Санкт-Петербургский государственный университет

**КУРШИН Максим Андреевич**

**Выпускная квалификационная работа**

**Процедурная генерация планов помещения**

Уровень образования:

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2015  
«Прикладная математика, фундаментальная информатика и программирование»

Научный руководитель:

доцент, кафедра компъютерных технологий и систем

кандидат ф.- м. н.

Коровкин Максим Васильевич

Рецензент:

Старший преподаватель,

кафедра технологий программирования

Стученков Александр Борисович

Санкт-Петербург

2019

**Содержание**

Введение…………………………………………………………………………3

Постановка задачи………………………………………………………………5

Обзор литературы………………………………………………………………6

Глава 1. Исследование………………………………………………………….8

* 1. Задача процедурной генерации помещения и термины…………...8
  2. Постановка критериев оценки сгенерированных помещений…....11
  3. Методы, основанные на древовидных картах………….………….16
  4. Метод ограниченного роста……………………………….………..21
  5. Выводы по главе…………………………………………………….27

Глава 2. Построение метода…………………………………………………....28

2.1 Эффективная структура данных…………………………...………..28

2.2 Синтезированный подход……………………………………………30

2.3 Основные сущности программы…………………………………….31

Выводы.…………………………………………………………………………37

Заключение………………………………………………………………………38

Список литературы……………………………………………………………..39

Приложение 1……………………………………………………………………40

Приложение 2…………………………………………………………………….41

ВВЕДЕНИЕ

Целью данной выпускной квалификационной работы является изучение методов процедурной генерации помещений, введение формальных критериев их оценки, а также составление алгоритма генерации плана помещения и его программная реализация на языке C# и игровом движке Unity3D, для быстрого автоматического создания игрового окружения.

Процедурная генерация (ПГК) – это набор алгоритмов и методов, используемых для автоматического создания контента: изображений, текстур, моделей, игровых объектов, сущностей и уровней.

Создание объектов методами ПГК – распространённая задача, появившаяся практически одновременно с медиа-развлечениями. Эти методы позволяют значительно упростить и ускорить процесс разработки тогда, когда требуется создать множество объектов с единообразными, но различными характеристиками, производство которых вручную займет значительные человеческие и денежные ресурсы, а иногда выполнить задачу можно исключительно методами процедурной генерации.

Характерным примером последнего является задача генерации уровней для игр жанра Rogue-like, где игровой уровень каждый раз обязан быть уникальным. В этом случае число уровней, которые необходимо составить стремится к бесконечности, а потому выполнить данную задачу без применения ПГК не представляется возможным.

Рассматриваемая в этой работе задача генерации помещения во многом схожа с задачей генерации уровней, так как имеет аналогичную цель – автоматическое создание игрового окружения для взаимодействия с игроком. Однако, такая задача имеет и свои особенности, например повышенное требование к реалистичности результата генерации. Если игровой уровень в первую очередь должен отвечать требованиям со стороны геймдизайна, то сгенерированное помещение в первую очередь должно выглядеть максимально схоже с реальным помещением.

Использование ПГК при разработке позволяет перекладывать большую часть рутинной и долгой работы на компьютер, что значительно сокращает время и, соответственно, бюджет, требуемый на разработку. Сейчас, когда контента в видеоиграх становится все больше и больше, а бюджеты доходят до сотен миллионов долларов – внедрение процедурной генерации как никогда актуально.

Постановка задачи

Первостепенной задачей данной выпускной работы является формулировка строгих критериев оценки результатов алгоритмов процедурной генерации помещений, так как достаточно строгих критериев не было предложено ни в одной из существующих работ. Каждый критерий должен давать возможность однозначно сравнить два метода, давая каждому из них простую численную оценку.

Затем с использованием введённых критериев и на основе анализа литературы, необходимо выделить из существующих решений самые эффективные, подробно проанализировать их, обозначив подходы, которые и сделали данные решения самыми эффективными.

В конце концов необходимо составить новый алгоритм генерации помещения и реализовать его. Алгоритм должен быть основан на самых эффективных существующих подходах, но предлагать усовершенствования, что-то, что качественно выделит его на фоне других работ.

Обзор литературы

Методы процедурного моделирования были предложены практически для каждого аспекта виртуальных миров, от ландшафтов до зданий. Обширный обзор методов процедурного моделирования можно найти в [1]. В некоторых работах, например в [2] рассматривается генерация города, с упором на фасад зданий, а не на их внутреннее строение.

В [3] предлагается основанный на графах метод создания помещений, в котором узлы представляют комнаты, а ребра соответствуют соединениям между ними. Этот граф преобразуется в пространственное расположение путем определения двухмерного положения для каждой комнаты. Для каждого узла, в зависимости от желаемого размера помещения, применяется определенное количество «давления», чтобы заставить его расширяться и заполнять оставшееся пространство здания. Данная работа предложила графы для решения задачи одной из первых, и многие работы ссылаются на нее, совершенствуя ее идеи. Одной из последних таких работ, а также одной из последних работ по теме процедурной генерации помещения за последнее время является работа [4], в ней предлагается подход, основанный на планарных графах. Так как данная статья является одной из самых новых, и так как в ней представлено достаточное количество примеров сгенерированных помещений, она будет использована для сравнения значений критериев других работ.

Продолжением идеи с графами является множество методов ([5]и [6] – лишь некоторые из них) основанных на Квадратированных Древовидных Картах, где на основе графа или дерева, строится древовидная карта. Схожий подход также представлен в [7], но основан уже на К-деревьях. Однако обладает схожим алгоритмом и схожими недостатками. Работа [6] является одной из последних работ с подобным подходом, и в ней он получает наибольшее развитие, поэтому при исследовании квадратированных карт, акцент будет сделан именно на данной работе.

Во всех этих методах графы используются и как тип входных данных, и как основа генерации комнат. Могут отличаться методы преобразования графа в помещение, тем не менее выходы данных алгоритмов схожи между собой визуально и обладают схожими недостатками, например ограниченным разнообразием форм помещений.

Однако работа [8] предлагает иной, не основанный на графах подход генерации комнат, построенный на крайне эффективном (с точки зрения критерия разнообразия) методе «ограниченного роста», который представляет огромный интерес для исследования и будет разобран в следующих параграфах данной дипломной работы. Данный метод также применяется в [9], но данная работа не предлагает никаких нововведений в метод, поэтому его анализ будет проводится по первоисточнику.

