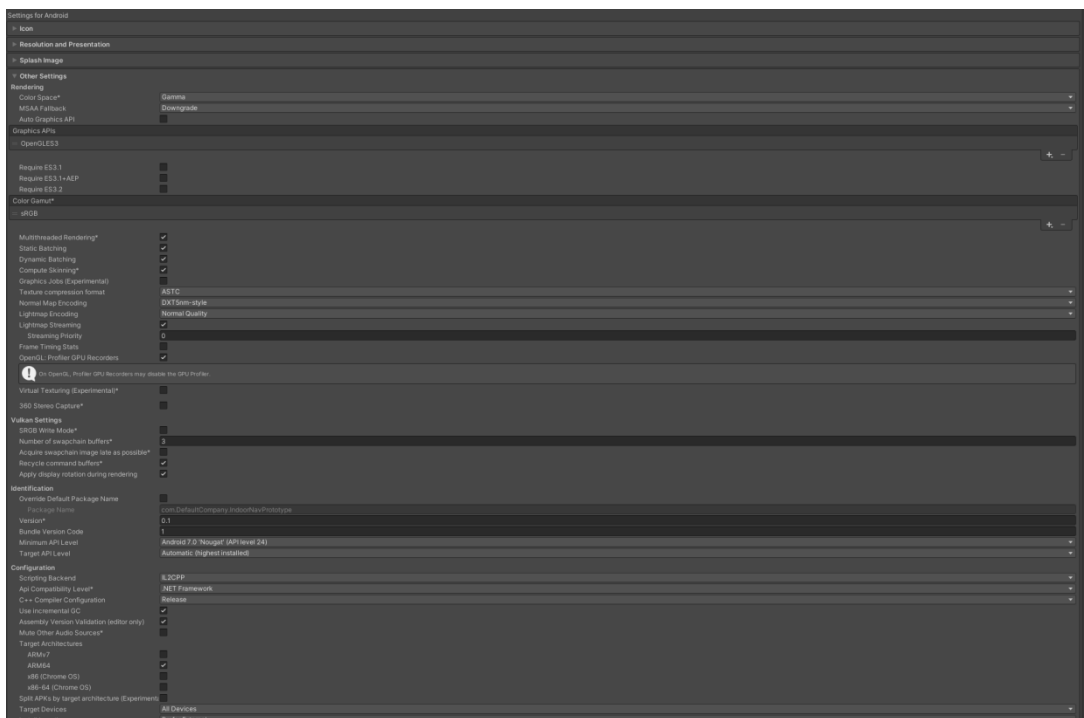


Рисунок 9 – Настройки игрока

4. Создание 2D карты тестового стенда, с реальными измерениями.

