

1. Цели и задачи

1.1. Глобальные цели и задачи

Результатом работы должно быть решение, позволяющее генерировать игровой уровень адекватный с точки зрения левелдизайна и логичный с точки зрения представляемой локации. Основным интерес в рамках работы представляют здания и городской ландшафт.

Аспект	Варианты	Мысли
Возможные результаты работы	<ul style="list-style-type: none">• stand along решение;• решение для UE4;• Houdini.	В качестве отправной точки было принято решение взять UE, т.к. достаточно знакомы с ним.
Игровое пространство	<ul style="list-style-type: none">• коридор• открытый мир.	Предварительно, вариант с коридором кажется более перспективным.
Как реализовать	<ul style="list-style-type: none">• сгенерировать город, после чего выделить в нем уровень;• пустить на уровень агентов, чтобы оценить «играбельность».	-

1.1. Ближайшие цели и задачи

К концу декабря сделать демо с генерацией района из 9-этажек на идеальной плоскости хотя бы самым простым способом. Хочется красивого декора (кондиционеры, трещины, подтеки, паралакс маппинг и т.д.).

1.2. Текущие цели и задачи

Посмотреть:

- генеральные планы городов;
- GDC и процедурная генерация;
- статьи про генерацию (прочитать и осознать).

Расставить приоритеты для рассмотренных методов.

1.3. Основные направления исследования:

- методы генерации городского ландшафта;
- левелдизайн;
- урбанистические принципы.

1.4. Источники, на которые следует обратить внимание:

- Статья Ubisoft про процедурную генерацию (где-то такая была);
- Интересная архитектура:
 - Морской вокзал
 - Архив ВМФ
 - Дома на ножках
 - Самый тонкий дом
 - Дом атомщиков
 - Дома авиаторов и т.д.
- Город из “Bladerunner”
- Генерация чанками
- Соляризация квартир и расстояния между зданиями
- Канал «ПроВокруг»

2. Научные статьи

Название: **General Methods for the Generation of Seamless Procedural Cities**

Ссылка:

<https://lup.lub.lu.se/luur/download?func=downloadFile&recordId=8929185&fileId=8929189>

Авторы: Tobias Elinder

Год: 2017

Описание:

Статья посвящена генерации города начиная с дорог и заканчивая фурнитурой. Автору хотелось модульности => генерация поэтапная и начинается с дорог, просто потому, что так проще.

Исходный полигон «разрезается» дорогами на множество полигонов меньшего размера. Далее эта информация используется для генерации зданий и тротуаров.

Полигоны между дорогами разделяются на полигоны под здания средствами Plot Generator'a (модуля, ответственного за разделение полигонов на контуры зданий и на остальную территорию).

Для генерации дорог используется алгоритм, выросший из L-систем. Идея заключается в использовании для генерации дорог приоритетной очереди. Алгоритм добавляет небольшие кусочки дорог на каждом шаге. Описание алгоритма: http://nothings.org/gamedev/l_systems.html

Для того, чтобы управлять процессом генерации используется тепловая карта.



(a) City 1, type = Normal



(b) City 2, type = Normal



(c) City 3, type = Normal

Figure 5.4: Resulting cities

Название: **A Survey of Procedural Techniques for City Generation**

Ссылка: <https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1097&context=itbj>

Авторы: George Kelly, Hugh McCabe.

Год: 2006

Описание:

В статье в принципе рассказывается о множестве разных подходов к генерации абсолютно разного контента. Относительно городского ландшафта рассмотрено несколько подходов.

Генерация на решетке

Плюсы:

- +быстро;
- +просто;
- +масштабируемо;
- +не нужны сложные входные данные;
- +результат просто редактировать вручную.

Минусы:

- не реалистично;
- однообразно.



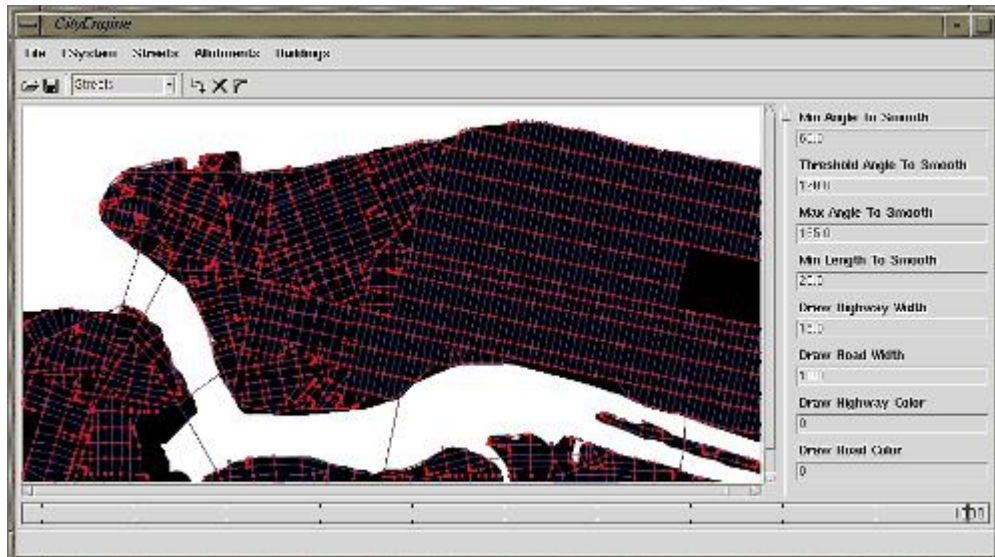
L-системы

Плюсы:

- +достаточно реалистично;
- +учитывает паттерны;
- +нет серьезных проблем с масштабируемостью, главное, чтобы входная карта была нужного размера.

