УДК 004.9

Е.Н. МИХАЙЛУЦА, А.В. ПОЖУЕВ, С.А. ВОЛИК
Запорожский национальный университет

АНАЛИЗ МЕТОДОВ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ ТРЕХМЕРНОГО ИГРОВОГО КОНТЕНТА

Благодаря мощному развитию современной игровой индустрии создание видеоигр все чаще воспринимается авторами и пользователями как отдельный вид художественного творчества. Как и любая другая культурная индустрия, видеоигры имеют огромный рынок, полный издателей и разработчиков, которые создают высокобюджетные продукты. К существующим в игровой индустрии проблемам относят скорость создания игрового контента и качество обработки игровых объектов, то есть отображение игрового объекта в наиболее реалистичном или нужном виде. В свете этого можно сказать, что процедурная генерация — незаменимый инструмент в геймдизайне.

В представленной работе рассматривается одна из проблем процедурной генерации – процедурная генерация помещений. В первом разделе проведен обзор нескольких самых актуальных методов процедурной генерации контента, которые могут быть использованы для генерации комнат и уровней в видеоиграх, кроме того, проанализированы методы создания планов зданий автоматическим способом. В частности, анализируются известные алгоритмы и подходы использования процедурной генерации сооружений. На основе проведенного анализа методов генерирования домов разработан инструмент для процедурного генерирования планов домов с их трехмерным отображением. В реализованной программной системе добавлена возможность корректировать различные настройки создаваемых объектов, а также возможность изменения пользователем моделей дома. Данная особенность системы открывает возможность генерирования здания методом конструктора из заранее разработанных частей. Как следствие, увеличилась вариативность зданий как внешне (формы зданий), так и внутренне (планы помещений), а также значительно уменьшились затраченные человеко-часы на создание более детализированного игрового объекта здания. Тестирование показало, что разработанная программная система оптимально подходит для применения в режиме редактирования игры.

Также в работе исследованы и проанализированы различные методы оптимизации отображения здания в 3D. В разработанной системе использованы такие методы оптимизации, как объединение элементов здания в единую меш-сетку, корректное создание игровых модульных частей здания, а также использование паттерна проектирования Poolobject для многократного повторения в построении домов. На основе результатов работы в будущем возможно ускорение разработки игр, например, для генерирования процедурно созданного города или создание моделей городов.

Ключевые слова: игровая индустрия, процедурная генерация помещений, компьютерная система, меш-сети.

О.М. МІХАЙЛУЦА, А.В. ПОЖУЄВ, С.О. ВОЛІК
Запорізький національний університет

АНАЛІЗ МЕТОДІВ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ТРИВИМІРНОГО ІГРОВОГО КОНТЕНТУ

Завдяки потужному розвитку сучасної ігрової індустрії створення відеоігор все частіше сприймається авторами і користувачами як окремий вид художньої творчості. Як і будь-яка інша культурна індустрія, відеоігри мають величезний ринок, повний видавців і розробників, які створюють високобюджетні продукти. До існуючих в ігровій індустрії проблем відносять швидкість створення ігрового контенту і якість обробки ігрових об'єктів, тобто відображення ігрового об'єкта в найбільш реалістичному або потрібному вигляді. У світлі цього можна сказати, що процедурна генерація - незамінний інструмент в геймдизайні.

У представленій роботі розглядається одна з проблем процедурної генерації - процедурна генерація приміщень. У першому розділі проведено огляд декількох найактуальніших методів процедурної генерації контенту, які можуть бути використані для генерації кімнат і рівнів у відеоіграх, крім того, проаналізовані методи створення планів будівель автоматичним способом. Зокрема, аналізуються відомі алгоритми та підходи використання процедурної генерації споруд. На основі проведеного аналізу методів генерування будинків розроблено інструмент для процедурного генерування планів будинків з їх тривимірним відображенням. В реалізованій програмній системі додана можливість

коригувати різні настройки створюваних об'єктів, а також можливість зміни користувачем моделей будинку. Дана особливість системи відкриває можливість генерування будівлі методом конструктора з заздалегідь розроблених частин. Як наслідок, збільшилася варіативність будівель як зовні (форми будівель), так і внутрішньо (плани приміщень), а також значно зменшилися витрачені людино-години на створення більш детального ігрового об'єкта будівлі. Тестування показало, що розроблена програмна система оптимально підходить для застосування в режимі редагування гри.