Также стоит отметить, что все приведенные статьи являются зарубежными, так как рецензируемых русскоязычных научных статей по данному вопросу обнаружено не было.

ГЛАВА 1. ИССЛЕДОВАНИЕ

В этом разделе будет рассмотрена общая постановка задачи генерации плана помещения на основе общих требований к методам процедурной генерации.

Будут введены и применены, до этого нигде не представленные методы формальной оценки реалистичности и разнообразия результатов алгоритмов генерации.

В итоге целью данной части дипломной работы является выделение самых гибких и эффективных методов из существующих решений, чтобы в дальнейшем использовать полученные знания для собственной реализации.

1.1 Задача процедурной генерации помещения и термины

Прежде чем двигаться дальше, требуется строго определить, что именно нужно получить при помощи алгоритма процедурной генерации. Поэтому, здесь и далее, под термином *помещение* будем понимать совокупность всех объектов, которые составляют или могут составлять часть здания, в том числе: комнат, коридоров и любых предметов, которые могут находится внутри этих комнат, например мебели и дверей. Соответственно, под задачей генерации помещения будем понимать задачу генерации некой совокупности данных объектов. Иногда в термин помещения будем также включать внешние стены здания, и еще реже ближайшую к зданию территорию. Однако чаще всего под объектами помещения будут пониматься комнаты, двери и окна, так как задача генерации внутреннего обустройства комнат, крайне обширна.

Теперь перейдем к непосредственно к задаче. Несмотря на обилие частных случаев, задача генерации помещения, как и любая другая задача процедурной генерации контента, должна удовлетворять ряду требований. В основном эти требования касаются результата алгоритма генерации.

В первую очередь, любое сгенерированное помещение должно удовлетворять свойству, которое в данной работе мы назовем *функциональностью*. А алгоритм, генерирующий функциональный контент, назовем функциональным, или удовлетворяющим свойству функциональности.

Функциональный контент – это такой контент, полученный алгоритмами процедурной генерации, который полностью удовлетворяет заданным в задаче требованиям. Так как в большинстве случаев ПКГ используется для видеоигр, то под функциональностью чаще всего понимают «играбельность», то есть возможность использования контента игроком по назначению. Например, процедурно сгенерированный уровень можно пройти, а процедурно сгенерированного противника можно победить. Соответственно в процедурно сгенерированном помещении можно попасть в любую комнату, выйти из помещения, а сами комнаты должны заполнять все пространство помещения.

Также, важным свойством алгоритма генерации является *разнообразие* генерируемого контента. Обладающий данным свойством генератор обязан каждый раз создавать различный контент. В нашем случае: алгоритм должен задавать случайным образом хотя бы одну характеристику генерируемого помещения. Чем больше характеристик задается случайным образом, и чем более широкий диапазон случайного выбора для этих характеристик, тем более разнообразным мы будем считать генерируемый контент.

Кроме того, алгоритм должен обладать свойством *надежности –* это свойство усиливает требование функциональности и означает, что ни при каких случайных значениях и допустимых исходных данных для результата генерации не нарушится свойство функциональности.

Последним и самым значимым будем считать свойство *правдоподобности* или *реалистичности*. Строго определить значение данного термина в общем случае практически невозможно, поэтому будем считать, что под правдоподобностью понимается создание генератором такого контента, который выглядит так, будто его создал человек, а не компьютер. Данное требование, однако, не может считаться строгим, поэтому в рамках данной работы, будем рассматривать предельный случай свойства правдоподобности, когда контент, созданный генератором, стремится к тому, чтобы выглядеть так, будто его создал человек, а не компьютер. В случае создания помещений, логичным будет требовать от генерируемого помещения максимальной схожести с реально существующими помещениями.

Теперь сформулируем задачу процедурной генерации помещения в общем случае. Требуется составить и реализовать алгоритм, удовлетворяющий свойствам функциональности, разнообразия, надежности и правдоподобности, результатом которого является некое помещение, соответствующее введенным начальным данным.

В других параграфах данной главы, мы рассмотрим существующие решения поставленной задачи и проанализируем подходы, которые в этих решениях используются.

1.2 Постановка формальных критериев оценки сгенерированных помещений и их применение

При анализе статей не было найдено ни одной работы по теме, которая бы вводила строгие критерии оценки сгенерированных помещений и которые могли бы быть использованы для сравнения различных подходов и методов.

В данной работе представлено два строгих критерия оценки помещений, основанных на свойствах ПГК: критерий реалистичности и критерий разнообразия.

Первый построен на идеи о близком к единице отношения характеристик сгенерированных помещений к реальным. Характеристики реальных помещений получены из интернет ресурса (1) на основе анализа более 50 планов реальных квартир. Их значения представлены в Приложении 1.

Второй на основе вариационных рядов и их коэффициента вариации. Здесь вычисляется набор характеристик для разных выходов программы при одних и тех же входных данных, из этих характеристик составляются вариационные ряды, а затем сами ряды анализируются.

Ниже приведены строгие формулировки критериев.

Критерий реалистичности:

Для оценки реалистичности будем использовать следующие 6 характеристик:

1. Отношение жилой площади помещения (жилые комнаты, гостиная) к нежилой (коридор, кухня, ванная).
2. Среднее значение соотношения длины к ширине прямоугольников, описывающих каждую комнату.
3. Соотношение площади коридора к площади всего помещения.
4. Среднее количество углов в комнатах
5. Среднее количество углов во внешних стенах помещения
6. Среднее количество дверей на пути в комнаты от входа в помещение

Через Cpn обозначим значение n-й характеристики процедурно сгенерированных помещений. Через Crn – реальных помещений.

Значение |1 - | - назовем критерием реалистичности процедурно сгенерированного помещения.

Чем ближе это значение к 0, тем более реалистичные помещения генерируются алгоритмом процедурной генерации.

Критерий разнообразия:

Для оценки разнообразия помещения, будем использовать следующие характеристики для каждой комнаты:

1. Координаты центра описанного прямоугольника комнаты: (X,Y) – координаты углов комнаты

()

1. Координаты углов описанного прямоугольника комнаты, относительно центра этого прямоугольника.

(maxX; maxY)

(minX; maxY)

**Комната**

(maxX;minY)

(minX; minY)

Рисунок 1.2.1. Описанный прямоугольник комнаты с обозначением координат углов

Вычисление координат углов относительно центра на примере левого нижнего угла:

(; minY - )

3.Количество углов в комнате

Составим вариационные ряды для каждой характеристики каждой комнаты для N выходов алгоритма генерации, при одинаковых входных данных:

- количество комнат и их типы (гостиная, кухня и так далее)

-общая площадь помещения

Для каждого такого ряда вычисляем коэффициент вариации Cv = σ / M, где σ – среднеквадратическое отклонение ряда, а М – его среднее арифметическое.