Минусы:

- нужны реальные входные данные;
- не подходит для генерации в реальном времени;
- нужно эмпирически подбирать множество параметров, чтобы получилось хорошо.



Агентные системы

предполагают производить симуляцию взаимодействия агентов для генерации города.

Генерация дорог происходит за счет экстендеров и коннекторов. Первые исследуют новые территории и пытаются проложить в них дороги. Вторые исследуют существующие дороги и при необходимости соединяют их.

Далее город разбивается на районы за счет симуляции взаимодействия агентов с разными интересами.

Плюсы:

- +реалистичный результат при относительно скромных входных данных;
- +естественно учитывает различия различных районов (коммерческих, жилых и т.д.);
- +достаточно управляемый процесс генерации.

Минусы:

- сомнительная масштабируемость;
- алгоритм требует много времени на работу.

Шаблонная генерация

На карту «накладывается» шаблон, которые алгоритм пытается «разрастить».

Плюсы:

+???

Минусы:

- паттерны хоть и взяты «из жизни», все же результат выходит даже близко не реалистичным;
- сомнительная масштабируемость;
- достаточно много входных данных (геоданные, карта высот, в определенных случаях плотность населения и т.д.);
- перформанс (???);
- практически никакого контроля над процессом генерации.

Название: **Citygen: An Interactive System for Procedural City Generation**

Ссылка: <http://www.citygen.net/>

Авторы: George Kelly, Hugh McCabe.

Год: ???

Описание:

Статья описывает механизмы, используемые внутри инструмента Citygen. Сначала генерируются главные дороги, потом генерируются второстепенные с использованием L-систем с разными заготовленными паттернами. Инструмент во многом ориентируется на возможность вручную править результаты генерации.

Главные дороги:

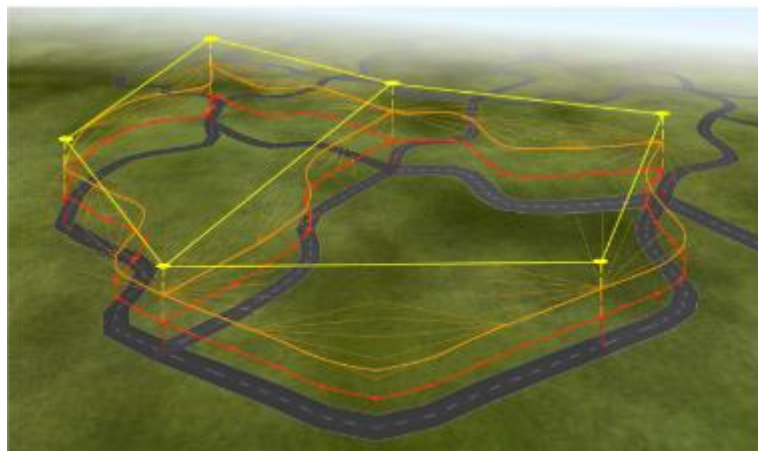


Figure 2 Primary road network graphs, Yellow: High level graph, Red: Low-level graph, Orange: Plot samples and interpolation spline.

Паттерны:



Figure 7: Road network pattern: Raster, Industrial and Organic

Заполнение ячеек в соответствии с определенным паттерном:

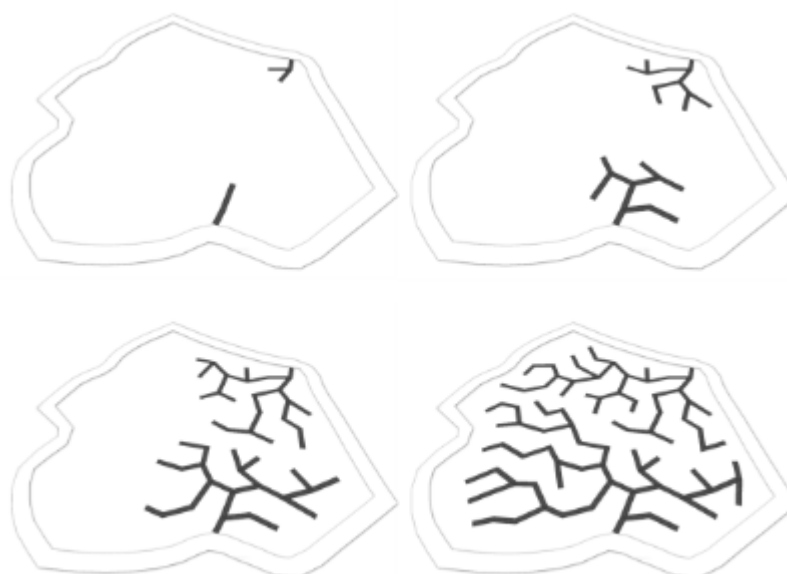


Figure 6: Road Growth 10, 100, 300 & 1000 steps.