Також в роботі досліджені і проаналізовані різні методи оптимізації відображення будівлі в 3D. У розробленій системі використані такі методи оптимізації, як об'єднання елементів будівлі в єдину меш-сітку, коректне створення ігрових модульних частин будівлі, а також використання патерну проектування Poolobject для багаторазового повторення в побудові будинків. На основі результатів роботи в майбутньому можливе прискорення розробки ігор, наприклад, для генерування процедурно створеного міста або створення моделей міст.

Ключові слова: ігрова індустрія, процедурна генерація приміщень, комп'ютерна система, меш-мережі.

O.M. MIKHAILUTSA, A.V. POZHUYEV, S.A. WOLIK
Zaporizhzhya National University

ANALYSIS OF PROCEDURAL GENERATION METHODS FOR 3D GAME CONTENT

Thanks to the powerful development of the modern gaming industry, the creation of video games is increasingly perceived by authors and users as a separate form of artistic creation. Like any other cultural industry, video games have a huge market full of publishers and developers who create high-budget products. The problems existing in the gaming industry include the speed of creating game content and the quality of processing game objects, that is, displaying the game object in the most realistic or desired form. In light of this, we can say that procedural generation is an indispensable tool in game design.

In the presented work, one of the problems of procedural generation is considered - procedural generation of premises. The first section provides an overview of some of the most current procedural content generation techniques that can be used to generate rooms and levels in video games, and analyzes methods for generating building plans automatically. In particular, well-known algorithms and approaches to using procedural generation of structures are analyzed. Based on the analysis of methods for generating houses, a tool for procedural generation of house plans with their three-dimensional display has been developed. In the implemented software system, the ability to adjust various settings of the created objects, as well as the ability to change the house models by the user, has been added. This feature of the system opens up the possibility of generating a building using the constructor method from pre-designed parts. As a result, the variability of buildings increased both externally (building shapes) and internally (floor plans), as well as significantly reduced person-hours spent on creating a more detailed game object of the building. Testing has shown that the developed software system is optimal for use in the game editing mode.

Also in the work, various methods of optimizing the display of a building in 3D are investigated and analyzed. The developed system uses such optimization methods as combining building elements into a single mesh networks, correctly creating playable modular parts of the building, as well as using the Poolobject design pattern for multiple repetitions in building houses. Based on the results of the work in the future, it is possible to accelerate the development of games, for example, to generate a procedurally created city or create city models.

Keywords: game industry, procedural room generation, computer system, mesh networks.

Постановка проблемы

Процедурная генерация контента – одно из актуальных и быстроразвивающихся направлений в сфере разработки компьютерных игр. Процедурная генерация подразумевает автоматическое создание наполнения игрового мира, к которому могут относиться всевозможные игровые объекты, такие как уровни, комнаты, предметы, противники, инструменты игрока.

Другими словами, процедурное генерирования – это инструмент, который может ускорить разработку большого количества игровых проектов и уменьшить количество затраченных человеко-часов на создание цифрового контента.

Анализ последних исследований и публикаций

В международном сообществе тема процедурной генерации весьма популярна. Однако, далеко не все полезные материалы являются научными работами – огромное количество ресурсов и статей, посвященных процедурной генерации, являются авторскими блогами, при этом они несут значимую ценность своей практичностью. Большая часть материалов является англоязычными, хотя в последнее время многие статьи активно переводятся на русский язык.

