Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы обработки информации и управления» (ИУ5)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Диалоговая генерация архитектурного дизайна зданий с использованием базы знаний

Студент ИУ5-32М _____ **Д.С. Кириллов**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

2024 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой __ИУ5__
(Индекс)
_____ ___В.И. Терехов___
(И.О.Фамилия)
« _____ » _____ 2024 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме ____Диалоговая генерация архитектурного дизайна зданий_____

_____с использованием базы знаний_____

_____

Студент группы ____ИУ5-32М__

_____Кириллов Денис Сергеевич_____
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

_____Исследовательская _____

Источник тематики (кафедра, предприятие, НИР) ____Кафедра _____

График выполнения НИР: 25% к ___ нед., 50% к ___ нед., 75% к __ нед., 100% к ___ нед.

Техническое задание _____
____Составить описание предметной области_____
____Проанализировать существующие системы-аналоги_____
____Составить обзор интеллектуальных систем и методов представления знаний_____
_____

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на ____ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

_____
_____
_____

Дата выдачи задания « _9__ » ____сентября____ 2024 г.

Руководитель НИР _____ __Ю.Е. Гапанюк_____
(Подпись, дата) (И.О.Фамилия)

Студент _____ __**Д.С. Кириллов**____
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре

Содержание

Введение

Генеративный искусственный интеллект играет важную роль в промышленности и имеет множество применений. Обычно он представляет собой алгоритмы принятия решений со стохастической вариативностью.

Генеративный искусственный интеллект способен создавать новые идеи и концепции на основе имеющейся информации, анализировать и выделять существенные элементы в данных и преобразовывать их в новаторские идеи. Генеративный искусственный интеллект способен видеть связи и взаимосвязи, которые часто остаются незамеченными для человека, а также обладает уникальной способностью переосмысливать информацию и предлагать альтернативные решения задач.

Важной его особенностью является поэтапное применение правил из нескольких предметных областей для имитации творчества на подобии того, как творит человек. Сначала генерируется эскиз, карта или иное упрощенное представление будущего результата, затем по особым правилам схематические линии связываются и дополняются деталями.

Этот подход применяется и в более сложных гибридных интеллектуальных системах искусственного интеллекта [1].

Генеративный искусственный интеллект способен учиться на примерах и самостоятельно формировать новые знания. Однако, несмотря на все свои преимущества, ГИИ все еще требует вмешательства человека для проверки результатов и принятия решений. Человек, в свою очередь, определяет, какие из идей и концепций, сгенерированных искусственным интеллектом, подходят для конкретной ситуации и могут быть реализованы на практике.

Таким образом, генеративный искусственный интеллект представляет собой мощный инструмент для создания новых идей и концепций, который может быть использован в различных областях.

Предназначение системы

Система диалоговой генерации архитектурного дизайна зданий предназначена для создания первоначального дизайн-проекта промышленных зданий. К этой категории относятся преимущественно большие многоэтажные помещения: производственные цеха, склады, торговые центры, муниципальные учреждения.

Помимо дизайна система также может быть использована для примерной оценки количества материала, необходимого для реализации проекта, что позволит пользователю оценить экономические характеристики проекта, его стоимость, на самых ранних этапах.

Пользователь задает с помощью веб-приложения изначальный контур здания от вида сверху. Указывает желаемую высоту и параметры локальной генерации: цветовую палитру, вариативность паттернов узоров на фасаде и т.д.

Сначала определяются центроиды плоских геометрических фигур. Определяется вид симметрии - радиальная или зеркальная. Основные пространства становятся залами, между ними прокладываются коридоры. Если пространство слишком большое, оно делится на комнаты.

Затем здание генерируется вверх. Учитывается параметр пользователя крыша: плоская / с узором / произвольная. Для произвольной накладывается

функция шумов с максимумами в центроидах - так генерируются башни, надстройки, балконы.

Выделяются помещения под коммуникации.

Далее генерируются детали интерьеров - лестницы, арки, дверные проемы, узоры на полах и стенах.

Разные этапы генерации имеют свои правила, которые содержатся во внешней базе знаний, содержащей формализованную информацию о том, какие цвета сочетаются, о том, как согласуются смежные помещения, о том, как генерируются торцы зданий, колонны внутри здания и т.д. Результат формируется процедурным алгоритмом.

Определение входных данных системы

Требуется обеспечить простой интерфейс ввода значений для генерации. Таким образом, воспользоваться системой сможет даже неподготовленный пользователь.

Основой будущего проекта здания является его внутренняя компоновка – план помещения, описывающий компоновку комнат исходя из предназначения помещений. Большие комнаты создаются под конференц-залы и офисные коворкинг-пространства. Малые комнаты используются сотрудниками офиса или иным персоналом. В промышленных зданиях могут встречаться технические помещения и огромные машинные залы, предназначенные для установки оборудования.

Ввод плана помещения предлагается осуществлять путем загрузки изображения специального формата, которое должно отображать в двухмерном виде минимально:

- Размер и форму здания;
- Размер и расположение комнат;
- Пути перемещения персонала: коридоры, лестницы, лифты;
- Окна;
- Входы в здание.

Далее предлагается задать параметры, описывающие вертикальную компоновку здания.

- Высоту этажей;
- Наличие технических этажей;
- Число этажей.

Ввод плана помещения должен осуществляться в специализированном графическом редакторе. Пользователю предлагается вводить элементы изображения по слоям. Сначала задается форма здания в горизонтальном разрезе, затем стены, далее коридоры, потом остальные элементы. На этом этапе уже применяются простейшие правила, выражающие ограничения предметной области:

- Комнаты имеют общие стены;
- Каждая комната должна быть соединена хотя бы с одной другой комнатой или коридором.
- Окно размещается в граничной стене между комнатой или коридором и улицей.

Остальные опции, влияющие на генерацию, предлагается вводить через традиционные формы. К основным таким опциям, прежде всего, относится параметры, задающую общую стилистику, отражающую архитектурные стили: конструктивизм, брутализм, авангард, модернизм, хайтек, футуризм, капиталистический утилитаризм и т.д.