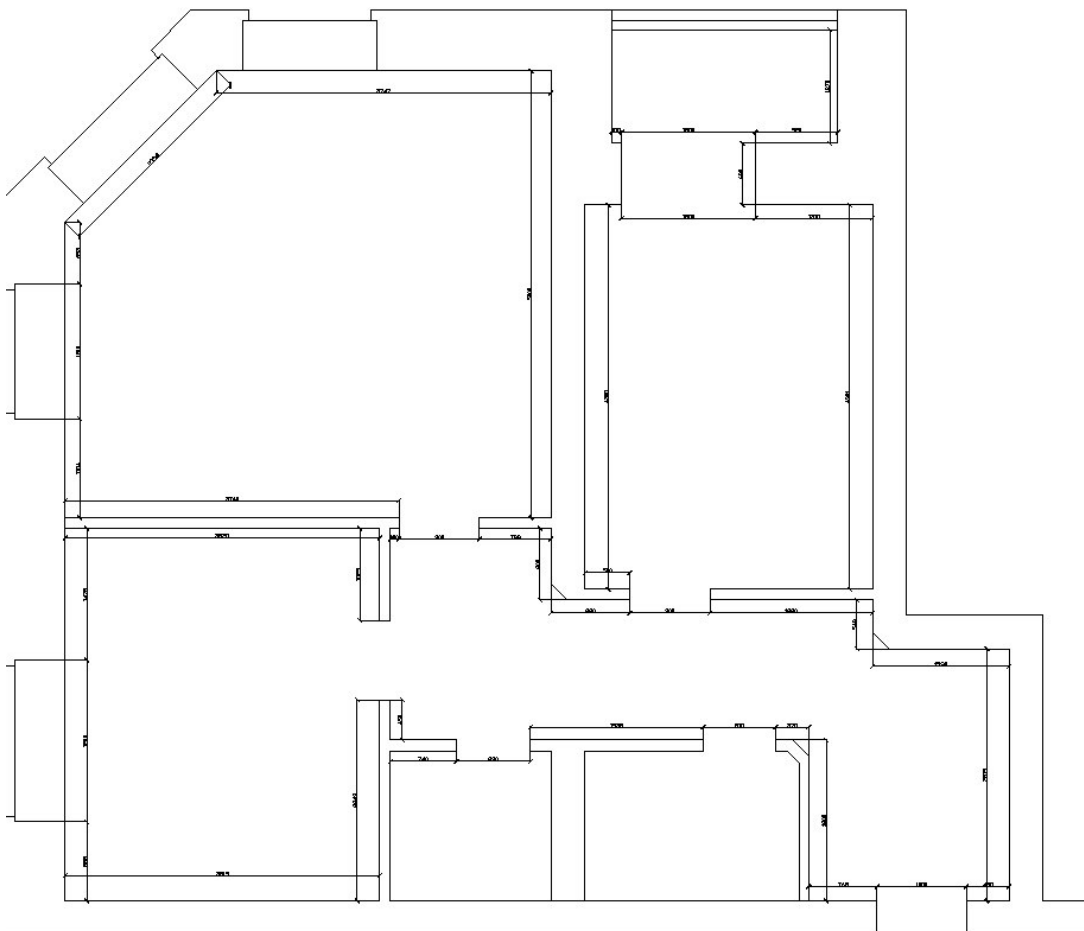


Рисунок 10 – 2D карта помещения

5. После настройки платформы Unity была разработана трехмерная модель тестового стенда. В качестве основы для построения стен использовалась 2D карта помещения, которая служил ориентиром для создания стен с использованием кубических объектов в Unity. Все размеры сделаны 1 к 1, учитывая, что 1 метр в Unity равен 1 метру в реальности.

1. Создайте папку для текстур и добавьте текстуру мини-карты.
2. Измените тип текстуры на Sprite (2D и UI).
3. Создайте пол, используя куб, и масштабируйте его, чтобы он соответствовал размерам комнаты.
4. Создайте новый материал для пола, установите шейдер на Unlit Texture и примените текстуру мини-карты.

Не бойтесь менять поворот вашего пола, также в графе Transform → Scale x и z – это ваши собственные значения, основанные на размерах вашего помещения, y – толщина пола может быть произвольной.

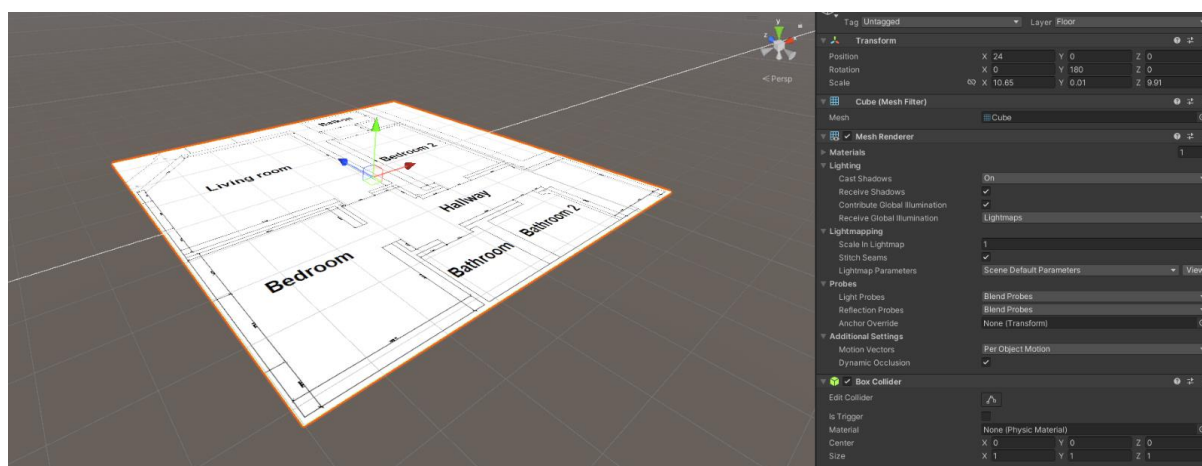


Рисунок 11 – FloorCube

Создайте пустой объект GameObject с именем "Environment" и сделайте пол дочерним объектом. Добавьте кубы для представления стен, корректируя их масштаб и положение в соответствии с планом комнаты.

Создайте материал для стен, используя шейдер, который позволяет им быть невидимыми, но при этом окклюдировать.