В книге [1] представлены самые современные сведения о процедурной генерации контента (PCG) для игр, в частности, процедурной генерации уровней, ландшафтов, предметов, правил, квестов или других типов контента. Активные академические исследователи и разработчики игр в своих исследованиях объясняют тип или область алгоритма, включая методы фракталов, методы на основе грамматики, методы на основе поиска и эволюции, методы на основе ограничений, а также нарратив, ландшафт и создание подземелий.

Первый в мире успешный эксперимент по полностью автоматизированному дизайну настольных игр описывается в книге [2]. Эволюционные методы использовались для получения новых наборов правил в рамках настраиваемого языка описания игр, а испытания самостоятельной игры использовались для оценки потенциала каждой производной игры по интересу игроков-людей.

В сборнике [3] представлены статьи авторов, освещающих тему процедурной генерации как метода построения сюжета, создания персонажей и диалогов в видеоиграх. Игры – это интерактивный вид искусства, который предполагает тесное взаимодействие авторов, игроков и компьютера как средства генерирования объектов и сюжетных ходов. В своих статьях известные разработчики компьютерных игр описывают инструменты построения игрового мира, создания характеров, тем и общей атмосферы, приводя в пример собственные проекты и рассказывая о том, как преодолевались трудности в работе над ними.

Обзор исследований, посвященных проблеме процедурной генерации контента, представлен в статье [4]. Авторами описываются алгоритмы генерации игровых правил и игр различных жанров, в том числе алгоритмы на основе парадигм эволюционного моделирования и логического программирования, а также способы автоматической оценки генерируемых игр. Кратко рассмотрены первые системы генерации игр, реализованные в универсальных игровых программах. Также описаны различные форматы представления и специализированные языки для описания игровых правил в подходящем для алгоритмической обработки виде.

Среди огромного количества Интернет-ресурсов хотелось бы обратить внимание на следующие. Так, в цикле статей [5] пошагово рассматриваются несколько алгоритмов генерации помещений, выявляются их области применения. В работе [6], опубликованной на Gamasutra, объясняется методика генерации случайных подземелий, впервые описанная разработчиком игры TinyKeep на Reddit, причем все шаги рассмотрены достаточно подробно и приведена демонстрация алгоритма. Среди англоязычных электронных ресурсов по данной тематике хотелось бы отметить работу [7]. В этом посте описываются два алгоритма создания сложных процедурных миров из простых наборов цветных плиток, включая ограничения на размещение этих плиток. Показано, что, тщательно спроектировав наборы плиток, можно создавать интересный процедурно генерируемый контент, например, пейзажи городов или подземелий со сложной внутренней структурой. На видео приведена система, создающая процедурный мир на основе правил, закодированных в 43 цветных плитках. В блоге [8] собрано и проанализировано несколько основных методов создания карт. Естественно, ни один метод не подходит для всех целей, поэтому большинство разработчиков в

конечном итоге будут настраивать выбранные методы, с целью максимального соответствия потребностям разрабатываемой игры. Также приведены связи с источником каждого изображения, иллюстрирующего метод, переходя по которым можно найти объяснение и алгоритм построения.

Автор работы [9] попытался реализовать процедурную генерацию помещения (экстерьер) вместе с созданием плана помещения, внутренним интерьером и расстановкой предметов в здании. Основные отличия от похожих работ заключаются в том, что автор выделяет жизненно важные виды комнат и жизненной важные виды предметов для комнаты. В работе [10] проведен анализ существующих подходов виртуальной визуализации, основанных на ручном и процедурном отражении, по характеристикам адекватности требованиям и уровню автоматизации. Автор проинспектировал все известные алгоритмы и подходы использования процедурной генерации сооружений и делает вывод о том, что процедурный подход наиболее перспективный в плане автоматизации, но имеет недостатки в плане детализации.