Определение формата выходного результата

3D-модели зданий могут быть представлены в разных форматах данных, включая:

1. OBJ (Wavefront Object) - один из наиболее распространенных форматов для хранения 3D-моделей;
2. STL (STereoLithography) - формат, используемый для представления трехмерных геометрических данных. Он обычно используется для 3D-печати;

3. DAE (Collada) - открытый формат, разработанный для обмена содержимым между 3D-приложениями;
4. 3DS (3D Studio) - формат, разработанный компанией Autodesk для своих редакторов, которые получили широкое распространение;
5. IFC (Industry Foundation Classes) - открытый стандарт, разработанный специально для обмена информацией о зданиях и строительных объектах. Он позволяет хранить различные аспекты здания, включая геометрию, материалы, параметры конструкции;
6. BIM (Building Information Modeling) - не формат данных, а концепция, объединяющая информацию о зданиях и строительных объектах в одном месте. BIM может быть представлен в различных форматах, включая IFC, Revit, ArchiCAD и другие.

Однако, генерация десятков сотен тысяч точек всей 3D-модели требует как значительных вычислительных ресурсов системы, так и высокой сложности алгоритма генерации. Несмотря на то, что архитекторы работают с 3D-форматами в специализированных редакторах для просчета прочностных и иных физических и экономических характеристик проекта, сама модель должна строится поэтапно, начиная с простого эскиза. При поэтапном подходе неизбежно возникновение ошибок [3], и их тем больше, чем сложнее процесс генерации. Для фильтрации этих ошибок применяют отдельные сложные алгоритмы, и они тоже должны использовать ту же базу знаний, что и алгоритм генерации. Сама база знаний также может содержать ошибки [2].

В виду вышесказанного, целесообразно разработать свое упрощенное представление модели здания, достаточно универсальное, чтобы, с одной стороны, дать информацию о геометрии, а с другой, чтобы сохранить свойства объекта в той же нотации, в которой ими оперирует база знаний. И уже поверх этой модели можно написать алгоритм преобразования, основанный, например, на структурном паттерне «адаптер», который позволит преобразовать модель в формат для вывода в том или ином 3D-визуализаторе или для экспорта в CAD-систему для дальнейшей проработки опытным архитектором.

Таким упрощенным представлением, реализованным на языке программирования, может быть обычный трехмерный массив, элементы которого являются хэш-таблицами, файлами или структурами данных. В этом случае, получается упрощенная структура, состоящих из «блоков» и имеющая манхэттенскую меру расстояния. Все расчеты в таком представлении упрощаются, а преобразование к современному представлению 3D-графики сводятся к еще одному этапу генерации – аппроксимации смежных блоков одного вида до поверхностей и сглаживанию.

Порядок генерации

Генерация заключается в последовательной обработке различными алгоритмами промежуточных структур данных. Первичной такой структурой является «многослойное» пользовательское 2D-изображение и вектор параметров, полученных из форм. Изображение должно быть построено по строгим правилам, обусловленными ограничениями предметной области, что говорилось в параграфе о входных данных системы.

На предварительном этапе система получает и анализирует пользовательский ввод. Определяются центроиды плоских геометрических фигур – будущих комнат и коридоров. Определяется вид симметрии - радиальная или зеркальная.

На первом этапе генерируются стены, дверные и арочные проемы, потолки.

На втором этапе здание генерируется вверх. Структура с предыдущего этапа мультиплицируется на N этажей. Между этажами по желанию пользователя могут быть добавлены технические этажи, не имеющие планировки и предназначенные для прокладки коммуникаций.

На третьем этапе генерируются опорные конструкции: колонны, столбы. Тут могут быть использованы эвристики: какое расстояние должно быть от колонны до другой несущей конструкции в зависимости от прочностных характеристик материала стен, обычно, железобетона. Такие эвристики помогут сделать расположение опор визуально реалистичным.

На четвертом этапе учитываются параметры, заданные пользователем, связанные с оформлением крыши: плоская / с узором / произвольная. Крыша с узором может быть задана уравнением поверхности. Для произвольной накладывается функция шумов с максимумами в центроидах - так генерируются башни, надстройки, балконы.

На четвертом этапе генерируются детали интерьеров - лестницы, узоры на полах и стенах, декоративные вставки из различных материалов.

Изменение параметров генерации во время вывода. Пересмотр убеждений

В системах вывода знаний, где могут меняться переменные, возникает проблема, когда при новых вводных нужно извлечь «новое» знание в контексте той же задачи. Пересмотр убеждений (belief revision) – это исследовательский подход, направленный на изучение того, как меняются логические рассуждения, как разумные агенты пересматривают свои убеждения, и в каких ситуациях они это делают. Убеждение здесь – это связь между разумным агентом и утверждением.

В искусственном интеллекте пересмотр убеждений обычно происходит в контексте алгоритмов машинного обучения. Когда модель обучается на основе данных, она запечатлеет знания, основанные на представлениях о связях и закономерностях в данных, и формирует то, что мы и называем убеждениями. Однако, по мере получения новых данных, эти убеждения могут потребовать переоценки или коррекции.

При пересмотре убеждений в машинной системе возникает множество технических проблем, таких как экспоненциальная сложность и необходимость сохранения убедительности и непротиворечивости [6]. Сама задача автоматизации мышления является крайне комплексной междисциплинарной задачей [4], и даже является предметом изучения философии.

Под пересмотром убеждений понимается процесс модификации набора утверждений (убеждений), в котором было обнаружено противоречие. Известны 2 основных подхода к работе с противоречивыми знаниями. Первый из них (различные виды предположения о замкнутости мира, очертания Маккарти) связан с "ограничением" исходной базы знаний так, чтобы запретить вывод

некоторых заключений по умолчанию. Второй подход, принятый в системах поддержки истинности (СПИ), сводится к управляемому зависимостями бэктрэкингу.

Существуют различные алгоритмы пересмотра убеждений [4, 5]:

- AGM Contraction
- AGM Revision
- Base Contraction
- Belief Base Revision
- Belief Base Semi-revision
- Generalized AGM Postulates
- AGM Compliance
- Relevance Compliance
- Relevance and Partial Meet Contraction
- Generalized Postulates

В контексте описываемой системы, пересмотр убеждений заключается в перегенерации отдельных частей здания, если пользователь на каком-либо из этапов решил вернуться назад, и изменить одну из опций, например, если он поменял вид узора напольного покрытия, или отказался от возведения дополнительных этажей.

Помимо такого использования, пересмотр убеждений может быть использован для устранения противоречий при генерации.

