

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 20.Б10-мм

Разработка плагина для импорта тайловых карт из редактора Tiled в игровой движок Godot

Мищенко Станислав Алексеевич

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ст. преп., И. В. Зеленчук

Санкт-Петербург
2022

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Особенности хранения и представления тайловой графики	6
2.2. Виды тайловых карт, поддерживаемых Tiled	6
2.3. Форматы экспорта в Tiled	8
2.4. Особенности хранения данных экспортированных карт в Tiled	9
2.5. Инструменты игрового движка Godot	9
2.6. Обзор существующих решений	10
2.7. Обзор используемых технологий	11
3. Метод	12
3.1. Архитектура проекта	12
3.2. Парсинг данных	12
3.3. Построение тайловой карты в движке	13
3.4. Объединение модулей	14
4. Тестирование	15
4.1. Тестирование корректного импорта различных тайловых карт	15
4.2. Сравнение представленного решения и аналогов	15
5. Применение	18
Заключение	19
Список литературы	20

Введение

В большинстве компьютерных игр большое внимание уделяется графике. На визуальную составляющую игрового программного продукта может тратиться значительная часть времени при разработке.

Ещё в самом начале развития игровой индустрии в условиях ограниченных возможностей вычислительной техники возникла необходимость создавать большие по размерам двумерные игровые миры и соответствующий объём графического контента [2]. С этой целью была придумана концепция тайловых карт. Тайл — изображение небольших размеров, являющееся составной частью большой картины. Таким образом, графика для всех игровых уровней может быть составлена из ограниченного числа таких фрагментов, что экономит как время людей, занимающихся созданием визуального оформления игры, так и ресурсы компьютера. Подобный принцип создания графики для игровых проектов за счёт своей простоты используется и сегодня.

В большинстве современных игровых движков есть встроенные инструменты для создания графики с использованием тайлов. Однако не везде предлагаемые решения для работы с тайловыми картами являются удобными в использовании и охватывают все потребности пользователей. Подобная проблема есть у игрового движка Godot [10]. Это свободно распространяемый движок с открытым исходным кодом. Он включает в себя инструменты для работы с двумерной и трёхмерной графикой, звуком, пользовательским вводом и не только. Игровой движок Godot постоянно поддерживается и постепенно набирает популярность у разработчиков игр. В то же время существуют сторонние решения, которые изначально были созданы для работы с тайловой графикой и поэтому предлагают пользователю больше возможностей в этой области, например Tiled [11]. Это программа с открытым исходным кодом, которая позволяет создавать и редактировать тайловую графику и предоставляет дополнительные возможности для работы с тайловыми картами.

Некоторых разработчиков игр может не устраивать встроенный в

Godot инструментарий для работы с тайловыми картами по причине его ограниченности или неудобства в использовании, и они предпочли бы использовать Tiled для работы с тайловой графикой.

1. Постановка задачи

Целью работы является разработка плагина для импорта тайловых карт из редактора Tiled в игровой движок Godot. Для её выполнения были поставлены следующие задачи:

1. Провести обзор инструментов для работы с тайловыми картами в игровом движке Godot и в редакторе Tiled и сравнить их;
2. Реализовать модуль для парсинга данных из формата, используемого в Tiled для экспорта тайловых карт;
3. Реализовать модуль для построения тайловой карты средствами движка на основе полученных после парсинга данных;
4. Объединить модули вместе в рамках одного плагина для движка;
5. Протестировать плагин на существующих тайловых картах.

2. Обзор

Для понимания специфики сравниваемых реализаций были описаны некоторые особенности игрового движка Godot, редактора Tiled и самой тайловой графики.

2.1. Особенности хранения и представления тайловой графики

Для экономии памяти компьютера и удобства использования, тайлы хранятся не как отдельные графические файлы, а объединяются в одном большом изображении, называемом атласом. Поскольку в большинстве случаев тайлы имеют одинаковый размер, то, зная его, можно просто выделить нужный тайл из атласа.



Рис. 1: Пример атласа для тайловой карты

2.2. Виды тайловых карт, поддерживаемых Tiled

В Tiled можно создавать тайловые карты пяти различных видов [3]:

- Ортогональная карта (Orthogonal map) — способ представления тайловой графики, при котором линии координатной сетки пер-

пендикулярны друг другу, образуя прямоугольные клетки (рис. 2).