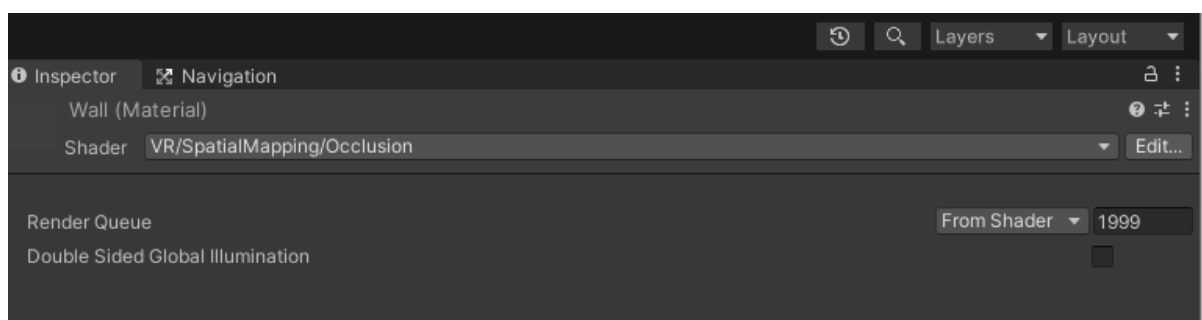


Рисунок 12 – VRSpatialMappingOcclusion материал для стен

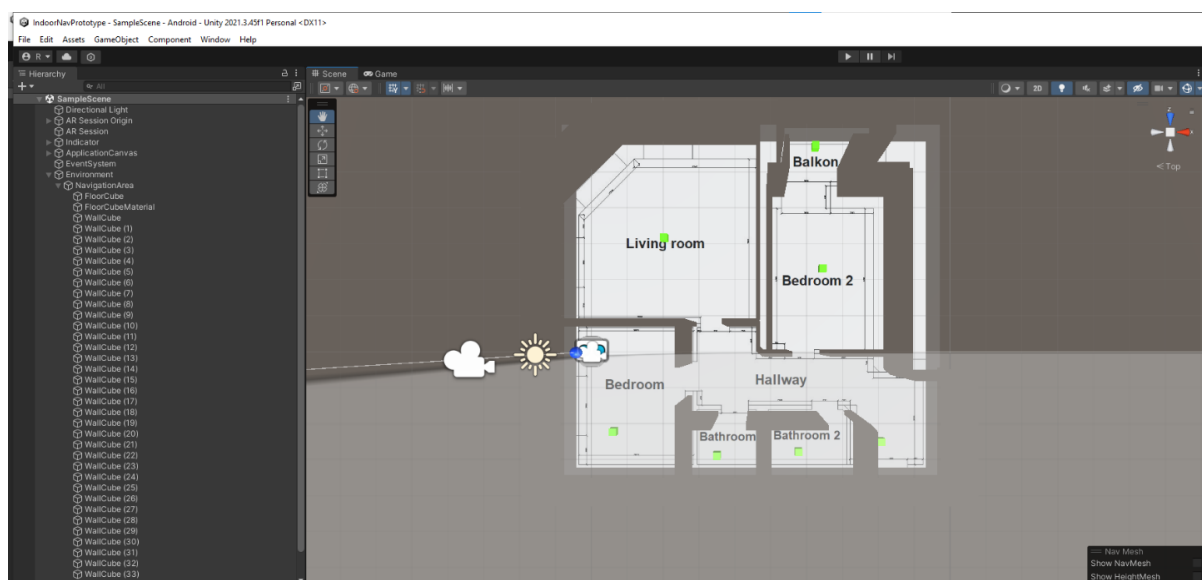


Рисунок 13 – 3D модель помещения.

6. Настройка навигационной сетки

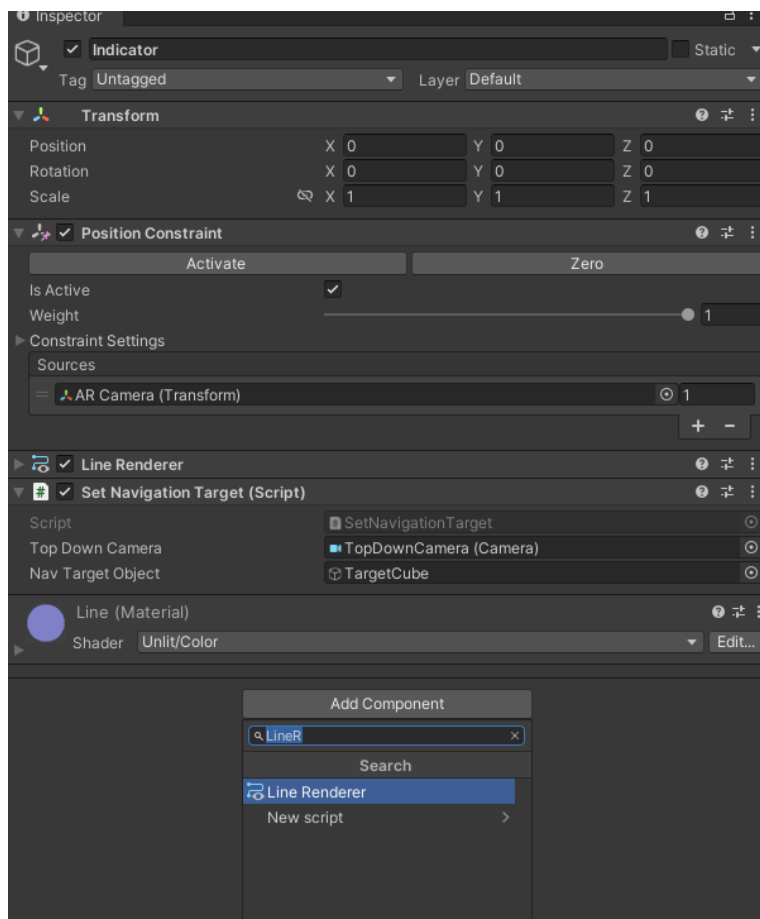
Go to Window → Ai → Navigation

Отметьте Environment как статичное и добавьте навигационную сетку. Настройте радиус агента, чтобы он проходил через комнату. Bake навигационную сетку, чтобы визуализировать проходимую область.