Название: **Procedural generation of road networks for large virtual environments**

Ссылка:

<https://scholarworks.rit.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=6536&context=theses>

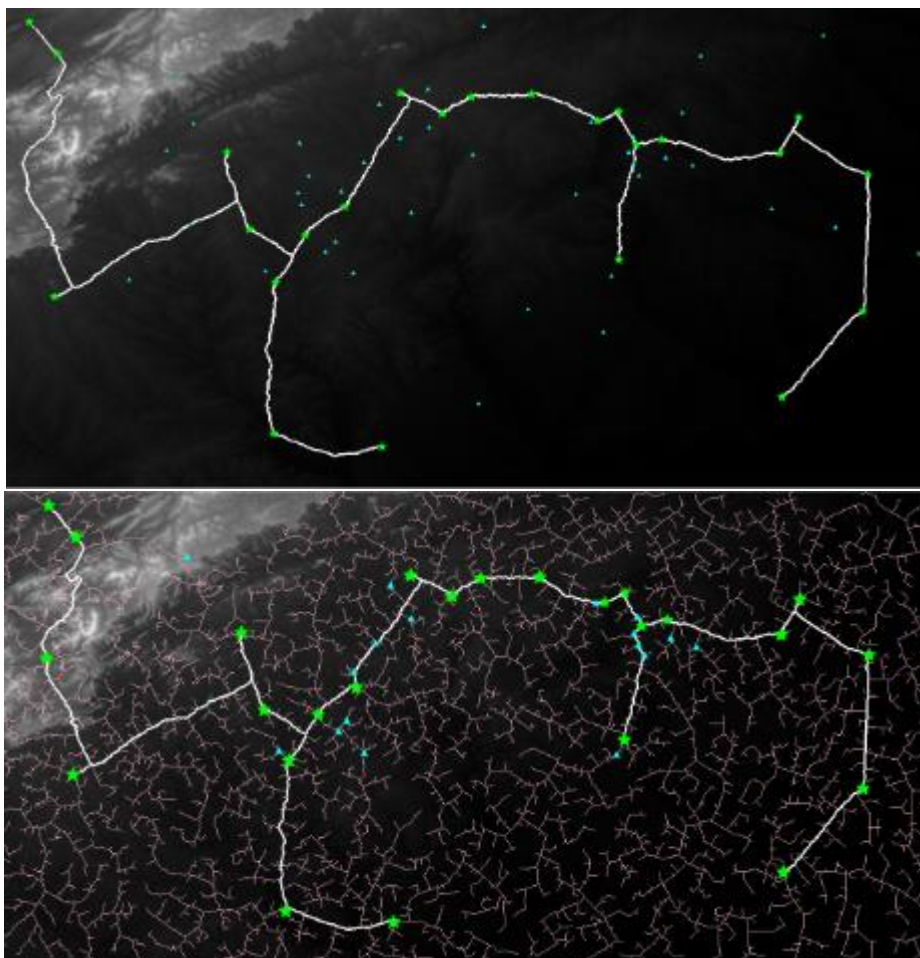
Авторы: Craig Martek

Год: 2012

Описание:

Подход использует карту и статистическую информацию о популяции. Пытается строить наиболее оптимальные маршруты между точками концентрации популяции, строя тем самым главные дороги.

После строятся второстепенные дороги, которые связывают с основными деревушки и города.



Итог: вряд ли эта статья на текущий момент нам очень интересна, т.к. это про генерацию дорог на более высоком уровне чем требуется.

Название: **Procedural Modelling of Urban Road Networks**

Ссылка:

https://www.researchgate.net/publication/260335265_Procedural_Modelling_of_Urban_Road_Networks

Авторы: Jan Beneš, Jaroslav Krivanek, Alexander Wilkie

Год: 2014

Описание:

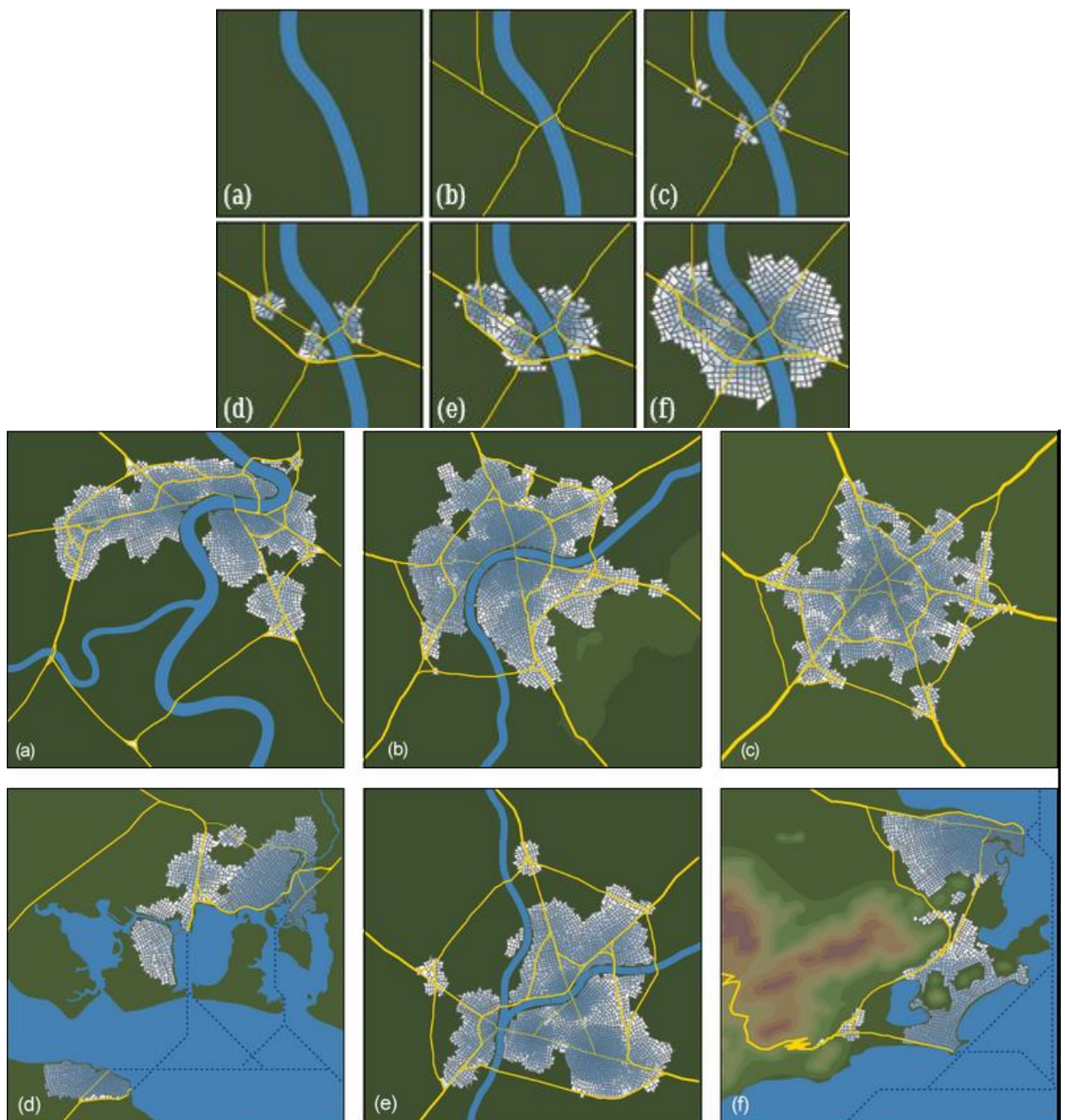
С лету можно сказать, что данная статья – хорошее основание для поисков фундаментальных работ в области (много полезных ссылок).

Очень любопытная работа, использующая симуляцию трафика и предположения о том, что город есть результат роста нескольких маленьких городков рядом, которые в итоге соединились.

Алгоритм:

- инициализирующая генерация основных дорог и поиск очагов роста городов;
- города обрастают второстепенными дорогами;
- симулируется трафик и определяется, хватает ли нам второстепенных и основных дорог, чтобы обеспечить необходимый трафик;
- на основании результатов предыдущего шага обрастаем (или нет) новыми дорогами

Чтобы обеспечить отличия, которые проявляются во времени, на разных итерациях цикла, описывающих разные временные периоды, используются разные паттерны для генерации второстепенных дорог.



Название: **ФОРМИРОВАНИЕ МИКРОРАЙОННЫХ ОБРАЗОВАНИЙ
НОВОСИБИРСКА В 1950–2010-е ГОДЫ**

Ссылка: <http://old.archvuz.ru/PDF/%23%2050%20PDF/ArchPHE%2350pp171-192Gashenko.pdf>

Авторы: Гашенко Антон Евгеньевич

Год: 2015

Описание:

Несмотря на то, что в статье идет речь о микрорайонах Новосибирска, все же из нее можно извлечь полезную информацию о микрорайонном строительстве в принципе.