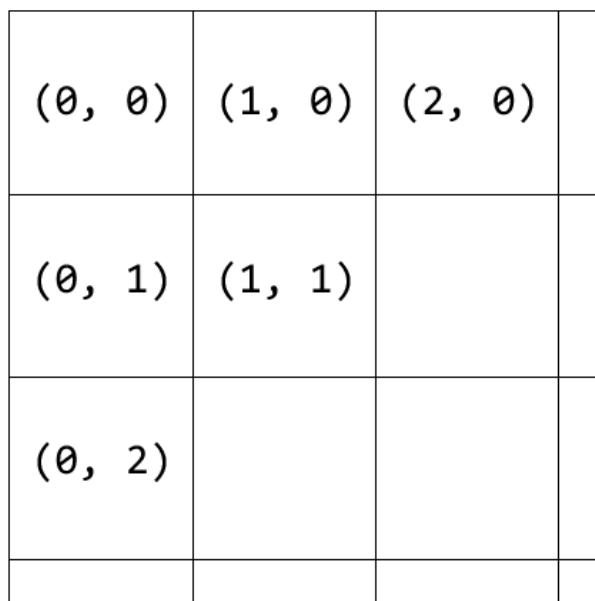


Рис. 2: Сетка ортогональной карты

- Изометрическая карта (Isometric map) — способ представления тайловой графики, при котором линии координатной сетки расходятся под одним углом от вертикали, образуя ромбовидные клетки (рис. 3).

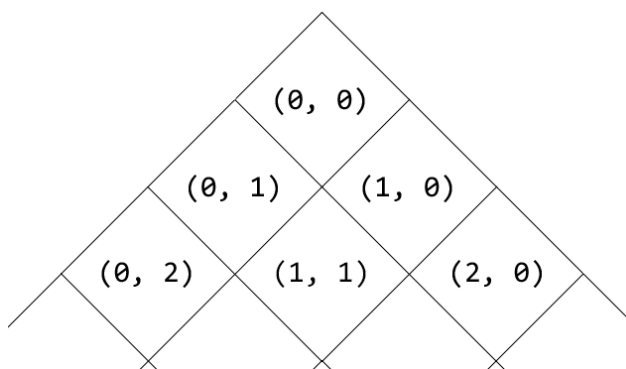


Рис. 3: Сетка изометрической карты

- Шахматная изометрическая карта (Staggered isometric map) — разновидность изометрической карты, отличающаяся порядком нумерования тайлов (рис. 4).

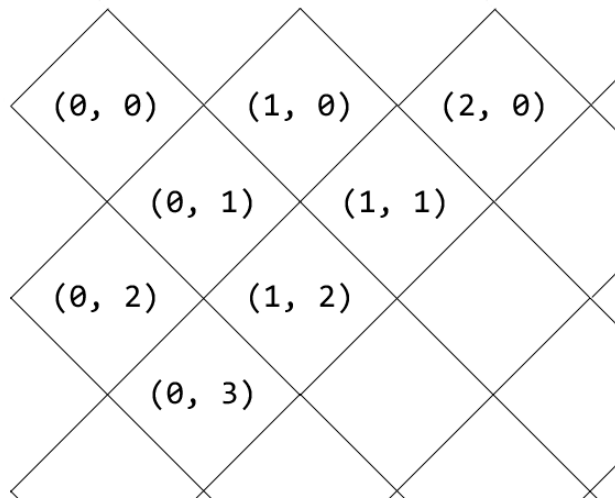


Рис. 4: Сетка шахматной изометрической карты

- Гексагональная карта (Hexagonal map) — способ представления тайловой графики, при котором используется координатная сетка, состоящая из шестиугольников (рис. 5).

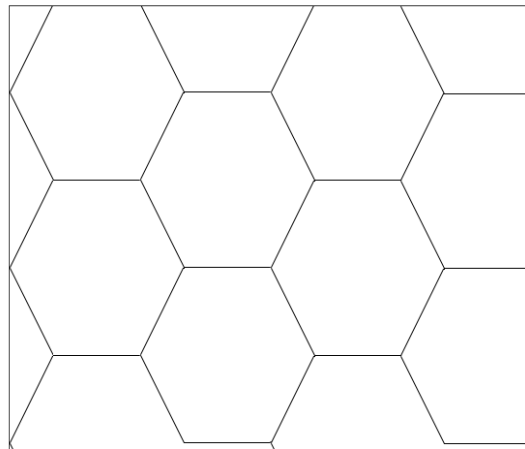


Рис. 5: Сетка гексагональной карты

2.3. Форматы экспорта в Tiled

Экспортировать созданную тайловую карту можно в нескольких форматах, предоставляемых редактором, из которых можно выделить два основных [3]:

- TMX — собственный формат редактора для хранения тайловых

карт, основанный на XML. Для этого формата информация, связанная с изображением, содержащим тайлы, может быть вынесена в вспомогательный файл формата TSX, который по структуре аналогичен TMX.