7. В процессе пространственного позиционирования была разработана система маркировки целевых локаций с использованием цветовой индикации и координатной привязки. Зеленый блок представляет

собой целевую точку навигации, а синяя сфера выступает в роли навигационного индикатора, размещенного в origin-точке картографической системы с координатами (0,0), что обеспечивает корректную инициализацию навигационного маршрута. В структуре индикатора интегрирована AR-камера, обеспечивающая захват пространственного окружения и наложение навигационной траектории на реальность. Дополнительно реализована топографическая камера верхнего обзора, позволяющая формировать мини-карту для альтернативной визуализации пространственных перемещений и предоставления пользователю альтернативной точки зрения.

1. Создайте целевой куб в ванной и назначьте ему зеленый материал.
 2. Создайте пустой объект индикатора и добавьте IndicatorSphere как ребёнка, чтобы представить текущее положение пользователя.
 3. Используйте ограничение позиции, чтобы индикатор следовал за целевым кубом.
-
8. Рендерер линии для навигационного пути
1. Добавьте рендерер линии, чтобы визуализировать путь от текущего положения к цели.

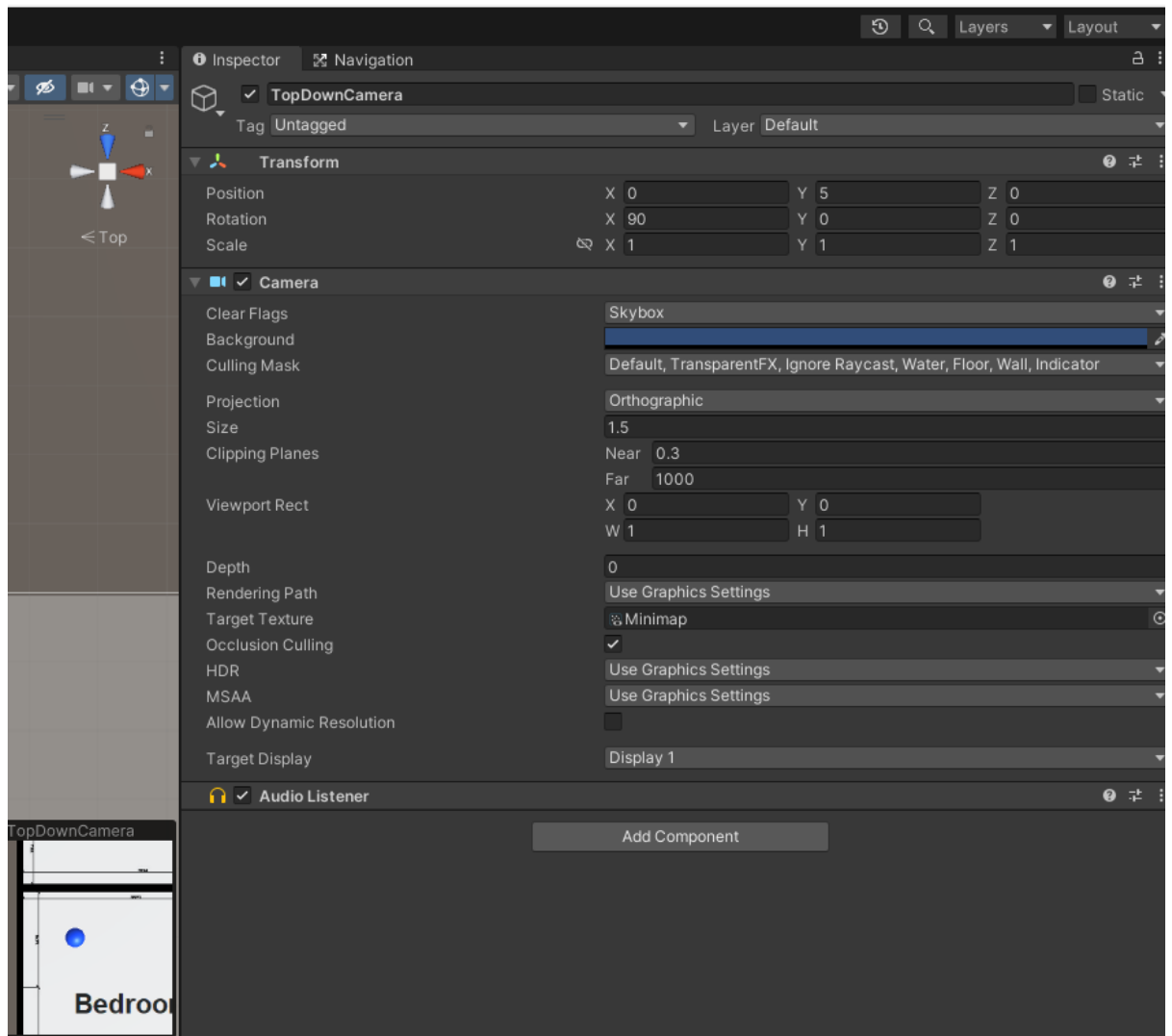


2. Создайте новый материал для линии и настройте его ширину.