Название: **Procedural generation of road networks using L-systems**

Ссылка: <https://liu.diva-portal.org/smash/get/diva2:1467574/FULLTEXT01.pdf>

Авторы: Martin Jormedal

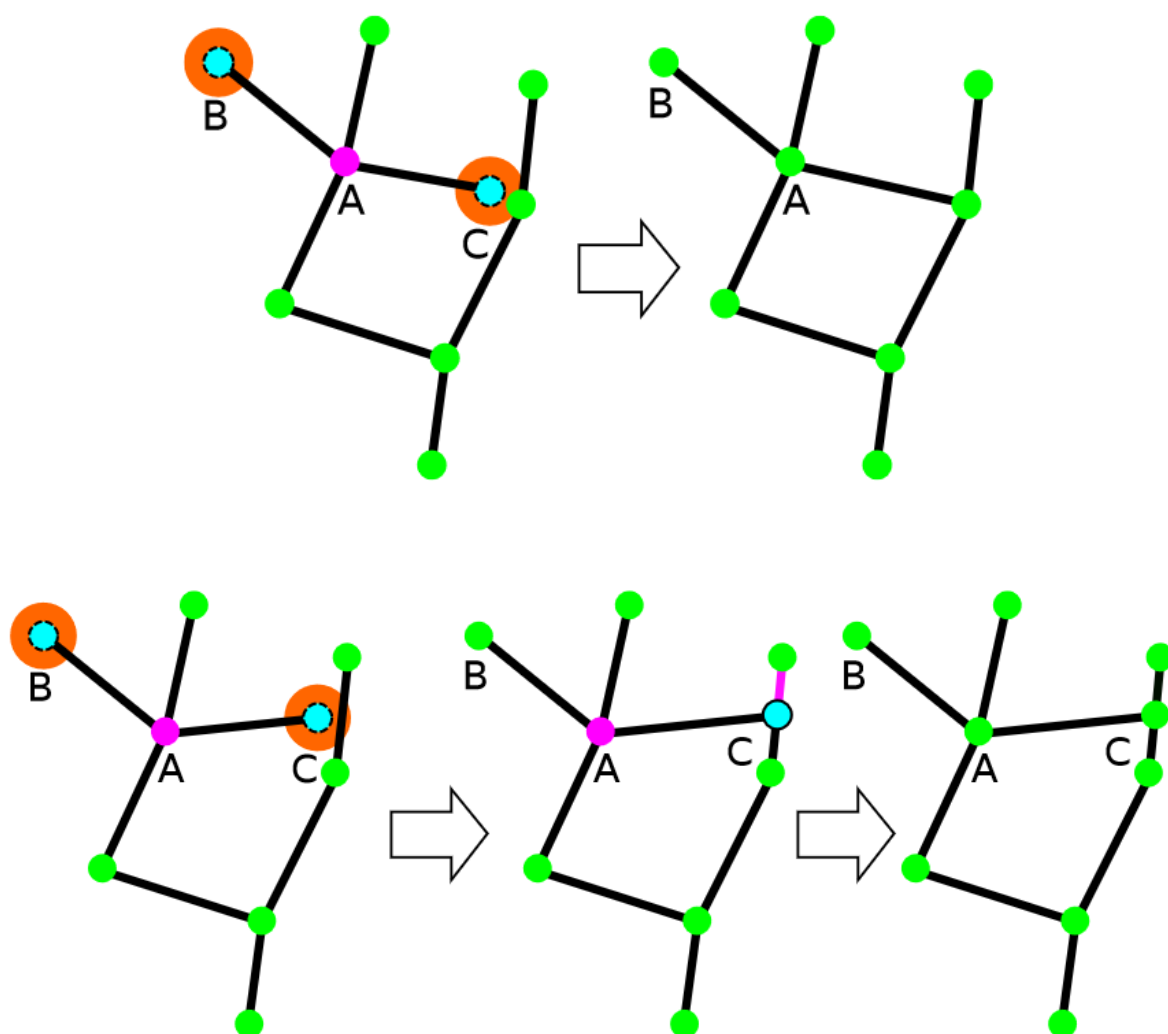
Год: 2013

Описание:

Статья посвящена имплементации дорожного генератора на L-системах. В ней достаточно подробно описывается механизм L-систем, начиная с контекстно свободных L-систем и заканчивая параметризованными L-системами и представлением L-систем в виде графов для решения проблемы ацикличности.

Описана реализация генератора дорог на L-системах, использующего высотную карту, тепловую карту и правила, заданные в XML файле для генерации.

Описан алгоритм разрешения пересечений:



Результаты:



2.1. Конференции

Про генерацию города в Spider-Man

<https://www.youtube.com/watch?v=4aw9uyj9MAE&list=WL&index=89>

Пайплайн:

- Ground (Сначала генерация земли с дорогами, тротуарами, некоторым декором);
- Buildings (Здания с разметкой);
- Traffic (Система трафика для транспорта и пешеходов);
- Props (Всякие пропсы).

Ground этап определяет структуру города. Все зависит от того, как будут расположены дороги и аллеи.

Далее применяется Ground Modifiers, которые, собственно, модифицируют полученную землю. Изменяют, деформируют, накладывают материалы, режут и т.д. Применяются к определенным территориям в игре.

Режем землю на квадраты (тайлы) (видимо, это важно для открытого мира). Тайлы имеют ID и содержат информацию о содержащихся в них территориях

Далее следует этап Ground Details: перекрестки, разметка, немного окружения.

Изначально город генерируется в виде «серых коробок» и дизайнеры могут руками поменять свойства домов, подвигать их, деформировать дороги и т.д.

Итоги:

Нужно стремиться уменьшать зависимости, чтобы ручные изменения не приводили к эффекту домино, когда изменения в одном месте приводят к каскаду изменений во множестве других мест.

Не стоит стремиться к невероятной точности с огромным числом ограничений. Рука дизайнера важнее.