- JSON — формат, в котором параметры карты представлены в виде словарей и массивов значений.

Остальные форматы экспорта являются адаптированными для конкретных игровых движков, в число которых не входит Godot.

2.4. Особенности хранения данных экспортированных карт в Tiled

Tiled определяет набор структур, используемых для хранения карты и её содержимого, включающих различные параметры. Но выделяется среди них формат хранения тайлов карты. Tiled определяет карту как упорядоченный набор глобальных индексов (global id) [3], каждый из которых имеет уникальный номер, однозначно определяющий тайл из конкретного набора, используемого в проекте. Эти наборы индексов могут храниться непосредственно, в виде таблицы чисел (CSV) [5], однако Tiled предлагает возможность закодировать данные с использованием алгоритма Base64 [4], а также дополнительно применить сжатие алгоритмами ZLib [7], GZip [6] или ZStd [8].

2.5. Инструменты игрового движка Godot

В игровом движке Godot полностью поддерживаются два языка программирования: C# и GDScript [1] - собственный язык движка, обладающий динамической типизацией и схожий по синтаксису с Python. В настоящее время использовать язык C# можно только в версии движка Godot Mono, которая имеет все возможности стандартной версии, в том числе использование GDScript как второго языка в проекте.

Для работы с тайловой графикой в движке имеются встроенные инструменты. Они позволяют формировать наборы тайлов, используемые для построения тайловой карты, и предоставляют интерфейс для назначения определённого тайла на выбранную на карте позицию [1].

Для формирования набора тайлов есть возможность как вручную выделять определённый участок изображения для каждого тайла, так и использовать инструмент для автоматической разметки, если изображение является атласом с тайлами фиксированного размера. В последнем случае на основе заданных параметров формируется сетка, по которой тайлы выделяются из атласа. Второй способ является более предпочтительным, поскольку значительно экономит вычислительные ресурсы системы.

2.6. Обзор существующих решений

Из всех решений, позволяющих каким-либо образом импортировать тайловые карты из редактора Tiled в игровой движок Godot, можно выделить одну реализацию, которая корректно работает с последней версией движка и не только извлекает данные из экспортированного формата, но и формирует на их основе тайловую карту [13].

2.6.1. Tiled Map Importer

Данное решение было реализовано Джорджем Маркесом. Оно создано с помощью языка программирования GDScript и позволяет импортировать тайловые карты из редактора Tiled в игровой движок Godot. Оно поддерживает все представленные в Tiled виды тайловых карт и позволяет использовать тайловые карты, экспортированные в TMX и JSON форматах. Для данных тайлов карты поддерживается как CSV представление, так и кодировка Base64 с возможностью последующего сжатия ZLib или GZip. Из минусов данного решения можно выделить то, что скорость его работы сильно зависит от размеров импортируемой тайловой карты. Причиной этому служит GDScript, который является интерпретируемым языком программирования с дина-

мической типизацией.

Также данное решение не использует возможность формирования тайлесетов на основе атласа, создавая тайлы по отдельности. Это создаёт проблемы с производительностью для тайловых карт большого размера.

Эта реализация поддерживает не все возможные гексагональные карты, поскольку не учитывает возможность поворота тайлов для данного типа карт.

2.7. Обзор используемых технологий

В данном решении был выбран C# в качестве языка программирования. Он обладает статической типизацией и является компилируемым, что даёт преимущество в скорости выполнения кода по сравнению с интерпретируемыми языками с динамической типизацией, в число которых входит GDScript.

Из двух представленных основных экспортируемых форматов был выбран JSON. Он аналогичен по структуре с TMX, но, в отличие от него, не имеет дополнительных вспомогательных файлов, описывающих параметры для изображения с тайлами. Вся информация о карте содержится в одном JSON файле.