Рисунок 14 – Line Renderer Component

9. Настройка камеры мини-карты

1. Добавьте камеру.
2. Установите камеру в ортографический режим и настройте ее положение (поверните на 90) и размер.



3. Создайте текстуру рендеринга для мини-карты и назначьте ее камере

Рисунок 15 – TopDownCamera

10. Пользовательский интерфейс для мини-карты

1. Создайте UI Canvas и добавьте Raw Image для отображения мини-карты.

2. Настройте размер и положение Raw Image, чтобы соответствовать желаемому макету.

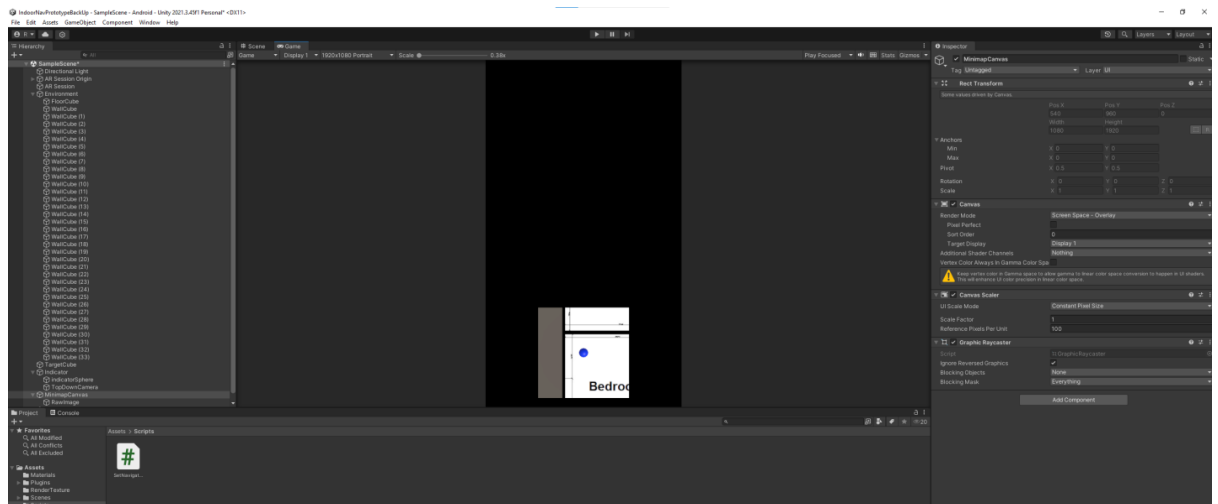


Рисунок 16 – TopDownCamera on User Interface

11. Скрипт SetNavigationTarget для корректного расчета NavLine.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.AI;
5
6  public class SetNavigationTarget : MonoBehaviour {
7
8      [SerializeField]
9      private Camera topDownCamera;
10     [SerializeField]
11     private GameObject navTargetObject;
12
13     private NavMeshPath path; // current calculated path
14     private LineRenderer line; // linerenderer to display path
15     private bool lineToggle = false;
16
17     private void Start() {