Далее считаем среднее арифметическое коэффициентов вариации всех рядов. Результат назовем показателем разнообразия определенных входных данных. Среднее арифметическое показателей разнообразия различных входных данных и будем называть критерием разнообразия процедурно генерируемых помещений. Чем больше полученное значение, чем более разнообразный генерируется контент.

Выбор характеристик для критериев:

Одним из главных преимуществ приведенных критериев является возможность использовать для них любые характеристики помещений, которые будут наилучше отражать цель исследования, что позволяет модифицировать и оптимизировать критерии.

Однако, в данной работе определенные характеристики, в частности для критерия разнообразия, были выбраны намерено. Большинство рассмотренных работ варьировали форму и положение комнат внутри помещения, а структура входных данных предполагала заранее определенные соединения дверьми. Поэтому для критерия разнообразия и были выбраны такие характеристики. Первая отражает вариацию положения комнаты, вторая ее формы. Чтобы вариация положения не учитывалась дважды - координаты углов комнаты, а точнее ее описанного прямоугольника рассчитываются относительно центра этого прямоугольника. Применяются характеристики именно описанного прямоугольника, а не самой комнаты, так как количество углов в комнате может также варьироваться. Последнее учитывается третьей используемой характеристикой. Для критерия реалистичности характеристики выбирались по схожему признаку, однако учитывалось, чтобы характеристики не зависели от масштаба плана, площади помещения и кол-ва его комнат.

С применением приведенных критериев оценивались результаты различных подходов процедурной генерации помещения. Оценка производилась на основе приводимых в работе изображений сгенерированных помещений, а вычисление необходимых координат проводилось при помощи пиксельных координат. Для расчёта критериев использовались все приведенные в статье примеры работы описанного алгоритма, в среднем от 6 до 15 сгенерированных помещений. Стоит отметить, что не во всех изученных работах присутствовало достаточное количество примеров выхода алгоритма, однако в большинстве случаев удавалось находить статью со схожим подходом, в которой при этом присутствовало необходимое количество результатов. Также нельзя не сказать о том, что для большинства работ значения критериев могут оказаться немного завышенными, так как предполагается, что авторы приводят наиболее удачные примеры работы своего алгоритма. Теперь рассчитаем значения критериев для основных работ, обозначенных в обзоре литературы.

Для удобства критерий реалистичность обозначим через R, а критерий разнообразия через D. Работа [4] основана на графах, и в ней приведено множество результатов работы метода, что позволяет в полной мере применить критерии. Кроме того, это одна из новейших работ, поэтому вычисление значения критериев для данного метода интересно в первую очередь для сравнения с ним значений других методов. Для метода, основанного на планарном графе: D = 16.161, R = 0. 247.

Данные значения нельзя назвать выдающимися, однако, как и было отмечено ранее, и сам метод не вызывает большего исследовательского интереса. А вот работа [6], в которой совершенствуются методы, основанные на древовидных картах, показывает впечатляющие результаты при вычислении критерия реалистичности: R = 0.063. Из определения критерия следует, что столь малое значение, указывает на то, что генерируемые помещения практически полностью совпадают с реалистичными. Однако, разнообразие данного метода впечатляет не очень сильно: D = 19.148. Очевидно, что это связано с практически всегда квадратными комнатами.

А вот метод [8] демонстрирует прямо противоположную ситуацию: R = 0.232, D = 29, 764. Это лучший показатель разнообразия из всех работ.

Итак, удалось выделить два лидирующих метода по двум критериям. Теперь, необходимо подробно разобрать эти методы, понять, что позволяет им демонстрировать столь впечатляющие значения, и попробовать применить полученные знания, для реализации алгоритма одинаково эффективного по обоим критериям.

1.3 Методы, основанные на древовидных картах

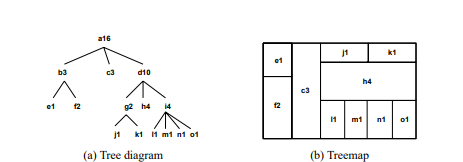


Рисунок 1.4.1. Пример построения обычной древовидной карты

При изучении существующих работ выяснилось, что очень многие решения (например, [5] и [6]) берут за основу Squarified Treemap - Квадратированные Древовидные карты\*. Такие методы позволяют быстро и эффективно заполнять прямоугольное помещение комнатами на основе простой древовидной карты. Данный метод подробно представлен в [10], здесь мы кратко разберем его основную идею.

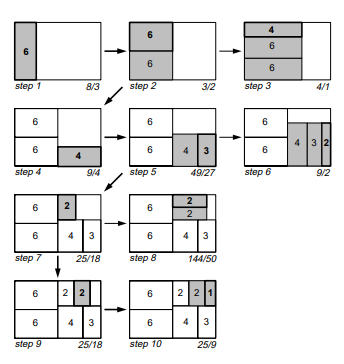


Рисунок 1.4.2. Принцип построения Квадратированной Древовидной Карты

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*- возможен неточный перевод

Древовидные карты применяются для наглядной визуализации древовидных структур, например - файловых систем. В [10] разбираются недостатки классического подхода древовидных карт в различных крайних случаях и предлагается иной метод их построения. Несмотря на то, что основной целью алгоритма остается визуализация древовидной структуры, метод вполне можно использовать для генерации помещения.

Сам алгоритм очень простой и представлен на рисунке 1.4.2. Прямоугольник заполняют меньшими прямоугольниками заданной площади так, чтобы соотношение их сторон не превышало заданных границ. Прямоугольники генерируются горизонтально (или вертикально) пока соотношение сторон остается приемлемым. Когда оно выходит из допустимых границ, алгоритм возвращается на шаг назад и запускается рекурсивно для оставшейся свободной зоны.

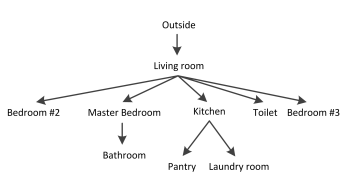


Рисунок 1.4.3. Иерархическое дерево будущего помещения

Данный принцип лег в основу многих работ, посвященных процедурной генерации планов помещения, однако особый интерес представляет статья Майсама Мирахмади и Абдалла Шами “A Novel algorithm for real-time procedural generation of building floor plans” [6]. В классический подход, который был заложен еще в статье [5], авторы внесли качественные изменения, которые значительно повысили реалистичность генерируемого помещения. Например, они оптимизировали процесс генерации коридора, минимизировав его площадь.