Для парсинга файлов выбранного формата была выбрана JSON библиотека, предоставляемая API движка Godot [1]. Она обладает достаточной скоростью работы и не требует включения в проект сторонних зависимостей, а значит использование плагина другими людьми не вызовет у них дополнительных трудностей.

3. Метод

3.1. Архитектура проекта

Плагин включает в себя два модуля, один из которых предназначен для парсинга данных, а другой для построения тайловой карты на основе полученных после парсинга данных (рис. 6).

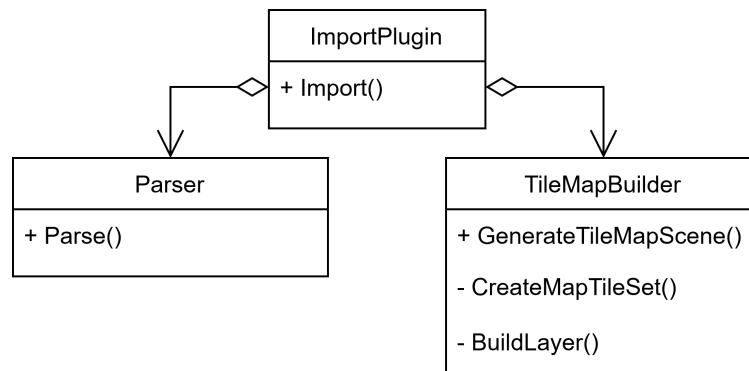


Рис. 6: Основные модули плагина

3.2. Парсинг данных

Написание плагина началось с реализации модуля парсинга данных. Изначально необходимо было объявить все структуры данных, которые используются в Tiled (рис. 7). Официальная документация Tiled явно

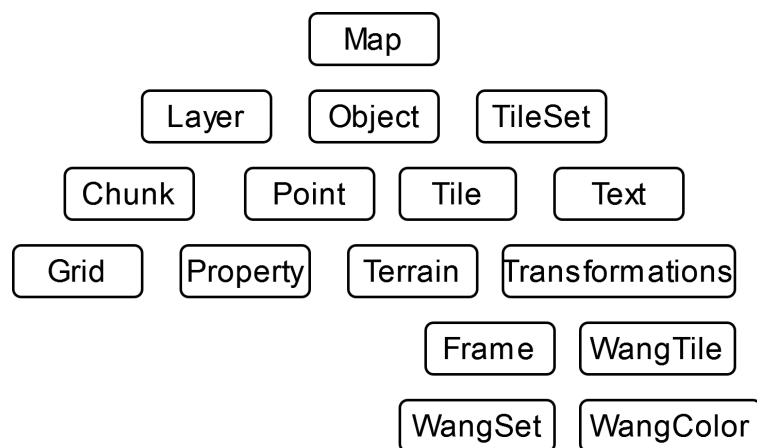


Рис. 7: Структуры данных, используемые в Tiled

указывает параметры и их тип для каждой структуры [3].

Следующим этапом было написание самого парсера данных. Изначально он был реализован как единый класс с большим количеством методов, каждый из которых отвечал за парсинг определённой структуры. Это создавало проблемы с расширением существующего кода, его отладкой. Также в некоторых структурах имелись параметры, которые были опциональными, однако их наличие могло влиять на парсинг других параметров. Это создавало необходимость постоянно проверять существование таких параметров, что создавало большие объёмы одинакового кода. Поэтому было принято решение применить паттерн Компоновщик (рис. 8). Это позволяет воссоздать аналогичную древовидную структуру, которая используется в экспортированном файле.