CppCon 2017: Alan Bucior “Building Better Worlds: Developing a Procedural City Tool for Alien...”

<https://www.youtube.com/watch?v=F9tGa-hbmTU>

Занятная штука. Заточено в первую очередь на то, чтобы дизайнерам было удобно манипулировать с результатами генерации. И тоже использованы L-системы.

2.2. Популярныe источники и интересные ссылки

The Sinking City

https://www.youtube.com/watch?v=b5AIpncO_9I

Краткий обзор инструмента, использованного в игре Singing City. Представляет собой инструмент для UE4.

Предполагает создание множества пресетов зданий. После задается «решетка» города в виде графа (вручную). В результате получаются районы, которым можно задать их предназначение (коммерческий район, жилой район и т.д.). После генерируются дороги и расставляются дома. Все это легко поддается кастомизации после того как процесс генерации был завершен.

Итог: результат простой, но в рамках данной игры этого, видимо, достаточно.

City procedural generation: Voronoï approach

<http://gamedevindie.com/city-procedural-generation-voronoi-approach/>

Простое решение, которое практически полностью построено на диаграммах Вороного. Результат, закономернo, простой, но выглядит аккуратно и красиво.

Быстро про основы советской микрорайонной застройки

<https://www.youtube.com/watch?v=JGVBv7svKLo>

В видео рассказывается про базовые принципы и логику микрорайонной советской застройки.

Обзорно об L-системах с полезными ссылками

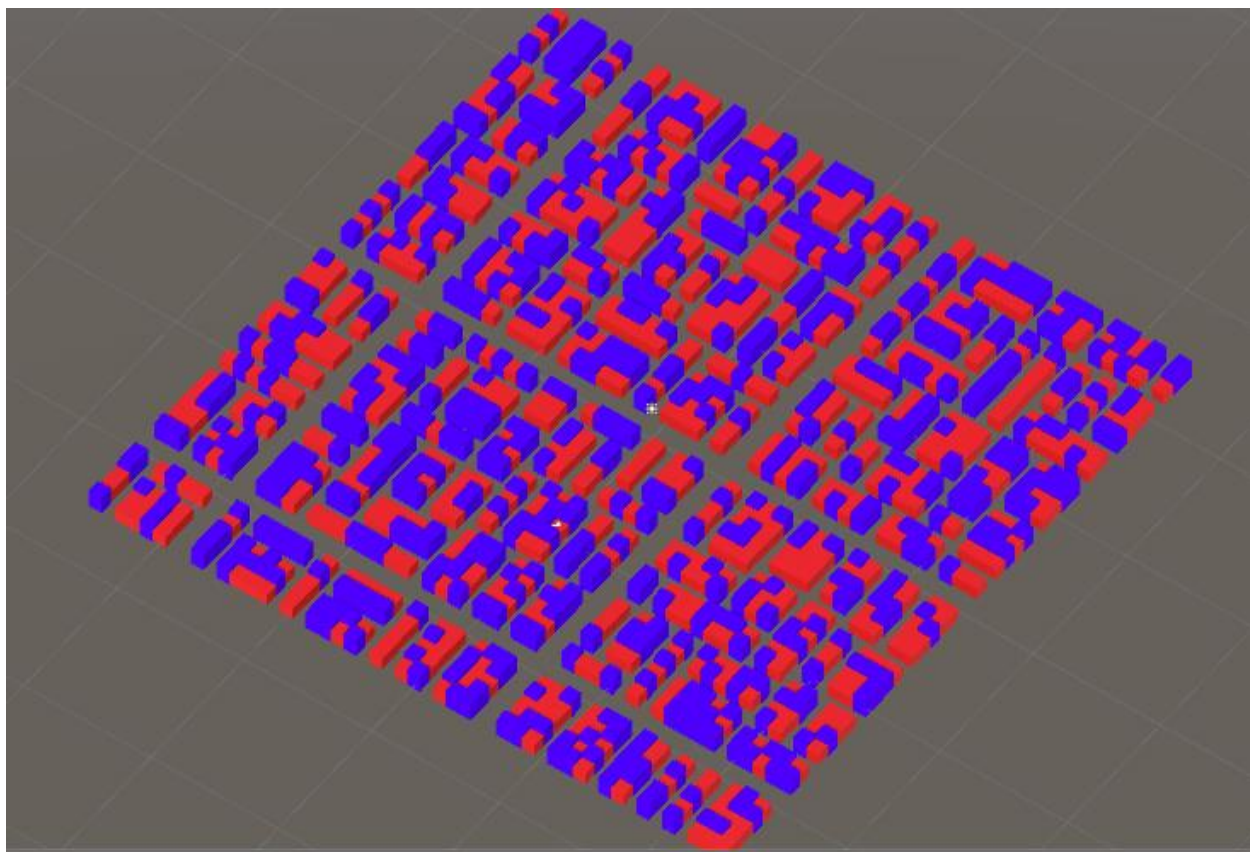
<https://medium.com/nerd-for-tech/a-peek-at-l-systems-29b847b23365>

Простенький пример с контекстно свободной грамматикой. Обзор.