18         path = new NavMeshPath();
19         line = transform.GetComponent<LineRenderer>();
20     }
21
22     private void Update() {
23         if ((Input.touchCount > 0) && (Input.GetTouch(0).phase == TouchPhase.Began)) {
24             lineToggle = !lineToggle;
25         }
26
27         if (lineToggle) {
28             NavMesh.CalculatePath(transform.position, navTargetObject.transform.position, NavMesh.AllAreas, path);
29             line.positionCount = path.corners.Length;
30             line.SetPositions(path.corners);
31             line.enabled = true;
32         }
33     }
34 }

```

Рисунок 17 – SetNavigationTarget скрипт

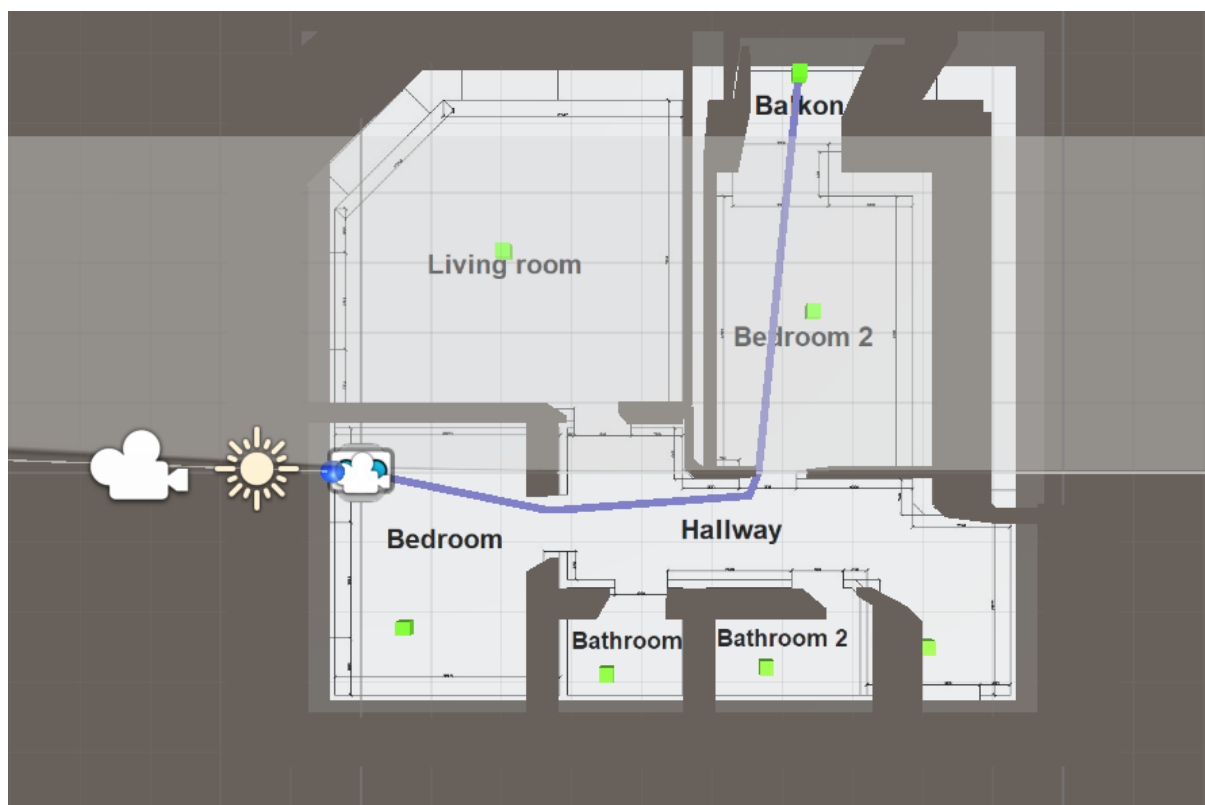
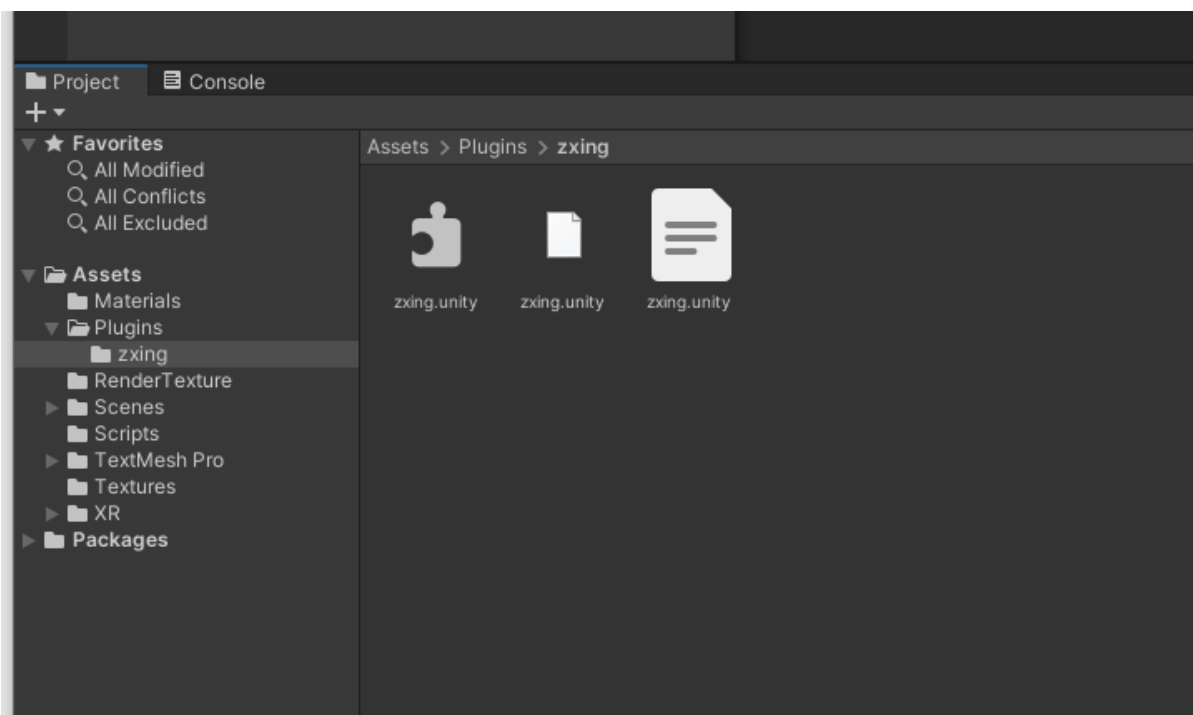


Рисунок 18 – Образец работы скрипта.

Таким образом, этот структурированный подход должен помочь вам реализовать собственную базовую версию AR-приложения для Indoor навигации в Unity. Если у вас есть конкретные вопросы или вам нужно больше разъяснений по какому-либо шагу, вы можете обратиться на почту fedyaneprit@gmail.com или задать вопрос в телеграмм канале, посвященному данному проекту, по ссылке <https://t.me/IndoorNav>.