Входными данными для данного алгоритма является площадь заполняемого помещения и комнаты, которые нужно сгенерировать. Комнаты подаются в виде древовидной структуры, изображенной на рисунке 1.4.3. Иерархия в древе имеет важное значение при генерации, например коридор генерируется так, чтобы с гостиной были соединены комнаты из третьего уровня дерева. Площадь комнат определяется случайным образом на основе их типа: гостиная, спальня, кухня и так далее. Затем, по иерархическому дереву строится квадратированная древовидная карта, которая и является сгенерированным помещением. Площади комнат и их положение задаются случайным образом на основе типа комнаты.

Далее начинается построение коридора, представляющее наибольший исследовательский интерес. Коридор строится таким образом, чтобы соединить гостиную со всеми остальными основными комнатами.

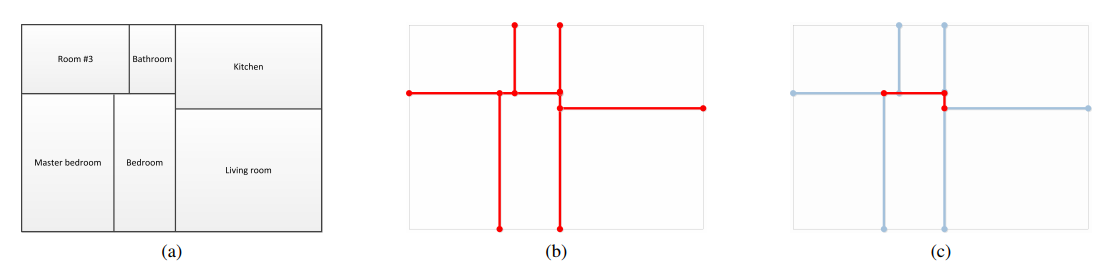


Рисунок 1.4.4.Поиск основы для построения коридора

Алгоритм:

1. Стены всех комнат, которые должны соединяться коридором, кроме внешних стен здания, заносятся в граф (рисунок 1.4.4 (b)).
2. Из этого графа исключаются все ребра, соединяющиеся с вершиной со степенью один (рисунок 1.4.4 (c)).
3. В получившемся графе ищется кратчайший путь, соединяющий все комнаты, чтобы минимизировать площадь будущего коридора.
4. На основе получившегося графа строится коридор, к нему отходит часть пространства соседних комнат. Позиция коридора выбирается так, чтобы длина стен, которыми он соединяется с комнатами, была не меньше минимально допустимой (чтобы была возможность поставить в этом месте дверь). Кроме того, коридор не должен менять форму комнат так, чтобы они становились недоступными. Коридор, который удовлетворяет всем указанным требованиям называется оптимальным. Если таких коридоров несколько - выбирается любой из них. Процесс поиска оптимального коридора показан на рисунке 1.4.5.

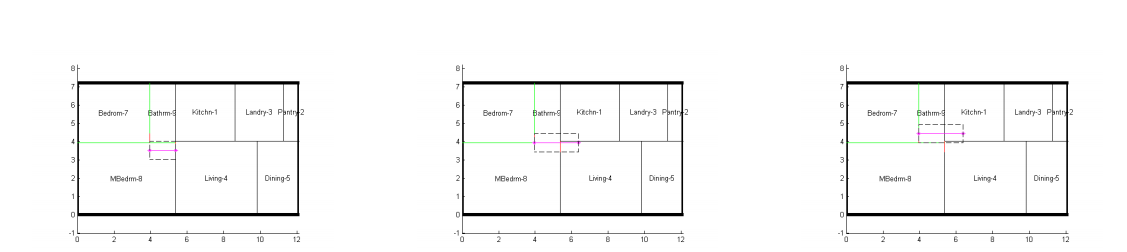


Рисунок 1.4.5. Выбор оптимального коридора

В заключительной стадии алгоритм генерирует двери и окна в помещении. Основой для генерации дверей также является иерархическое дерево (рисунок 1.4.3), однако авторы уточняют, что при необходимости можно добавить дополнительные проходы между комнатами. Сами двери генерируются случайным образом в стене между теми комнатами, между которыми должен быть проход согласно иерархическому древу или специальным условиям.

Окно генерируются случайным образом на внешних стенах, исключая участки, соединяющиеся с комнатами, в которых окон быть не должно (например, в ванной комнате или кладовке).

В разобранной статье предлагается ряд важных и крайне эффективных подходов. Самым существенным будем считать оптимизированный алгоритм генерации коридора. Данное решение является уникальным для работ, изученных в ходе нашего исследования, и потому представляет особый интерес для реализации. Не менее эффективным и интересным является и подход с иерархическим деревом, так как оно представляет собой одновременно и визуально понятную структуру для легкого использования в качестве входных данных, а также многократно применяется в алгоритме генерации.

Однако данное решение, как и многие другие, основано на Квадратированных Древовидных Картах, что накладывает серьезные ограничения на гибкость алгоритма, самым серьезным из которых является возможность генерировать исключительно прямоугольные помещения. Однако, такой подход предоставляет строгий контроль над соотношением характеристик комнат. Это, а также перечисленные выше преимущества алгоритма, позволили ему продемонстрировать лучшие значение критерия реалистичности.

1.4 Метод ограниченного роста

Рассмотрим работу Нидерландской команды из Делфтского университета технологий и Нидерландской организации прикладных научных исследований под названием “A constrained growth method for procedural floor plan generation”, написанную в 2010 году [8].

В этой статье рассматривается один из методов генерации помещения, в том числе предлагается интересный метод для генерации комнат, который и представляет для исследования наибольший интерес.

В начале статьи авторы формулируют задачу и рассуждают о целесообразности своего исследования, рассматривают примеры видеоигр, где процедурный подход к наполнению мира контентом оправдан. Далее дается общий обзор некоторых существующих решений, среди них многие из тех, что были рассмотрены нами выше, выделяются сильные и слабые стороны решений, а также отмечается какие из них легли в основу нового исследования.