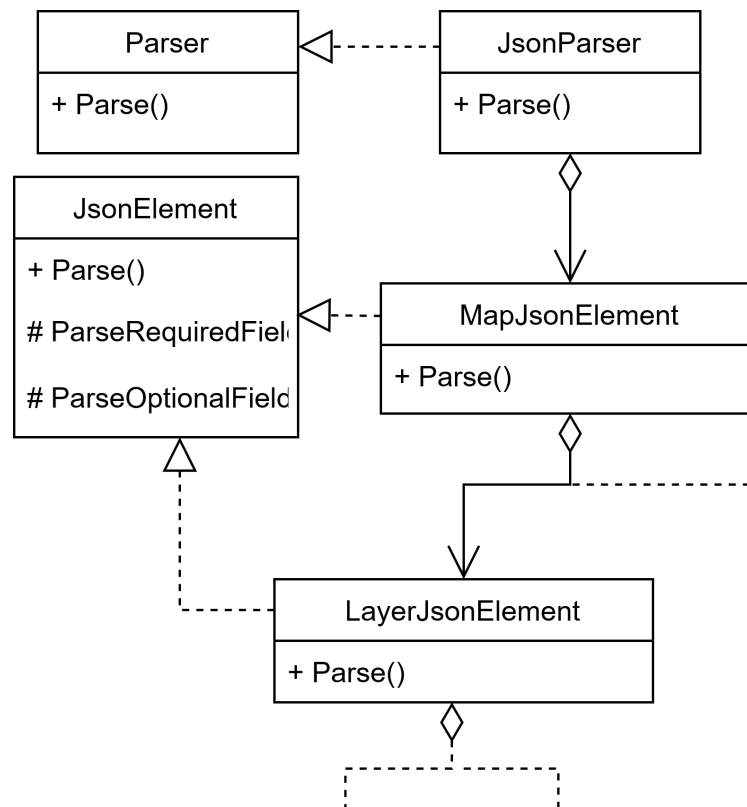


Рис. 8: Устройство модуля парсинга

3.3. Построение тайловой карты в движении

После того, как модуль парсинга данных корректно извлекал данные из файла, необходимо было написать модуль, который бы воссо-

здавал карту в движении по полученной информации.

Для этой цели использовался встроенный инструмент, который позволяет создавать тайловые карты. Благодаря возможности изменять его параметры через API [1] получилось создать игровую сцену с построенной тайловой картой, которую пользователь сможет открыть после того, как закончится процесс импорта.

Карты ортогонального и изометрического типа поддерживались встроенным в движок инструментом и поэтому импорт карт этих типов не вызывал трудностей.

Однако поддержки гексагональных и шахматных изометрических карт в движении не предусматривалось. Встроенный инструмент не имел поддержки гексагонального позиционирования тайлов, поэтому его пришлось реализовывать в коде. Что касается шахматных изометрических карт, которые отличаются от изометрических способом нумерования тайлов, то проблема отсутствия поддержки этого типа была решена переводом координат из одной системы в другую.

3.4. Объединение модулей

Когда модули парсинга и построения карты в движении были полностью реализованы, их взаимодействие было налажено в классе плагина, который реализует определённый движком интерфейс и позволяет автоматически обрабатывать добавляемые в проект файлы [1].

4. Тестирование

4.1. Тестирование корректного импорта различных тайловых карт

Был проведён тест плагина на импорт различных карт из примеров карт в официальном репозитории Tiled [12]. Среди них имелись карты всех видов и с различными наборами параметров. Все они корректно отображались в движке после импорта [9].

4.2. Сравнение представленного решения и аналогов

Конфигурация оборудования на котором проводилось тестирование:

- Процессор — AMD Ryzen 5 3550H (2.10 ГГц).
- Видеокарта — NVIDIA GeForce GTX 1650 4 ГБ.
- Оперативная память — 8 ГБ (2400 МГц).
- Операционная система — Windows 10.

4.2.1. Тестирование импорта тайловой карты

Для теста была взята ортогональная тайловая карта размера 480 на 480 тайлов. Размеры каждого тайла: 32 на 32 пикселя. Данные тайлов были представлены закодированными с помощью алгоритма Base64 и сжатыми с помощью ZLib.

Размер тайловой карты намеренно был выбран большим, поскольку это снижает погрешность измерения и позволяет проверить работу представленных реализаций в нагрузке. Тестируемым решениям придётся декодировать данные тайлов, что также увеличивает вычислительную сложность.

Экспортированный файл вместе с изображением, содержащим тайлы, перемещался в папку с проектом, что запускало выполнение программы импорта. Время выполнения алгоритма измерялось с помощью встроенных в язык инструментов посредством нахождения разницы во времени между моментами окончания и начала работы плагина.

4.2.2. Тестирование процесса отрисовки кадра

Для теста использовалась тайловая карта с теми же параметрами, что и в предыдущем тестировании.

Созданная игровая сцена с тайловой картой запускалась в движке, так, чтобы вся тайловая карта была видна на экране. Это необходимо для того, чтобы создавать максимальную нагрузку для процесса отрисовки изображения. Измерялось время, затрачиваемое на отрисовку одного кадра с помощью встроенного в игровой движок отладчика.