Задача оценки количества материалов

Функция оценки количества материалов нужна для получения экономических параметров проекта на самых ранних этапах.

Для ее реализации достаточно выполнить первые 2 этапа генерации, описанных в параграфе, посвященному этапам генерации. Следующие этапы носят преимущественно декоративный характер.

Декорации – это прежде всего дополнительный материал, который, как правило, стоит дороже основного материала (материал стен, полов и т.д.), но обладает привлекательными внешними свойствами.

В дизайне интерьеров используется множество дополнительных материалов: зеркала, стекла, деревянные вставки, пластиковые плинтуса, и т.д. В экономическом смысле это выглядит как потребность в некотором дополнительном списке материалов, каждый из которых применяется ограниченно и согласовано с другими, так, как это описано айдентикой дизайн-проекта, т.е. выбранной пользователем стилистикой.

В зависимости от стилистики перечень этих дополнительных материалов и их пропорция к количеству основных материалов варьируются. Эти пропорции, которые можно рассматривать как вероятностные коэффициенты, определяются стилистическими пользовательскими параметрами.

Для апробации такой оценки была создана модель системы. С ее помощью был сгенерирован датасет, и на этих данных была построена модель линейной регрессии.

Простая модель системы

На вход модель системы принимает упрощенные схемы плана здания, содержащие комнаты, маршруты между ними, стены. Затем система генерирует трехмерную карту с размерами, заданными в условных единицах, и имитирует применение к полученному детерминированному результату случайные модификации, описывающие применение вариативных декораций к помещениям.

Схемы плана здания – это 2 изображения 32x32 «пикселя», описывающие в условных единицах расположение маршрутов для перемещения внутри здания и контуры комнат, по которым будут выстроены стены.

Помимо графического плана на вход модели поступают параметры:

- "floorHeight" – высота помещения. Параметр варьируется от 3 до 7 у.е., соответствующих соответственно 2,7 м для жилых помещений и 7 – для технических, под установку оборудования или коммуникаций;
- "numFloors" – число этажей;

- "hasTechnicalFloor" – флаг наличия технических этажей между основными. Под прокладку коммуникаций.

Статистически входные схемы можно описать двумя параметрами:

- "nPathsBlocks" – суммарная «длина» маршрутов, описывающая коридорное пространство;
- "nWallsBlocks" – суммарная длина стен.

Выходные параметры – это трехмерная карта помещения, аппроксимированная до структур, состоящих из блоков, имеющая манхэттенскую меру расстояния.

Эта трехмерная карта задается массивом со знаками типа блока:

Выходная карта имеет следующие статистические параметры, учитываемые в датасете:

- "P" – суммарное число блоков, описывающих участки пола, предназначенного для перемещения;
- "F" – суммарное число блоков, описывающих участки пола, предназначенного для установки оборудования или проживания персонала;
- "C" – суммарное число блоков, описывающих потолки;
- "CW" – суммарное число блоков, описывающих стены технических этажей;
- "W" – суммарное число блоков, описывающих стены помещений, предназначенных для установки оборудования или проживания персонала.

Данное представление выходной модели помещения позволяет абстрактно оценить затраты на возведение соответствующего проекта в материалах. Материал отражает функциональное предназначение части помещения: несущие конструкции, напольное покрытие, декорации. Количество разного вида

материалов выливается для пользователя прежде всего в стоимость реализации созданного им проекта.

В зависимости от типа материала определяется:

- Число материалов для декорации этого типа
- Случайная пропорция этих материалов

Описание анализируемых данных

С помощью программной модели был составлен датасет. Были подготовлены 8 схем помещений 32x32 у.е.^{2}, и для каждой было подготовлено по 100 вариаций дизайн-решения. Планы помещения, состоящие из двух слоев – контуры стен и маршруты перемещения – приведены в приложении 1 на рисунках 1 – 8.

Столбцы датасета описаны в параграфе выше.

Таблица 1. Фрагмент датасета.

№	N Paths Blocks	N Walls Blocks	Floor Height	Num Floors	Has Technical Floor	P	F	C	CW	W
0	231	94	3	4	0	924	3172	4462	0	1530
1	179	93	6	2	1	358	1690	3402	564	1403
2	320	84	5	2	0	640	1408	2200	0	1004
3	385	135	5	1	0	385	639	1127	0	832
4	159	201	6	1	0	159	865	1081	0	1578
5	303	154	5	2	1	606	1442	3314	564	1978
6	189	110	4	2	0	378	1670	2161	0	1180
7	367	177	6	2	0	734	1314	2287	0	2852

Подготовка данных к анализу:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Загрузка данных
df = pd.read_csv('dataset-2023-12-21T13_07_35.528Z.csv', encoding = 'latin-1', sep = ',')

# Вывод первых 5 строк таблицы
print(df.head().T)

df.info()

df.describe()
```

```
names = ["nPathsBlocks", "nWallsBlocks", "floorHeight", "numFloors", "hasTechnicalFloor", "P", "F", "C", "CW", "W"]
```

Вывод:

Таблица 2. Статистические данные колонок df.describe().

	N Paths Blocks	N Walls Blocks	Floor Height	Num Floors	Has Technical Floor	P	F	C	CW	W
count	800	800	800	800	800	800	800	800	800	800
mean	266,6	131	4,8	1,9	0,3	512,1	1444,9	2494,8	167	1423,1
std	82,8	40,3	1,4	1	0,4	341,2	848,5	1334,5	300,8	707,5
min	159	84	3	1	0	159	639	1075	0	298
25%	186,5	93,7	4	1	0	303	793	1153	0	864
50%	267	122,5	5	2	0	378	1314	2242	0	1296
75%	331,7	159,7	6	2	1	693	1730	3342,2	282	1831,7
max	385	201	7	5	1	1925	4325	6927	1410	4044

```
                    0    1    2    3    4
inputSchemeName     0    0    0    0    0
nPathsBlocks      231  231  231  231  231
nWallsBlocks       94   94   94   94   94
floorHeight         3    3    4    3    7
numFloors           4    4    1    3    1
hasTechnicalFloor   0    0    1    0    0
P                 924  924  231  693  231
F                3172 3172  793 2379  793
C                4462 4682 2223 3376 1108
CW                  0    0  282    0    0
W                1530 1481  490 1111  851

RangeIndex: 800 entries, 0 to 799
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   inputSchemeName   800 non-null    int64
 1   nPathsBlocks      800 non-null    int64
 2   nWallsBlocks      800 non-null    int64
 3   floorHeight       800 non-null    int64
 4   numFloors         800 non-null    int64
 5   hasTechnicalFloor 800 non-null    int64
 6   P                 800 non-null    int64
 7   F                 800 non-null    int64
...
 9   CW                800 non-null    int64
 10  W                 800 non-null    int64
dtypes: int64(11)
memory usage: 68.9 KB
```

Колонка с именем далее не нужна.