Первое, что стоит отметить — это используемая в работе нидерландской команды структура данных - двумерная сетка (матрица). Такой подход, по мнению авторов является наиболее близким к тому, как выглядят настоящие архитектурные планы (рис. 1.5.1). Более того, авторы статьи утверждают, что данная структура данных является оптимальной, а также позволяет работать с любыми внешними стенами. Однако, у того подхода есть несколько недостатков, о которых авторы не упоминают. Во-первых, размер комнат может быть задан лишь с точностью до размерности сетки, а уменьшение размерности сетки приведет к увеличению требуемой для хранения структуры данных памяти. Во-вторых, если помещение не прямоугольное, то часть элементов в матрице не будут задействованы при работе алгоритма, так как будут принадлежать к пространству за внешними стенами, а комнаты в этой зоне не генерируются. Из-за данной особенности структуры данных количество используемой памяти возрастает еще сильнее, когда на вход программе подаются внешние стены сложной формы. При этом, у данной структуры данных практически нет значимых преимуществ, кроме сходства с реальной архитектурной сеткой, однако данное преимущество не играет никакой роли в генерации помещения для видеоигры. Во-второй главе мы предложим более эффективную, на наш взгляд, структуру данных, не зависящую от предъявляемой к помещению точности.

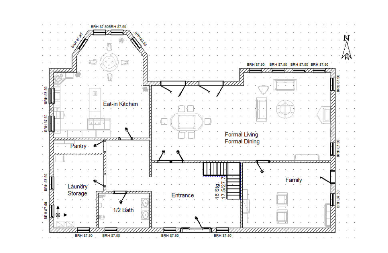


Рисунок1.5.1. Пример реального архитектурного плана помещения

Интересным является решение о введении иерархии в генерацию, такой же как в [5]. Помещение может быть разбито на N зон, к каждой из которых алгоритм применяется отдельно, это позволяет создавать непересекающиеся группы комнат, например личная и гостевая зоны.

Входными данными, для данного алгоритма являются:

* внешние стены помещения, задающиеся как элементы столбцов и строк матрицы
* размерность элементов сетки
* Список комнат и привязанный к каждой комнате желаемый размер, а также то, к какой зоне привязана комната
* Правила соединения комнат (т. е. какие комнаты должны находится рядом друг с другом и между какими комнатами должны сгенерироваться двери)

На выходе алгоритм должен сгенерировать помещение, полностью заполненное заданным количеством комнат и двери между этими комнатами.

Теперь рассмотрим сам приведенный в статье алгоритм процедурной генерации.

В первую очередь элементам матрицы, находящимся за пределами внешних стен, присваивается значение 0, а всем элементам внутри помещения 1. Теперь внутри помещения случайным образом выбирается несколько точек, которые станут заготовками будущих комнат. Попробовав различные варианты, авторы статьи пришли к выводу, что комнаты, наиболее реалистичные комнаты, получаются тогда, когда их точки-заготовки размещаются не около внешних стен, а на расстоянии, связанном с характеристикой комнаты желаемый размер. Так, чтобы в квадрате с центром в точке-заготовке были исключительно значения равные единице. Когда одна точка сгенерирована, небольшая область вокруг нее заполняется нулями. Таким образом исключается генерация двух точек слишком близко друг к другу. Размещение точек также учитывает правила соединения комнат и генерирует рядом соответствующие точки.

Когда точки расставлены начинается применение метода ограниченного роста, который, как уже было отмечено, и представляет наибольший интерес.

Алгоритм:

1. На основе точек заготовок создаются прямоугольные комнаты, в соответствии с характеристикой желаемый размер. Рост начинается с комнат, желаемый размер которых наибольший. Соотношение сторон комнаты задается случайным образом, но так, чтобы сохранялось требование правдоподобности.
2. Далее каждая комната по очереди увеличивается в размерах сохраняя прямоугольную форму, пока это возможно.
3. Затем комнаты начинают расти в форме буквы L, пока это возможно.
4. На конечном этапе алгоритм анализирует сетку в поисках свободных участков и привязывает их к наиболее подходящей комнате.

Работа алгоритма проиллюстрирована на рисунке 1.5.2.

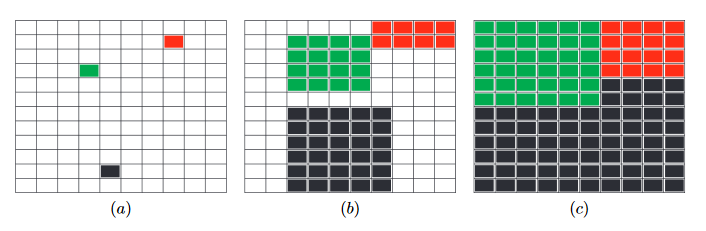


Рисунок 1.5.2. Пример генерации прямоугольного помещения с тремя комнатами методом ограниченного роста

Представленный в статье алгоритм обладает рядом заметных преимуществ, особенно перед более классическим подходом с древовидными картами, как в [6]. Во-первых, подобный алгоритм может создавать комнаты любой (не только прямоугольной) формы и работать с любыми (не только прямоугольными) помещениями, что значительно повышает разнообразие форм комнат внутри помещения, из-за чего данный алгоритм и стал лучшим по критерию разнообразия.

Когда все комнаты сгенерированы и все пространство помещения заполнено остается только сгенерировать двери, основываясь на входных данных о соединении комнат и принципах достижимости в графах. Двери при этом генерируются примерно в центре стены между комнатами, которые нужно соединить, с небольшим случайным смещением.

В заключительной части авторы обсуждают сильные и слабые стороны своего решения, а также пути его улучшения. Они отмечают, что выбранная структура данных, основанная на прямоугольной сетке, совершенно не подходит для помещений с непрямыми углами. Кроме того, нидерландская команда считает, что можно улучшить алгоритм генерации дверей, если создавать их не в случайном месте, а так, чтобы обеспечить наиболее оптимальный маршрут между комнатами.

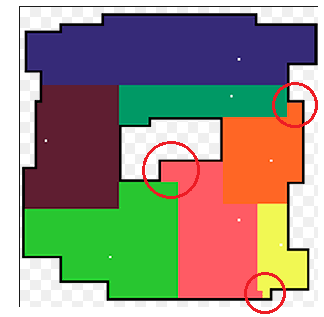


рис. 1.5.3 Пример небольших ответвлений в комнатах, возникающих в результате применения метода ограниченного роста.

(Результат работы сторонней реализации описанного метода из свободного источника.)

