Проблема:

Необходимо разработать универсальную систему генерации уровней, основанную на предыдущих успехах игрока.

Анализ существующих решений:

Аналоги если и есть, то существуют с закрытым кодом, это не российские разработки, и заточены под конкретные проекты. Были найдены только методы генерации уровней(статьи 3 и 4).

Под сложностью игры мы будем подразумевать классическое ее значение, взятое из настольной игры Dungeons & Dragons. Суть в том что вызов игроку должен соответствовать уровню его силы(это уже проверено годами). Т.е. если численно представленная сила игрока равна 1000, то и суммарная сила или сложность препятствий на его пути должна быть связана с этим значением. Она может быть немного выше(если игрок предпочитает игру посложнее), или немного ниже, если игрок не уверен в своих силах. Важно одно, нам необходимо ввести численные значения препятствий, как в указанной выше игре, отображающие их сложность, и таковые же для игрока(или их аналоги).

Так как средство должно быть универсальным, все задачи, ловушки и противники с которыми будет сталкиваться игрок будут максимально упрощены, чтобы наглядно продемонстрировать эффективность метода и не заострять внимание на геймплее. Т.е сложность препятствий будет обозначаться их кол-вом, а сила игрока будет обуславливаться победой или поражением на предыдущем уровне.

Геймплей:  
Игрок оказывается в лабиринте, цель собрать все монетки. Для простоты, все что может делать игрок это перемещаться.

Уровень состоит из множества комнат, в части или во всех комнатах могут быть препятствия.

Виды препятствий:

1. Ловушка(статична)
2. Монстр(движется по одному маршруту)

Уровень меняется в одном из двух случаев:

1. Игрок собрал все монетки(игра усложняется)
2. Игрок коснулся препятствия, т.е. умер(игра упрощается).

Алгоритм:

Возьмем стартовую величину сложности 5.

1. Генерируем кол-во ловушек от сложности
2. Генерируем кол-во монстров от сложности
3. Генерируем кол-во ключевых вершин лабиринта от сложности
4. Объединяем ключевые вершины попарно, при этом удаляя одну из них из выборки. Это нужно чтобы гарантировать связность нашего графа вершин.
5. Нормальным распределением располагаем ловушки и монстров в комнатах
6. Старт игры.
7. Победа увеличивает сложность игры на 1, поражение уменьшает.

Использованные источники:  
1. Эффективный метод генерации случайных геометрических графов для моделирования беспроводных сетей URL: <http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=pdm&paperid=566&option_lang=rus>

2. Создание миникарты  
<https://habr.com/ru/post/304632/>

3. Статья про генерацию уровней без учета сложности:  
<https://cyberleninka.ru/article/n/generator-urovney-dlya-igrovyh-prilozheniy-1>

4. Статья на хабре про генерацию уровней:

<https://habr.com/ru/post/321602/>

Введены арифметические зависимости от сложности, дабы она не росла экспоненциально.

Скрины сгенерированных уровней по сложности от 5 до 10:

Далее описаны результаты работы программы:

New level generating. Difficult: 5

Count of traps: 1

Count of Enemies: 5

Map size: 10 X 10

Count of vertices and collectable objects: 4



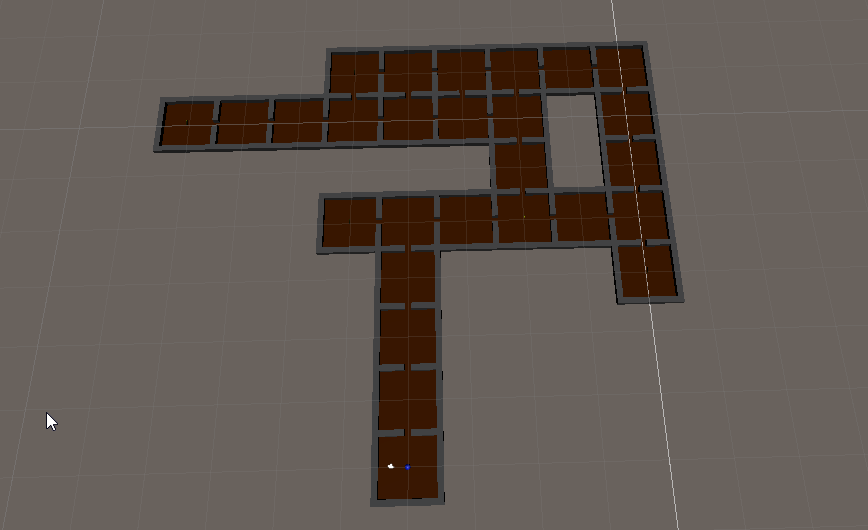
New level generating. Difficult: 6

Count of traps: 3

Count of Enemies: 6

Map size: 12 X 12

Count of vertices and collectable objects: 6



New level generating. Difficult: 7

Count of traps: 2

Count of Enemies: 7

Map size: 14 X 14

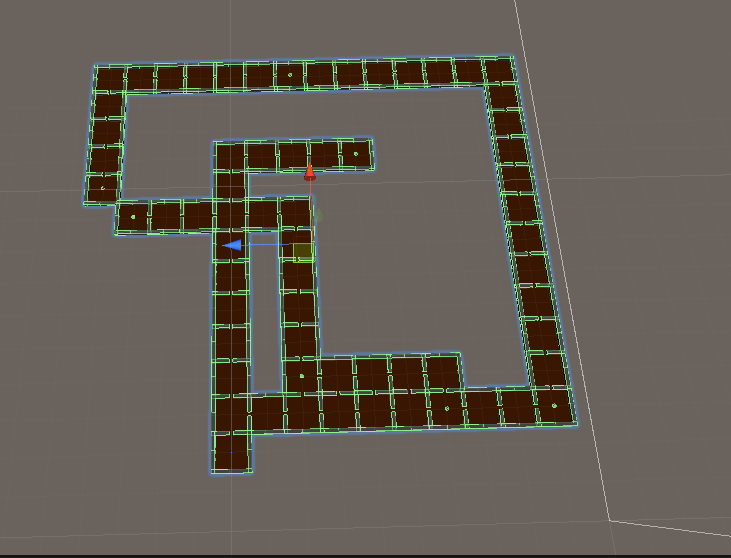
Count of vertices and collectable objects: 6

New level generating. Difficult: 8

Count of traps: 6

Count of Enemies: 8

Map size: 16 X 16

Count of vertices and collectable objects: 8

New level generating. Difficult: 9

Count of traps: 5

Count of Enemies: 9

Map size: 18 X 18

Count of vertices and collectable objects: 8



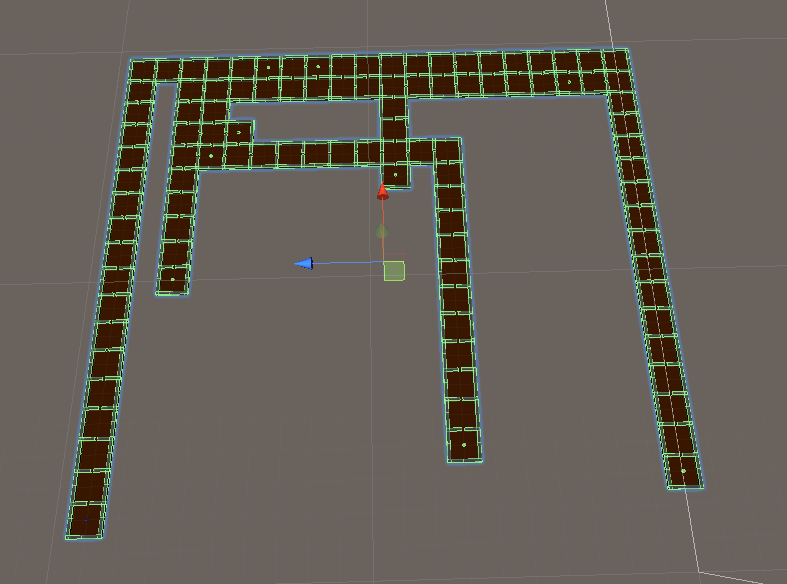
New level generating. Difficult: 10

Count of traps: 4

Count of Enemies: 10

Map size: 20 X 20

Count of vertices and collectable objects: 10



Данные уровни сгенерированы со следующими модификаторами сложности  


Процесс генерации препятствий и лабиринта содержит в себе случайные числа:  
количество ловушек = от 0 до currentDifficult \* countOfTrapsModificator

количетсво монстров = currentDifficult \* countOfTrapsModificator

размер карты = currentDifficult \* mapSizeModificator на currentDifficult \* mapSizeModificator

количество важных вершин(по ним строится лабиринт и каждая из них содержит собираемый объект) = currentDifficult \* maxCountOfVerticesModificator. Это значение обязательно четное