```
df = df.drop(columns=['inputSchemeName'])
```

Распределение различных групп переменных датасета:

```
sns.boxplot(df[["nPathsBlocks", "nWallsBlocks"]])
```

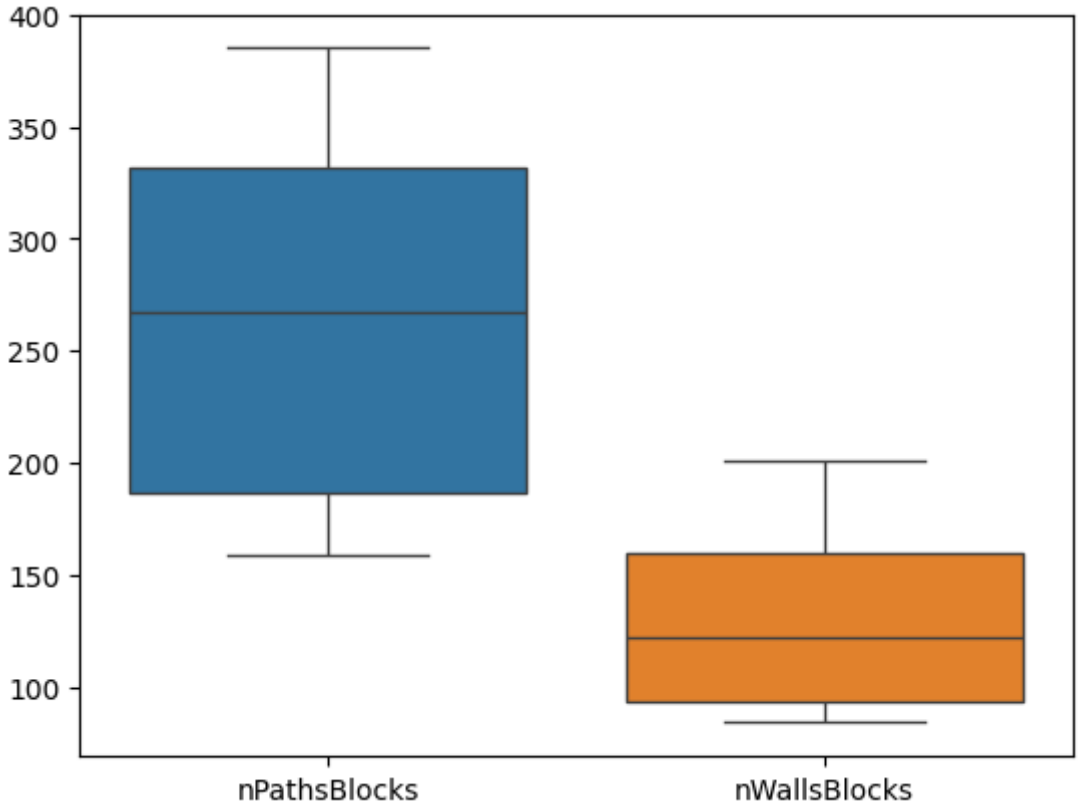

Рисунок 9. Диаграмма размаха для входных данных со схемы.

```
sns.boxplot(df[["floorHeight", "numFloors", "hasTechnicalFloor"]])
```

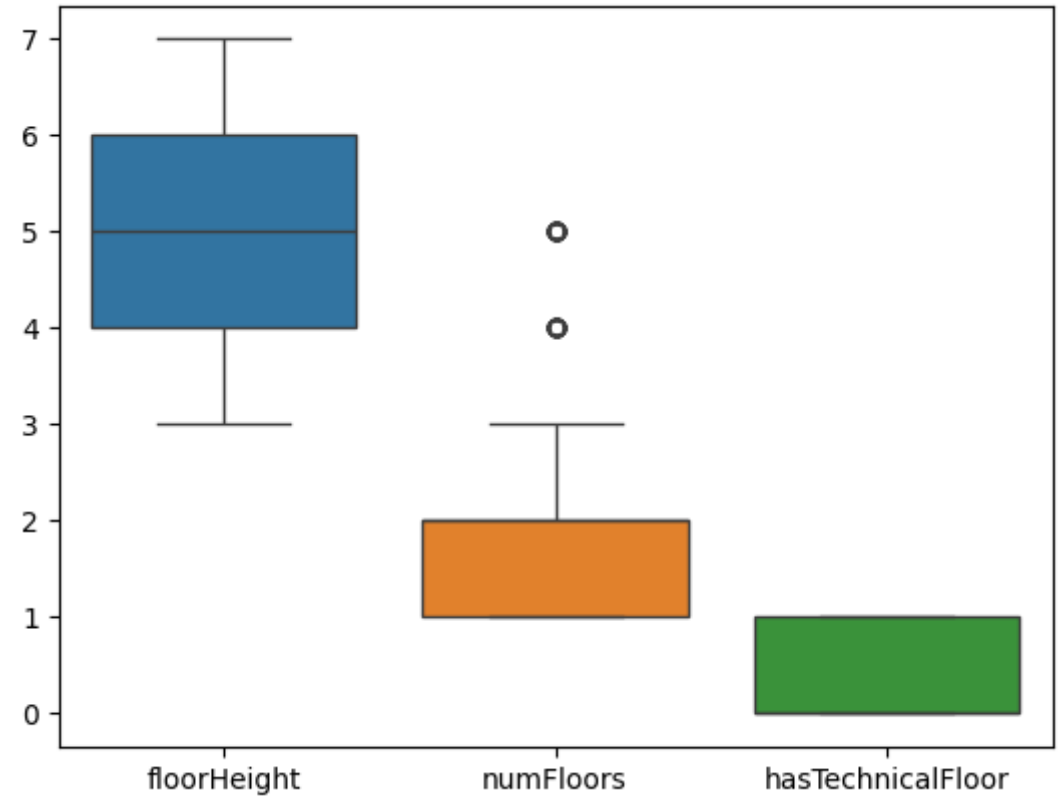

Рисунок 10. Диаграмма размаха для остальных входных данных.