Готовый результат можно просмотреть по ссылке:

https://rutube.ru/video/private/05178005c3e73078d5982785015c407f/?p=SRj6AE8ZVOJFUG_SmyPqTA



В проект встроена функция обновления местоположения при его потере в обычном режиме с помощью считывания QR кода. Zxing.Net Library — бесплатная библиотека с открытым исходным кодом, которая поддерживает декодирование и генерацию штрих-кодов. Декодирование QR-кода позволит приложению интерпретировать данные, закодированные в отсканированном QR-коде. Эта библиотека импортируется в проект как плагин, позволяющий сканировать QR-код для последующей локализации.

Рисунок 19 – Zxing.Net Library plugin.

QR Code Generator

Используйте бесплатный веб-сайт для генерации QR-кодов, чтобы создать QR-код для локализации в дальнейшем. Имя QR-кода должно совпадать с названием, используемым в проекте Unity, чтобы система могла правильно отслеживать информацию при сканировании. Это обеспечит корректное сопоставление местоположения на карте Unity с данными, полученными при сканировании QR-кода.

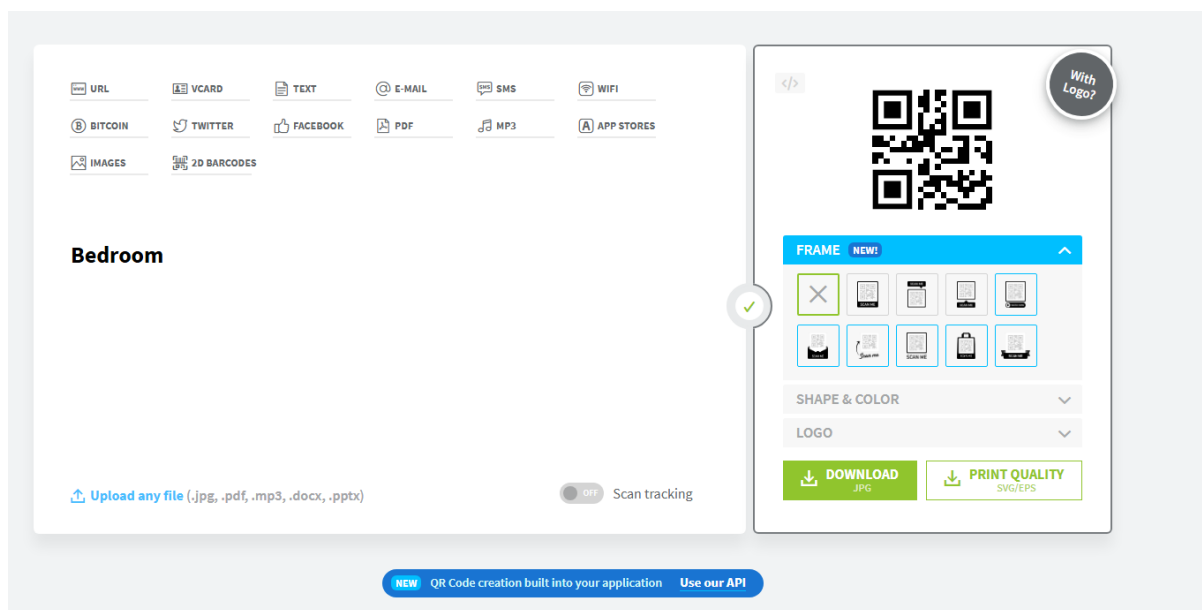