4.2.3. Исследовательские вопросы

Данная работа преследует следующие цели, в сравнении с другими решениями:

- Процесс импорта карт занимает меньше времени.
- Время отрисовки кадра для созданной сцены с тайловой картой занимает меньше времени.

4.2.4. Результаты

Было проведено 10 замеров для обоих из проводимых тестирований. Результаты представлены в таблице 1.

Реализация	Параметр	Импорт тайловых карт (время, мкс)	Процесс отрисовки кадра (время, мкс)
Представленное решение	Мат. ожидание	967.0	30.0
	Среднекв. откл.	11.0	1.5
Tiled Map Importer	Мат. ожидание	4006.0	336.0
	Среднекв. откл.	119.0	13.4

Таблица 1: Результаты тестирований

В условиях эксперимента, как видно из результатов тестирований, представленная реализация, по сравнению с решением Tiled Map Importer, примерно в 4 раза производит импорт тайловой карты, а время отрисовки кадра для запущенной игровой сцены занимает примерно в 11 раз меньше времени. Это показывает, что представленный плагин выигрывает у аналога по обоим из рассматриваемых параметров.

5. Применение

Данный плагин может использоваться для создания тайловых карт для компьютерных игр в Tiled и последующего их импорта в игровой движок Godot. Также представленное решение позволяет импортировать тайловые карты из большой библиотеки карт, когда-либо созданных в Tiled.

Заключение

Был реализован плагин для импорта тайловых карт из редактора Tiled в игровой движок Godot.

Были выполнены следующие задачи:

- Проведён обзор и сравнение инструментов для работы с тайловыми картами в игровом движке Godot и в редакторе Tiled.;
- Реализован модуль для парсинга данных из формата, используемого в Tiled для экспорта тайловых карт;
- Реализован модуль для построения тайловой карты средствами движка на основе полученных после парсинга данных;
- Модули объединены вместе в рамках одного плагина для движка.
- Плагин протестирован на существующих тайловых картах.

Репозиторий GitHub: https://github.com/mi-sts/godot_tiled_importer

В дальнейшем планируется развивать данный плагин. В него можно добавить поддержку следующих вещей:

- Парсер для формата файлов ТМХ.
- Поддержку объектов тайловых карт.
- Поддержку анимированных тайлов.
- Поддержку свойств.
- Поддержку для импорта фигур.

Список литературы

- [1] Godot Documentation - 3.4 branch. — URL: <https://docs.godotengine.org/> (online; accessed: 2022-05-20).
- [2] Graphics in early arcade games: tilemaps. — URL: <https://www.retrogamedeconstructionzone.com/2020/01/graphics-in-early-arcade-games-tilemaps.html> (online; accessed: 2022-05-20).
- [3] Lindeijer Thorbjørn. Tiled Documentation. — URL: <https://doc.mapeditor.org/> (online; accessed: 2022-05-20).
- [4] Wikipedia. Base64. — URL: <https://en.wikipedia.org/wiki/Base64> (online; accessed: 2022-05-20).
- [5] Wikipedia. Comma-separated values. — URL: https://en.wikipedia.org/wiki/Comma-separated_values (online; accessed: 2022-05-20).
- [6] Wikipedia. GZip. — URL: <https://en.wikipedia.org/wiki/Gzip> (online; accessed: 2022-05-20).
- [7] Wikipedia. ZLib. — URL: <https://en.wikipedia.org/wiki/Zlib> (online; accessed: 2022-05-20).
- [8] Wikipedia. ZStd. — URL: <https://en.wikipedia.org/wiki/Zstd> (online; accessed: 2022-05-20).
- [9] Демонстрация работы плагина (Godot Tiled Importer). — URL: <https://www.youtube.com/watch?v=SzVh1GypC-w> (online; accessed: 2022-05-20).
- [10] Официальный сайт игрового движка Godot. — URL: <https://godotengine.org> (online; accessed: 2022-05-20).
- [11] Официальный сайт редактора тайловых карт Tiled. — URL: <https://www.mapeditor.org> (online; accessed: 2022-05-20).

- [12] Репозиторий GitHub mapeditor/tiled (примеры тайловых карт). — URL: <https://github.com/mapeditor/tiled/tree/master/examples> (online; accessed: 2022-05-20).
- [13] Репозиторий GitHub vnen/godot-tiled-importer (Tiled Map Importer). — URL: <https://github.com/vnen/godot-tiled-importer/> (online; accessed: 2022-05-20).