Не смотря на наличие некоторых недостатков, представленный метод ограниченного роста является крайне мощным и эффективным инструментом генерации разнообразных комнат, позволяя создавать множество не похожих друг на друга помещений. При этом у алгоритма практически нет серьезных ограничений, какие есть у методов на основе древовидных карт. Существенным же недостатком данного алгоритма является шанс появления небольших ответвлений от комнат, которые выглядят не естественно. Однако, такие ответвления в большинстве случаев появляются только, если внешние стены помещения представляют сложную фигуру, и в этом случае сгенерировать помещение без данных небольших фрагментов практически невозможно.

Недостатками работы нидерландской команды является применяемая структура данных, которую можно значительно улучшить, повысив гибкость и существенно уменьшив нагрузку на память, что и описывается в главе 2. Кроме того, не рассматривается способов эффективной генерации коридоров, которые очень часто являются важным элементом помещения.

Несмотря на все перечисленные выше недостатки метод ограниченного роста обладает рядом неоспоримых преимуществ, таких как гибкость применения и разнообразие генерируемого контента, именно поэтому этот метод был выбран основой нашей реализации алгоритма процедурной генерации.

1.5 Выводы по главе

Мы рассмотрели несколько подходов к решению задачи процедурной генерации планов помещения. Затем, с использованием введенных критериев были обозначены самые эффективные, и, соответственно, представляющие наибольший исследовательский интерес, работы по теме. Теперь необходимо выделить те методы, которые во время анализа были отмечены как наиболее эффективные и наиболее удовлетворяющие предъявляемым к задаче процедурной генерации требованиям, чтобы в итоге применить данные подходы в нашей реализации.

Очевидно, что метод ограниченного роста является наиболее разнообразным согласно критерию, благодаря практически полному отсутствию ограничений к применению. При этом развитие метода древовидных карт в работе [6], благодаря тому, что оно опиралось на сторгую структуру входных данных и предлагало оптимизированный и реалистичный подход к генерации коридора, важной части любого жилого помещения, показало себя как самое реалистичное по критерию решение.

В следующей главе, полученный исследовательский материал будет применен на практике с целью реализовать синтетический подход, одинаково эффективным по обоим критериям. При этом стоит ожидать, что реалистичность у синтетического метода будет немного ниже, чем у древовидных карт, так как метод ограниченного роста может вносить излишнее, нереалистичное разнообразие.

Глава 2. Построение метода

В данной главе мы применим полученные в процессе исследования знания, чтобы получить новый синтезированный алгоритм. Основой этого алгоритма станет метод ограниченного роста, так как он показал наилучшее значение по критерию разнообразия. В дополнение к нему будет использован метод генерации коридора, а также модель входных данных и генерация дверей из статьи [6], так как сгенерированные помещения именно из этой статьи показали наилучшие значения по критерию реалистичности. Кроме того, будет использована новая структура данных, обеспечивающая алгоритм дополнительными преимуществами.

2.1 Эффективная структура данных

Одним из главных недостатков алгоритма из статьи [8] является недостаточно гибкая и недостаточно оптимизированная структура данных – прямоугольная сетка. Выделим ее основные недостатки:

* С повышением требуемой точности будет повышаться количество элементов сетки, а значит и расход памяти, и время на выполнение алгоритма.
* Невозможность использовать структуру данных для создания не прямых углов, что ограничивает потенциал для расширения функционала алгоритма.
* Генерация помещения сложной формы создаст внутри сетки неиспользуемое пространство, что вызывает неоправданное повышение расхода памяти и скорости работы алгоритма.

Теперь предложим крайне простую структуру данных, решающую все перечисленные проблемы.

Каждая комната (а также само помещение) задается в виде набора точек на плоскости – координат ее углов в декартовой системе координат (X;Y). Точки упорядочены так, чтобы две соседние формировали стену комнаты. При этом важно, чтобы список не формировал двух подряд идущих горизонтальных или вертикальных стен.

Легко заметить, что такая структура данных действительно избавлена от вышеописанных проблем:

* Объем используемой памяти зависит только от количества комнат и количества их углов, но не от требуемой точности построения. Кроме того, в большинстве случаев количество всех углов всех комнат значительно уступает количеству элементов сетки.
* Если от программы потребуется генерировать помещения с непрямыми углами, достаточно будет просто хранить вместе с каждой точкой угол, под которым в ней соединяются стены.

Генерация помещения с непрямыми углами тем не менее не будет рассмотрена в данной дипломной работе, так как она порождает множество сложных в решении задач, однако возможность использование приведенной структуры данных для этой цели несомненно является преимуществом.

Также данная структура данных все еще отлично подходит для применения метода ограниченного роста. Доступное пространство для роста вычисляется как расстояние до ближайшей стены другой комнаты, или внешней стены. Сам рост осуществляется за счет смещения соответствующих углов комнаты в определенном направлении. Также, при добавлении новых точек (когда формируются L-образные или более сложные комнаты), необходимо, следить за сохранением упорядоченности их списка. Для гарантии последнего, будем применять алгоритм построения минимальной оболочки множества точек, в виде замкнутой ломаной линии, у которой в каждой точке два отрезка соединяются под прямым углом. В рамках данной работы назовем такую оболочку комнатой, а отрезок, соединяющий две точки стеной комнаты.

Необходимого результата позволит достичь нижеописанный алгоритм. Пусть имеется набор точек, необходимо отсортировать этот набор так, чтобы две последовательные точки формировали стену комнаты, если сделать этого не получается, будем считать набор точек некорректным. На практике некорректным может оказаться только набор входных точек. В алгоритме крайне важна очередность проверок, если ее не соблюсти, отсортированный список построится некорректно.

Алгоритм:

1. Находим точки с наименьшим значением по Х, а затем среди них точку с наименьшим значением по Y. Заносим данную точку в новый список и отмечаем, как текущую.
2. Ищем точку, ордината которой равна ординате текущей точки. Если таких несколько в первую очередь выбираем ближайшую сверху, если сверху точек нет - ближайшую снизу, добавляем ее в список и отмечаем, как текущую.
3. Далее ищем точку, абсцисса которой равна абсциссе текущей. Если на прошлом шаге взяли точку сверху, то в первую очередь отдаем предпочтение ближайшей левой точке, в противном случае – ближайшей правой.
4. Если на каком-то шаге не удается найти точку – список некорректный.
5. Если в неотсортированном списке еще остались точки – возвращаемся на шаг 2.

Не смотря на сложность алгоритма, он гарантирует, что две соседние точки будут представлять стену (вектор) комнаты и что все такие вектора будут отсортированы по часовой стрелке. (рис. 2.1.1)

Кроме того, теперь для каждой стены можно легко узнать направление внутрь помещения, просто повернув вектор стены по часовой стрелке, что очень важно для решения поставленной задачи.