```
sns.boxplot(df[["P", "F", "C", "CW", "W"]])
```

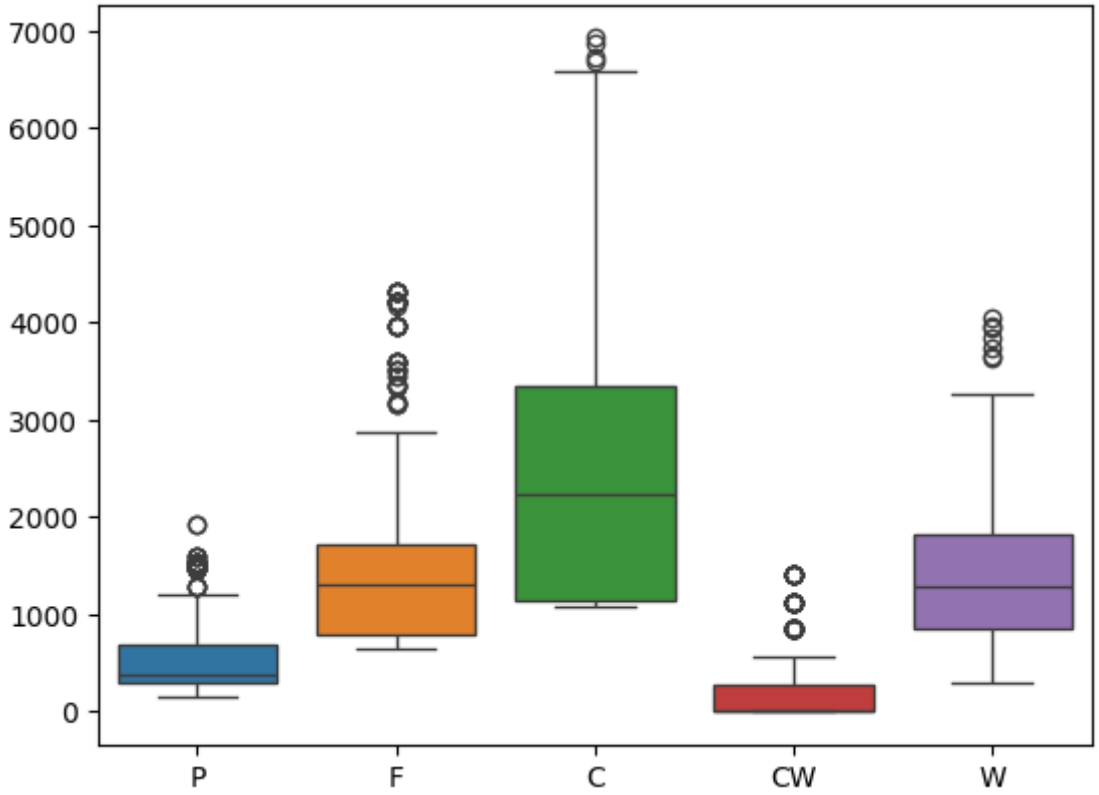

Рисунок 11. Диаграмма размаха для выходных данных модели.

Посмотрим, как коррелируют параметры.

```
df[df.columns].corr(method='spearman')
sns.heatmap(df.corr(method='spearman'), annot=True);
```

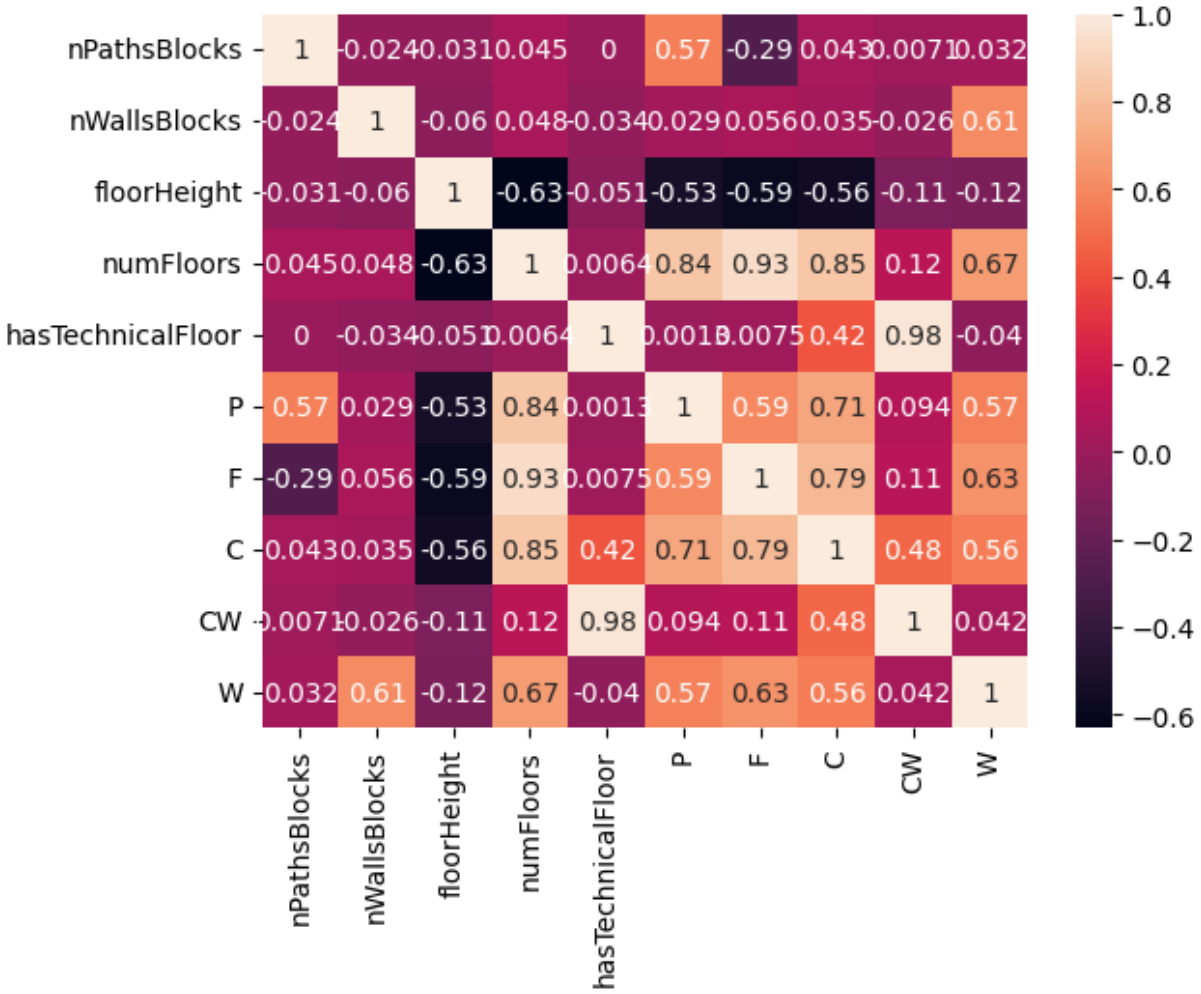

Рисунок 12. Тепловая карта матрицы корреляций.

Интерпретация значений коэффициента корреляции:

до 0,2 - очень слабая

до 0,5 - слабая

до 0,7 - средняя

до 0,9 - высокая

свыше 0,9 - очень высокая корреляция

Вполне ожидаемо коррелируют между собой число входных клеток пола и число выходных блоков пола (параметры nPathBlocks и P), число входных клеток, описывающих стены, и число выходных блоков стен (параметры nWallsBlocks и W), флаг наличия технического этажа и число выходных блоков стен технического этажа (параметры hasTechnicalFloor и CW), число этажей и число выходных блоков стен, полов и потолка (параметры numFloors и W, F, P, C).

Линейная регрессия

Для расчётов был загружен пакет statsmodels и объект OLS, благодаря которому можно легко рассчитать линейную регрессию. Для моделирования данные сохраняются в две переменные Y и X. Первая содержит зависимую переменную, вторая — независимые.

Далее создается объект модели, а потом с помощью метода fit вычисляется наилучшая регрессионная линия.

Создается модель линейной множественной регрессии, описывающая зависимость числа выходных блоков-стен от всех входных параметров "nPathsBlocks", "nWallsBlocks", "floorHeight", "numFloors", "hasTechnicalFloor".