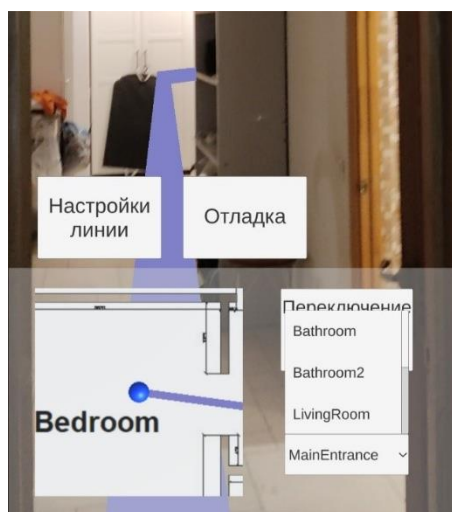


Рисунок 20 - QR Code Generator

Также были добавлено еще несколько функций, улучшающих пользовательский опыт.



Пользователь может выбрать пункт назначения для навигации после определения своего текущего местоположения в здании. Нажатие кнопки "Выбрать пункт назначения" в панели меню вызывает две кнопки в верхнем левом углу: "Показать линию" и раскрывающийся список локаций. Пользователь может выбрать желаемый пункт назначения и нажать "Показать линию" для включения AR-линии навигации на экране. Система позволяет пользователю скрывать линию, чтобы она не блокировала обзор при выборе пункта назначения.

Рисунок 21 – DropDown panel

Иногда из-за некоторых ошибок линия AR-навигации может блокировать обзор пользователя, поэтому панель переключения, которая может регулировать высоту навигационной линии, будет очень полезной. Она может опускать линию, когда она мешает, и поднимать ее, когда пользователь не видит линию.

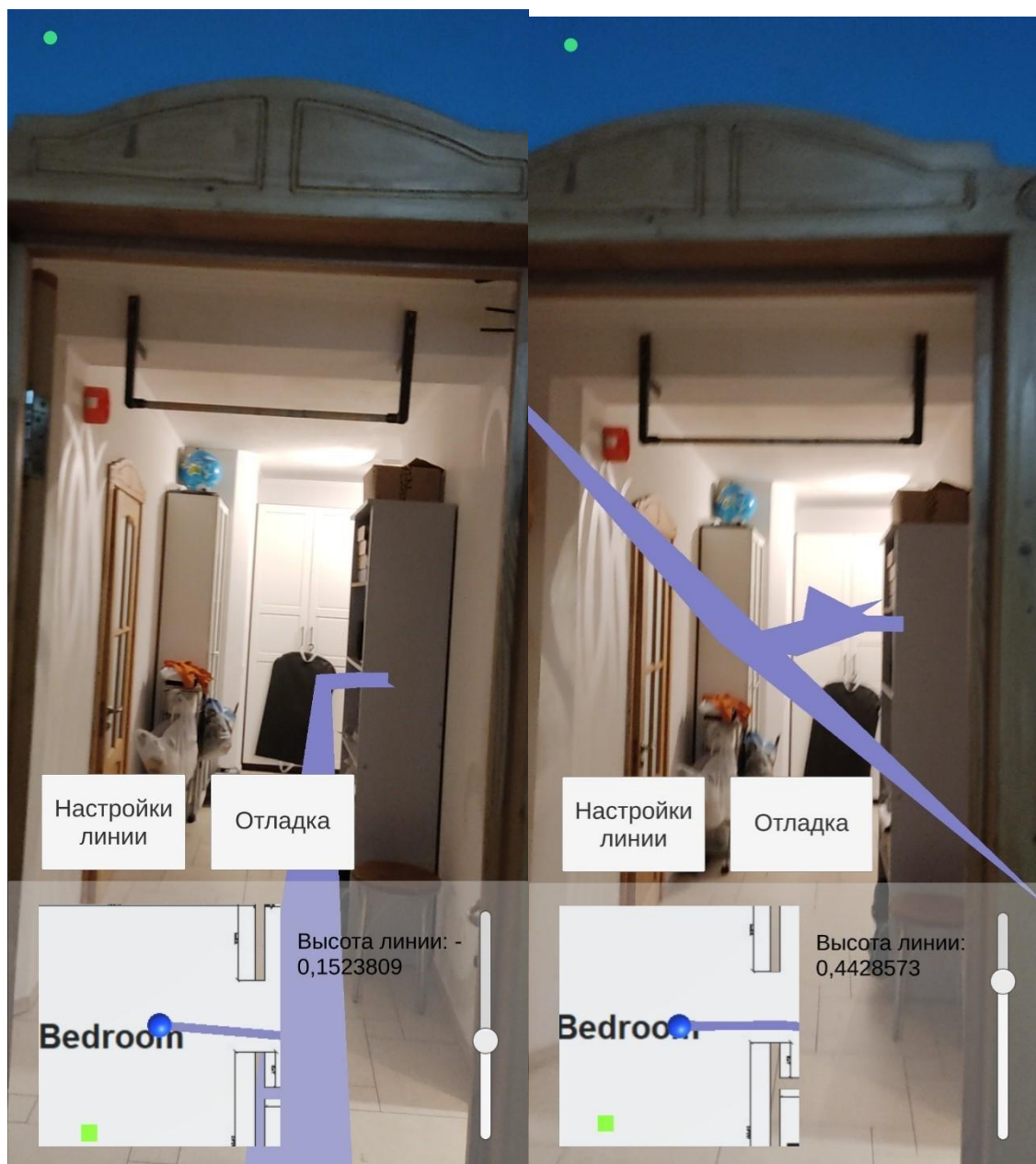


Рисунок 22 – Линия блокирует обзор
линии

Рисунок 23 – Изменение высоты