Рисунок 2.1.1. Разбиение помещения на прямоугольники

2.2 Синтезированный подход

В качестве входных данных для программы будем использовать иерархическое дерево помещения, как в подходе с Древовидными картами, через которое задаются сами комнаты и логика их соединения, так как данная структура входных данных оказалась крайне эффективной и универсальной, задающей логику построения дверей и генерации коридора. Также входными данными являются внешние стены помещения в виде набора точек.

В первую очередь внутри помещения генерируются точки-заготовки будущих комнат. Для этого разобьем изначальное помещение на набор прямоугольников следующим образом. Через каждую угловую точку проведем горизонтальную линию, если линия проходит внутри помещения и пересекается со стеной не в угле, то мы используем эту линию, для разбиения на прямоугольники.

Точки генерируются внутри таких прямоугольников, на определенном расстоянии от стен и от других точек-заготовок.

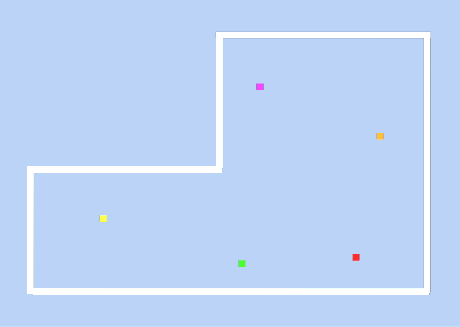
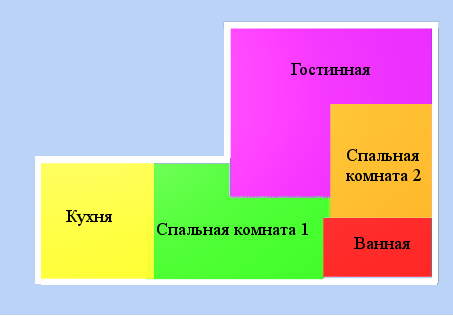


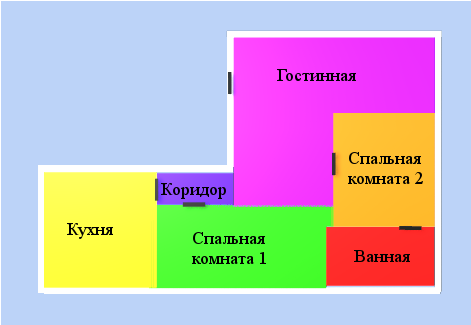
Рисунок 2.2.1. Точки заготовки метода

Далее применяется метод ограниченного роста, как описано в параграфе 1.5. (рис. 2.2.2)



*Рис.2.2.2. Окончание работы метода ограниченного роста*

После этого генерируется коридор, как описано в параграфе 1.4, на основе входного дерева, и на его же основе между комнатами создаются двери. (рис. 2.2.3)



*Рис. 2.2.3. Финальный результат работы алгоритма.*

Синтезированный алгоритм составлен, он выполняет главную цель – решает задачу процедурной генерации помещения, поставленную в самом начале данной работы.

Теперь необходимо вычислить значения критериев реалистичности и разнообразия для синтезированного алгоритма, чтобы сравнить его с результатами алгоритмов из [1] и [9]. Для данного алгоритма критерии считались полностью автоматически, и учитывались все выходы конечной версии программы, а это значит, что для вычисления критериев для данного алгоритма использовалась значительно большая выборка, чем в случае с другими работами, в общей сложности для вычисления было задействовано более 100 сгенерированных помещений, что точно также могло сказаться на значении критериев. Некоторые другие примеры результатов работы синтезированного алгоритма приведены в Приложении 2.

Тем не менее, для синтезированного метода мы имеем: D = 28,013, R = 0,131. Значение критерия разнообразие алгоритма примерно равно результатам работы [8]. При этом, как и ожидалось значение критерия реалистичности немного снизилось, по причинам, описанным в конце 1.6.

Однако, важно отметить, что хоть синтезированный метод и удовлетворяет каждому отдельному критерию хуже, но зато лучше удовлетворяет им в совокупности.

Кроме того, описанный подход все равно обладает одним неоспоримым преимуществом – гибкостью, в основном благодаря методу ограниченного роста и введенной структуре данных. Первый можно настроить для работы практически с любыми видами помещений, от квартир и частных домов до огромных небоскребов и супермаркетов, он также позволяет использовать шаблоны помещений с заранее размещенными определёнными объектами, например лестницами, что является крайне важным при разработке видеоигры, где правила могут диктоваться не только критериями реалистичности, но и геймдизайном. Новая структура данных же освобождает описанный в [8] метод от неповоротливости используемого там подхода.

В итоге, мы получили метод процедурной генерации помещений, обладающий неизмеримым потенциалом для применения и расширения, и при этом не уступающий другим существующим методам.

2.3 Основные сущности программы

При проектировании программы был применен объектно-ориентированный подход.

Class Point – объект точки, обозначающей угол комнаты. В данной реализации использовался только для хранения координат, но можно расширить функционал класса, добавив поле для хранения угла поворота.

Class Space –объект помещения, владеет списком Point – набором своих углов и методом Sort(), в котором реализован алгоритм из параграфа 2.1.

Class Room – объект комнаты, наиболее важная сущность, наследуется от класса Space, также имеет набором методов для реализации процедурной генерации: внутренних, таких как CreateNewWall(), CheckSpace(),MoveWall() и глобальных, таких как Grow(), GrowLShape(), FillSpace(), ReactOnHall(),CreateDoorOnWall().

Class PGManager – объект управляющий процессом процедурной генерации и получающий входные данные, задает порядок роста комнат, собирает информацию о процессе их генерации и передает комнатам необходимые входные данные.

При написании программы использовался язык программирования C# для логики и игровой движок Unity3D для визуализации результата работы алгоритма.

Выводы

Одним из главных нововведений данной работы является постановка строгих критериев оценки сгенерированных помещений, и их использование для выделения самых эффективных подходов. Главным преимуществом введенных критериев является возможность модифицировать набор характеристик, на которых они построены, тем самым приспосабливая критерии к конкретным ситуациям и целям. Из недостатков же новых критериев можно отметить зависимость критерия реалистичности от собранной по реальным помещениям статистики, на что может быть потрачено значительное время.