```
import statsmodels.api as sm
from statsmodels.regression.linear_model import OLS

X = df[["nPathsBlocks", "nWallsBlocks", "floorHeight", "numFloors", "hasTechnicalFloor"]]
Y = df["W"]
X = sm.add_constant(X, prepend=False)

model = OLS(Y, X)
res = model.fit()
print(res.summary())
```

Вывод:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      W   R-squared:                       0.881
Model:                            OLS   Adj. R-squared:                  0.880
Method:                 Least Squares   F-statistic:                     1170.
Date:                Thu, 21 Dec 2023   Prob (F-statistic):               0.00
Time:                        18:35:08   Log-Likelihood:                -5534.3
No. Observations:                 800   AIC:                         1.108e+04
Df Residuals:                     794   BIC:                         1.111e+04
Df Model:                           5
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
nPathsBlocks               0.1514      0.105      1.439      0.150      -0.055       0.358
nWallsBlocks              10.9741      0.216     50.767      0.000      10.550      11.398
floorHeight              218.2082      7.939     27.487      0.000     202.625     233.792
numFloors                570.6061     10.266     55.584      0.000     550.455     590.757
hasTechnicalFloor          9.1207     18.819      0.485      0.628     -27.821      46.062
const                  -2213.3484     67.937    -32.579      0.000   -2346.706   -2079.990
==============================================================================
Omnibus:                        2.858   Durbin-Watson:                   1.832
Prob(Omnibus):                  0.239   Jarque-Bera (JB):                2.723
```

```
Skew:                          0.109   Prob(JB):                     0.256
Kurtosis:                      3.184   Cond. No.                  2.43e+03
...
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.43e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Во второй колонке указаны коэффициенты уравнения линейной регрессии. Подставляя их как коэффициенты уравнения, получаем формулу:

$$W \approx 0.1514 \times nPathsBlocks + 10.9741 \times nWallsBlocks + 218.2082 \times floorHeight + 570.6061 \times numFloors + 9.1207 \times hasTechnicalFloor - 2213.3484$$

Далее указаны статистические параметры по каждой переменной: среднеквадратичная ошибка и персентили.

Поле resid объекта модели res покажет отклонения (ошибку) по каждой переменной.

```
res.resid
```

По этим данным можно построить гистограмму ошибки модели.

```
plt.hist(res.resid, 100)
plt.title("Ошибка модели")
plt.grid(True, 'both')
plt.show()
```

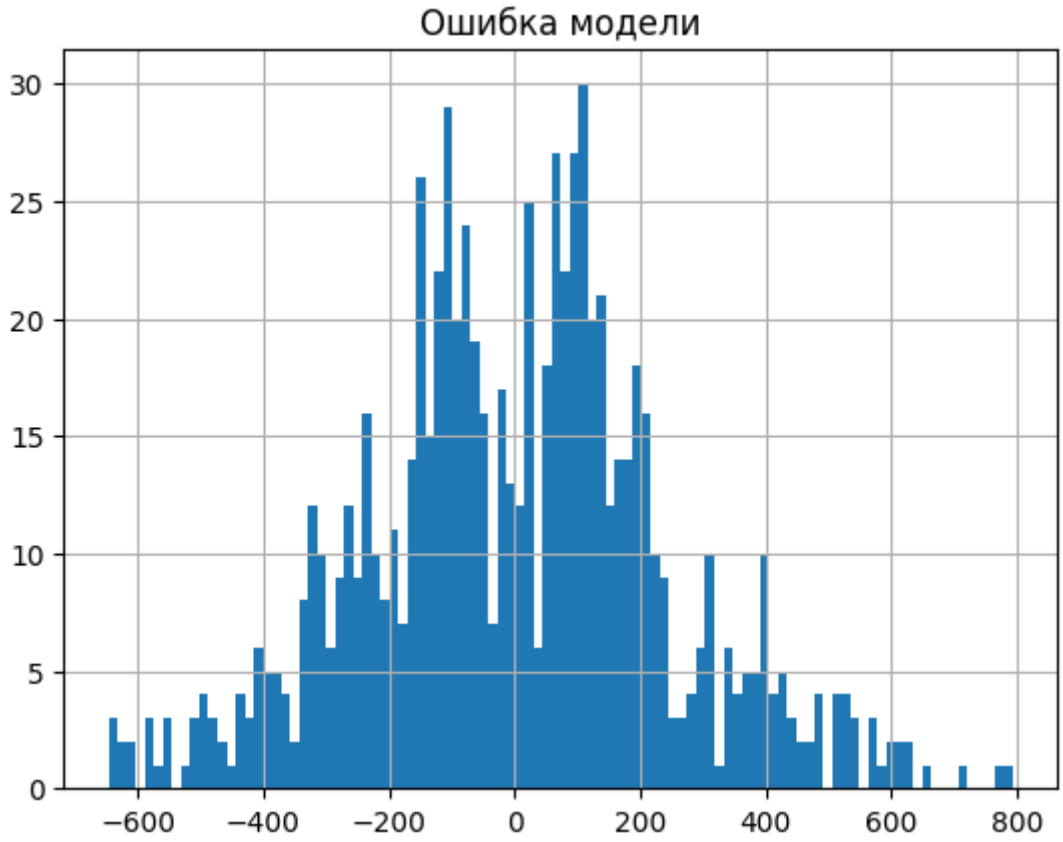

Рисунок 13. Гистограмма ошибки модели.

Полученная модель теперь может применяться для предсказания числа выходных блоков-стен от всех входных параметров без необходимости генерации трехмерной карты.

Выполним такое предсказание.

Пусть задан входной вектор

$$X_1 = (400, 100, 7, 2, 1)$$