Предельно простой пример использования L-систем

<https://projectlucky Luciano.wordpress.com/2016/04/18/l-systems-map-generation-algorithm/>

По сути описан алгоритм, который позволяет получить результаты чуть более интересные, чем решетка.



А здесь наглядная имплементация такого подхода:

<https://gamedev.stackexchange.com/questions/86234/using-l-systems-to-procedurally-generate-cities>

Procedural Generation For Dummies: Road Generation

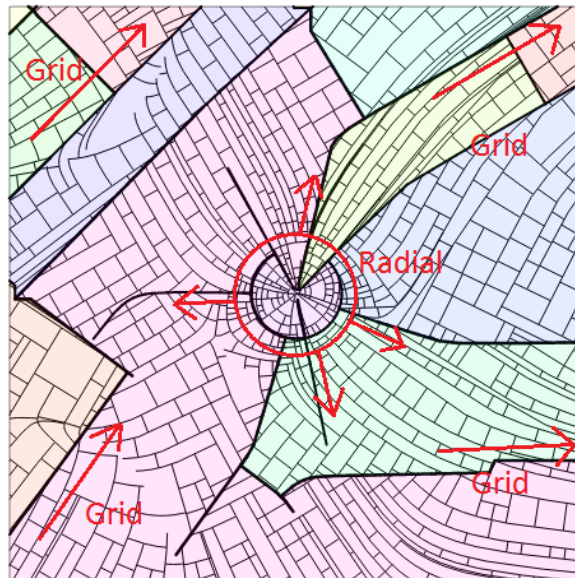
<https://martindevans.me/game-development/2015/12/11/Procedural-Generation-For-Dummies-Roads/>

Не смотря на название, статья ссылается на ряд научных статей и довольно доходчиво объясняет суть алгоритма генерации дорог на L-системах. Что еще более приятно, статья является пояснением для кода:

<https://github.com/martindevans/Base-CityGeneration/tree/master/Base-CityGeneration/Elements/Roads/Hyperstreamline>

Поэтому данная статья может стать неплохой отправной точкой.

Для управления генерацией используются тензорные поля. Они определяют паттерн, по которому генерируются дороги.



Данная статья берет идеи из этой статьи:

Interactive Procedural Street Modeling

http://www.sci.utah.edu/~chengu/street_sig08/street_project.htm

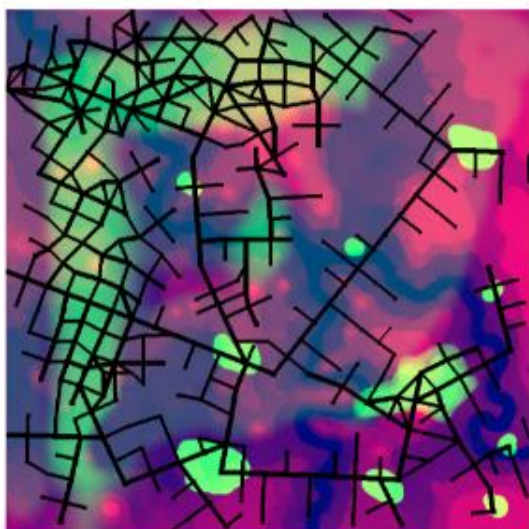
Здесь как раз и описан подход с тензорными полями. Эта идея направлена в первую очередь на повышение контроля над процессом генерации.

2.3. Код

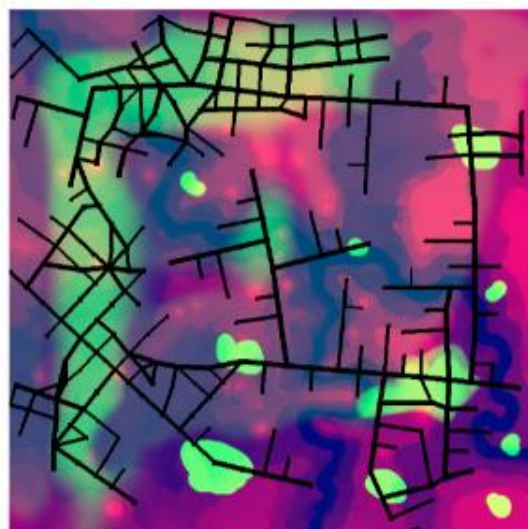
pokitoz/Procedural-road-network-generation

<https://github.com/pokitoz/Procedural-road-network-generation>

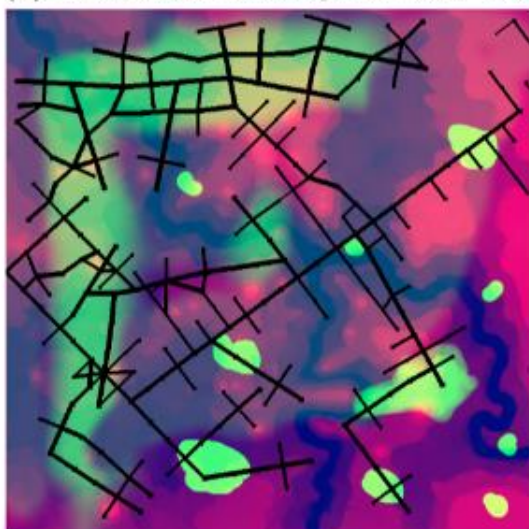
Код генерации дорог на L-системах. Результаты не слишком впечатляющие, но по крайней мере от этого можно будет оттолкнуться.