Кроме того, были выделены самые эффективные подходы к решению задачи процедурной генерации помещения, и на их основе составлен новый синтезированный метод, обладающий повышенной гибкостью применения, и расширяемостью. При этом алгоритм демонстрирует прекрасные значения критериев, хотя и незначительно уступает лучшим показателям.

Заключение

В данной дипломной работе были разобраны различные подходы к методам процедурной генерации планов помещений для виртуальных миров. А затем выделены несколько основных подходов: основанные на графах, на древовидных картах и на методе роста. Было отмечено, что среди рассмотренных работ не предложено строгих критериев оценки результата алгоритма процедурной генерации. Такие критерии, а конкретнее критерий разнообразия и критерий реалистичности, были введены и применены при оценке различных существующих решений в данной бакалаврской работе. Они позволили выделить наиболее эффективные подходы и использовать для реализации нового синтезированного метода, сочетающего в себе самый лучший метод по критерию разнообразия и самый лучший по критерию реалистичности, и получить результаты сравнимые с результатами коллег.

В результате выполнения бакалаврской работы разработана программа для процедурной генерации помещений, в основе которой лежит новая гибкая структура данных, обеспечивающая возможность ее дальнейшего расширения и применения для реализации компьютерной игры.

В дальнейшем планируется улучшить визуальное представление помещения, а также расширить функционал программы, добавив, например, генерацию мебели в помещениях, создание других типов помещений и построение трехмерного помещения, вместо двухмерного.

Список используемой литературы:

[1] M. Smelik, Klaas Jan de Kraker and Saskia A. Groenewegen TNO Defence, Security and Safety The Hague, The Netherlands; Tim Tutenel and Rafael Bidarra Delft University of Technology Delft, The Netherlands, “A Survey of Procedural Methods for Terrain Modelling.”

[2] Pascal Muller, Peter Wonka, Simon Haegler, Andreas Ulmer, Luc Van Gool, K.U. Leuven , “Procedural Modeling of Buildings”, ETH Zurich

[3] Jess Martin, “Procedural house generation: A method for dynamically generating floor plans”, 2006

[4] Xiao-Yu Wanga,Yin Yangb , Kang Zhanga, “Customization and generation of floor plans based on graph transformations”, University of New Mexico, Albuquerque, NM, USA, 14 July 2018

[5] F. Marson and S.R. Musse. “Automatic Generation of Floor Plans Based on Squarified Treemaps Algorithm” IJCGT International Journal on Computers GamesTechnology, 2010.

[6] Maysam Mirahmadi, Abdallah Shami “A Novel algorithm for Real-Time procedural generation of building floor plans”, 26 Nov 2012

[7] Katja Knecht, “Generating Floor Plan Layouts with K-d Trees and Evolutionary Algorithms”, Politecnico di Milano University, Italy

[8] Ricardo Lopes, Tim Tutenel, Ruben M. Smelik, Klaas Jan de Kraker and Rafael Bidarra “A constrained growth method for procedural floor plan generation”, 2010

[9] Daniel Camozzato “A method for growth-based procedural floor plan generation”, Porto Alegre, 2015

[10] M. Bruls, K. Huizing and J. van Wijk, “Squarified treemaps”

Интернет ресурсы:

(1) Сайт spb.fedned.ru - был использован для поиска планов квартир жилых домов.

Приложение 1. Реальные планы помещений

|  |  |
| --- | --- |
| Характеристика | Значение |
| Отношение жилой площади помещения (жилые комнаты, гостиная) к нежилой (коридор, кухня, ванная). | 1,357 |
| Среднее значение соотношения длины к ширине прямоугольников, описывающих каждую комнату. | 1,557 |
| Соотношение площади коридора к площади всего помещения. | 0,156 |
| Среднее количество углов в комнатах | 4,13 |
| Среднее количество углов во внешних стенах помещения | 4,6 |
| Среднее количество дверей на пути в комнаты от входа в помещение | 1,2 |

Приложение 2. Примеры результатов синтезированного метода процедурной генерации помещений

Кухня

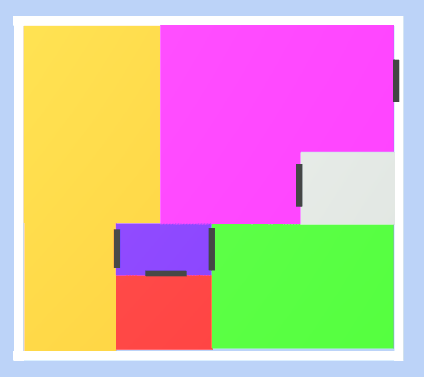
Ванная

Коридор

Кладовка

Гостинная

Спальная комната



Кладовка

Кухня

Коридор

Ванная

Спальная комната

Гостинная

Кладовка

Коридор

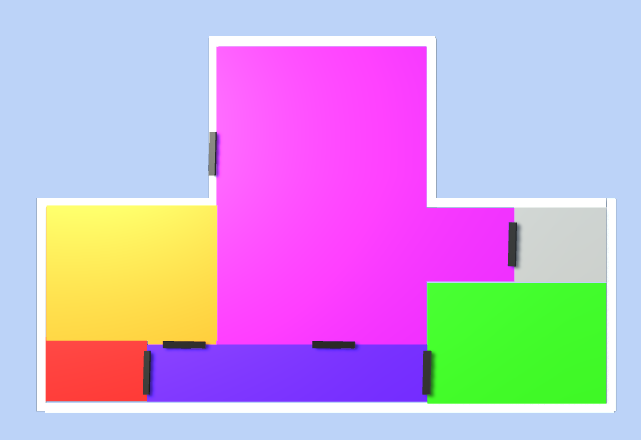
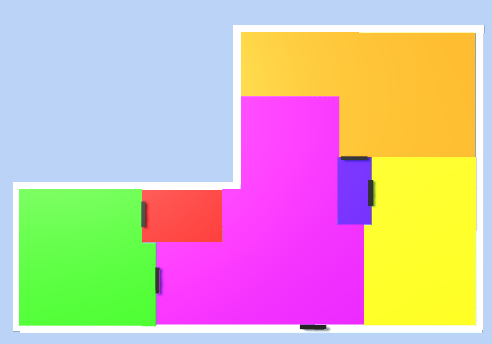
Ванная

Кухня

Гостинная

Спальная комната

Спальная комната 2



Кухня

Коридор

Спальная комната 1

Ванная

Гостинная

Кухня

Ванная

Коридор

Спальная комната 2

Спальная комната 1

Гостинная