```
X_for_prediction = pd.DataFrame({
  'nPathsBlocks': [400],
  'nWallsBlocks': [100],
  'floorHeight': [7],
  'numFloors': [2],
  'hasTechnicalFloor': [1],
})
X_for_prediction = sm.add_constant(X_for_prediction, has_constant='add', prepend=False)

res.predict(X_for_prediction)
```

Вывод: 1622.422074, что является адекватной оценкой.

Выводы

Была изучена проблематика задачи диалоговой генерации архитектурного дизайна зданий с использованием базы знаний. Были определены форматы входных и выходных данных модели и обозначены основные принципы и технические решения для такой системы, которые могли бы быть применены при ее реализации.

Кроме того, в ходе изучения, был указан порядок поэтапной генерации структуры, которая является представлением планировки трехмерного здания.

Также была написана упрощенная модель системы, демонстрирующая состоятельность концепции, и позволяющая показать принципы решения некоторых задач полноценной системы.

Литература

1. Kasabov N., Kozma R. Hybrid Intelligent Adaptive Systems: a Framework and a Case Study on Speech Recognition II Intelligent Systems (Гибридные интеллектуальные адаптивные системы), 1998.
2. Долинина О. Н. Классификация ошибок в базах знаний экспертных систем // Вестник СГТУ. 2010. №2с.
3. Дзен.Статьи. Канал Наука и образование «Забавные ошибки при генерации лиц искусственным интеллектом» [Электронный ресурс] — URL: https://dzen.ru/a/XnX-RmJ8rT_QLoC9 (Дата обращения: 21.11.2023).
4. Tennant, Neil, Changes of Mind: An Essay on Rational Belief Revision // Oxford. 2012. C.14
5. Moretto Ribeiro, Márcio, Belief Revision in Non-Classical Logics // Springer, 2013.
6. Козаченко Н. Критерии рациональности изменения убеждений: непротиворечивость // Логические исследования / Logical Investigations. 2010. Т. 16. С. 134-155.

Приложение 1. Планы помещений для составления датасета.

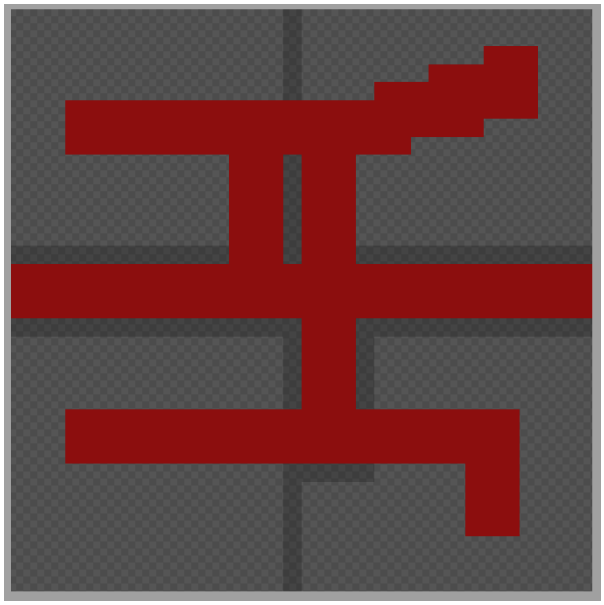

Рисунок 1. План помещения №1.

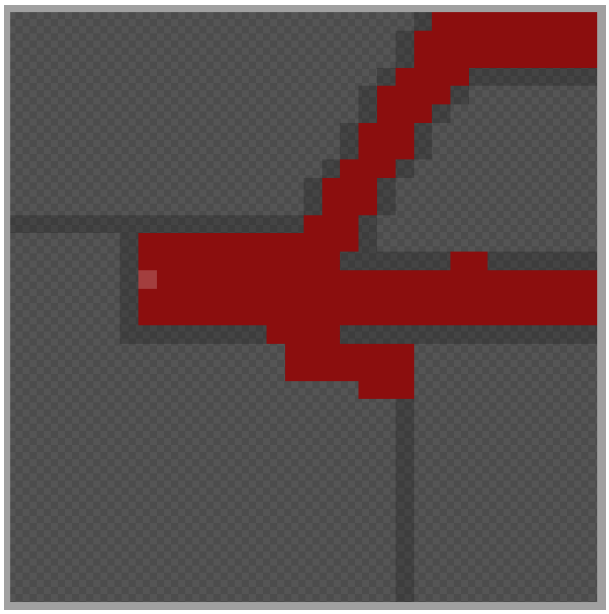

Рисунок 2. План помещения №2.

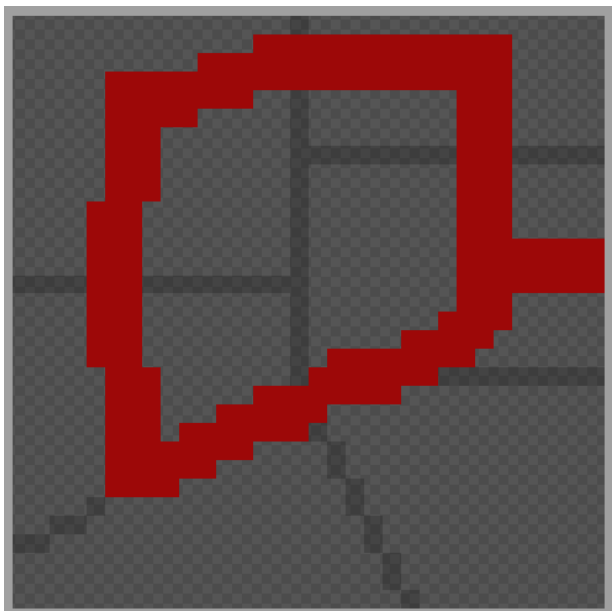

Рисунок 3. План помещения №3.

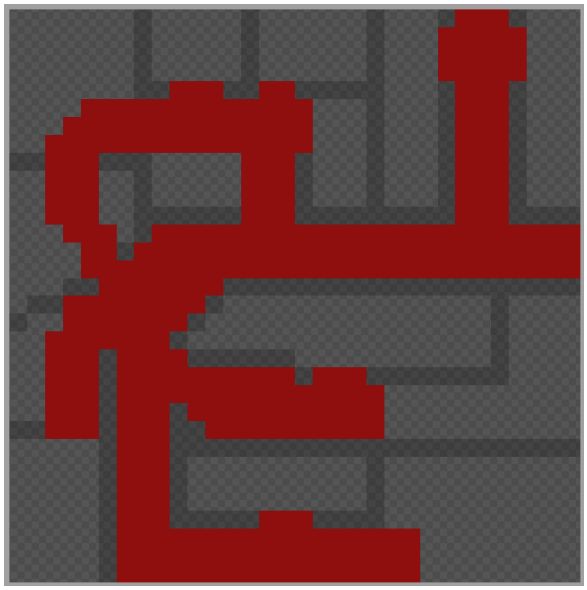

Рисунок 4. План помещения №4.

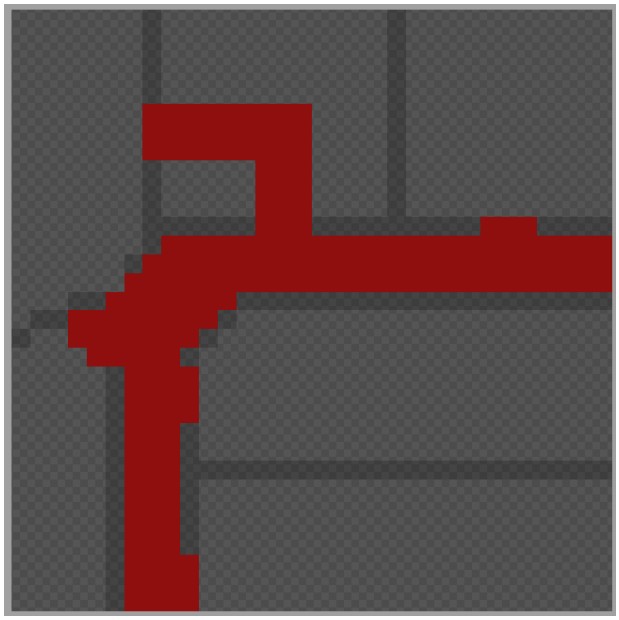

Рисунок 5. План помещения №5.

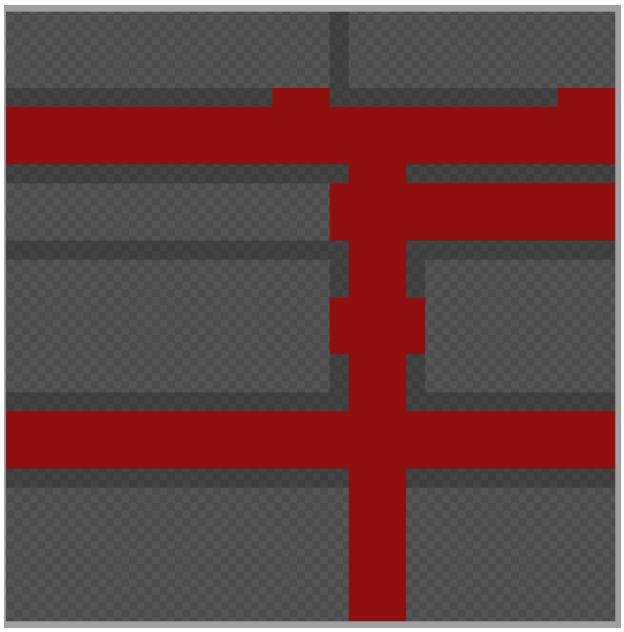

Рисунок 6. План помещения №6.

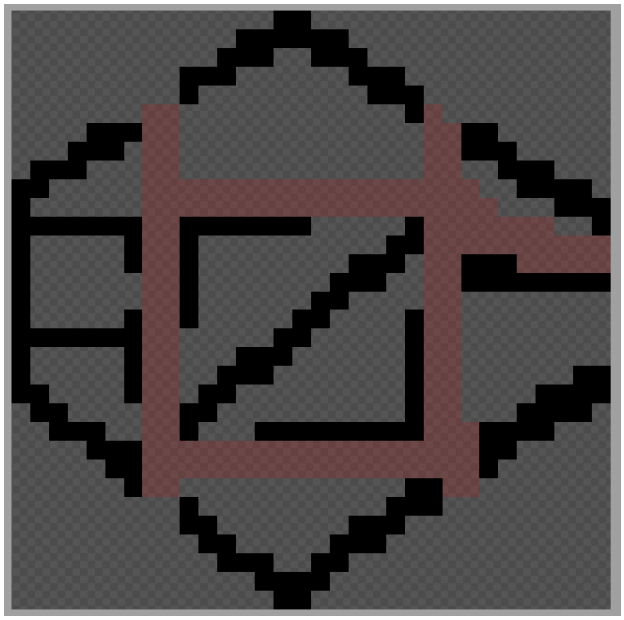

Рисунок 7. План помещения №7.

Рисунок 8. План помещения