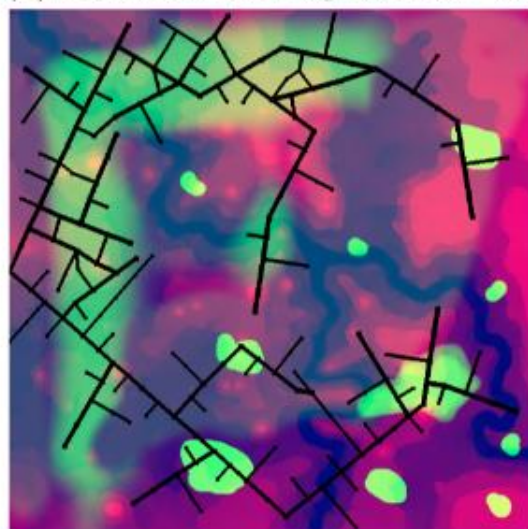
(a) Parameter `sub_square_size=20`.



(b) Parameter `sub_square_size=30`.



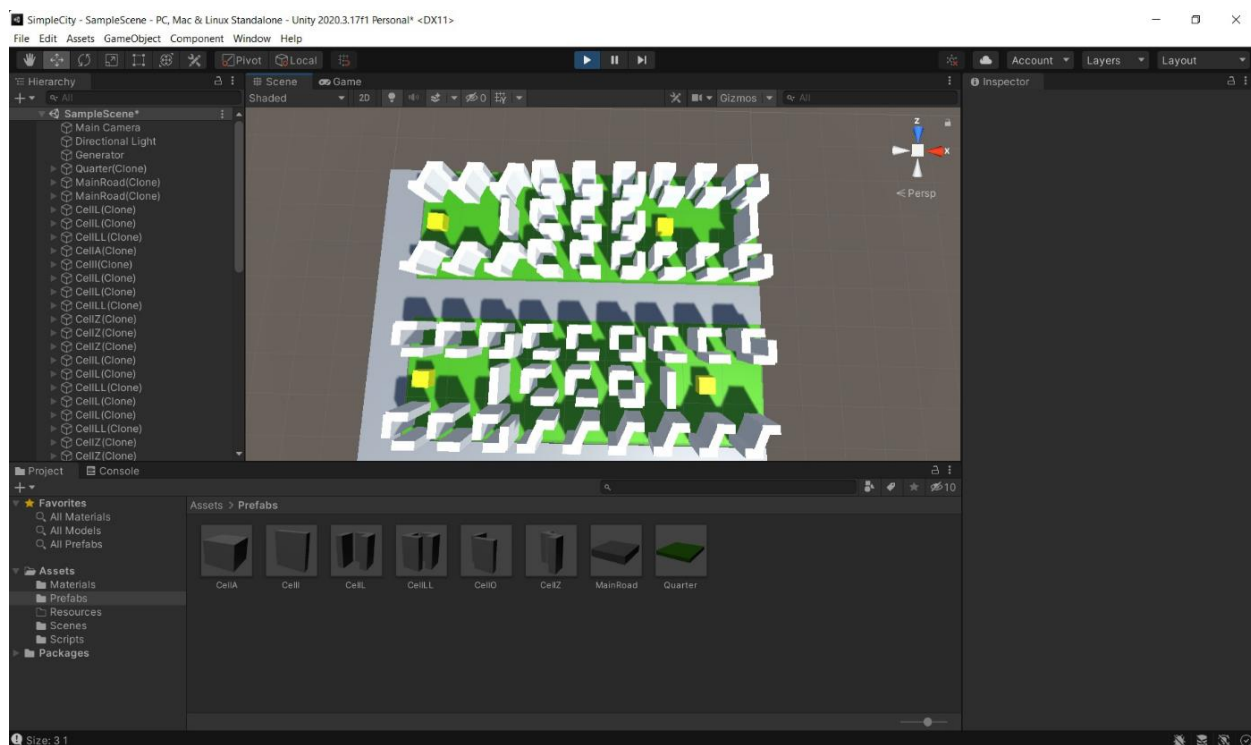
(c) Parameter `sub_square_size=40`.



(d) Parameter `sub_square_size=45`.

3. Собственные наработки

Вдохновившись статьей о простой генерации города с помощью L-систем и желая получше разобраться, как L-системы работают, решил попытаться реализовать собственно простое решение. Получил следующие результаты:



На скриншоте представлена простейшая генерация микрорайонной застройки.

4. TODO:

https://www.researchgate.net/publication/257406695_Procedural_Generation_of_Villages_on_Arbitrary_Terrains

<https://www.sciencedirect.com/science/article/pii/S2352146515002690>

Example driven generation:

<https://www.youtube.com/watch?v=VGjZhFAiPBo>

http://www.ignaciogarciadorado.com/p/2015_CGF/2015_CGF_ExampleRoads.pdf