Автор работы [11] представил метод быстрой процедурной генерации правдоподобных зданий с помощью их контуров. Известно, что существует несколько методов, различающихся по скорости и количеству деталей в результатах. Метод, представленный в этой работе, предпочитает скорость и полуавтоматический подход. При его применении контуры здания используются в качестве входных данных вместе с некоторыми параметрами, которые задает пользователь. Автор научной работы [12] попытался реализовать генерацию плана помещения. Входными данными для его программы являются контуры здания и выходы из нее (окна, двери), после введения которых на основе контуров здания создается подобие сетки для размещения комнат. Далее выбираются ячейки с выходами наружу, в которых помещаются комнаты, с последующим увеличением комнаты на одну клетку методом итераций, на заключительном этапе для получения доступа ко всем комнатам создается коридор.

Авторы доклада [13] представили вычислительную модель для процедурной генерации фасадных и внутренних стилей зданий для использования в трехмерных виртуальных средах игр и симуляций.

Цель исследования

После проведенного исследования проблем процедурной генерации игрового контента и существующих вариантов создания планов зданий автоматизированным образом, созрела необходимость спроектировать и создать собственную систему, с помощью которой можно создавать планы помещений в реальном времени. Объектом исследования является структура самого здания, как явления, а именно архитектурные особенности, планировка здания и прочее. Реализация процедурного генерирования помещений должна позволять генерировать помещения с возможностью изменения таких настроек, как количество этажей, тип крыши, моделей стен, окон и прочее. Это поможет увеличить вариативность зданий, как внешне (формы зданий), так и внутренне (плана помещения), а также уменьшить количество потраченных человеко-часов при создании более детализированного игрового объекта здания.

Изложение основного материала исследований

Целью использования процедурной генерации является создание игрового контента без участия человека (что не только менее затратно, но и оказывает помощь геймдизайнеру в решении задач), разработка других типов игр (улучшение показателей разнообразия и реиграбельности), адаптация игр под игрока «на лету», улучшение контента с помощью алгоритмических решений, а также формализация геймдизайна в

качестве научной задачи. Среди целевых свойств генерации обычно рассматриваются скорость, надежность, контролепригодность, разнообразие и креативность.

Программная система процедурной генерации состоит из двух основных модулей, первая из которых реализует генерирование 2D плана дома и включает в себя следующие компоненты: подготовка формы здания, нахождение нужной площади, планирования различных этажей здания, генерирования квартир и комнат. Во втором модуле реализована 3D визуализация сгенерированных 2D планов домов посредством использования гибких инструментов для отображения 3D объектов игрового движка Unity3D. Также добавлен модуль работы с полигонами, который выполняет поиск площади многоугольника и место нахождения точки по отношению к границам полигона.

Реализация алгоритма построения плана помещения базируется на применении метода роста комнат с помощью оценок подходящих кандидатов, который предложен в статье [12]. Идея этого алгоритма заключается в разбиении контура здания на зоны в виде сетки, в которые можно положить комнаты для дальнейшего роста. В случае алгоритма выращивания комнат построенная сетка основана на входных параметрах окон и углах здания, или же, при отсутствии их, строится с фиксированным шагом. К реализации алгоритма контроля площади здания добавлена возможность контроля площади для прямоугольного многоугольника. Для вычисления площади применяется алгоритм триангуляции [14] для входящего полигона с последующим вычислением площади треугольников. Реализованное здание панельного типа имеет форму прямоугольника, в качестве входных данных которого задается количество подъездов и этажей. На основе этого выбирается наиболее длинное ребро прямоугольника и делится на n + 2 точек, где n - количество подъездов, с помощью которых образуются прямоугольники подъездов. Добавление входа в подъезд имеет похожее решение, но в данном случае у самого длинного ребра находится точка середины, которая и является входом в здание. Поскольку коридор и лестница должна иметь конкретное место у входа, то алгоритм реализации добавления коридора, лестницы, лифта статически добавляет лестницы к входу как комнату 2x6 метров, коридор добавляется как горизонтальная комната к лестнице размером 6x2 метра, лифт был добавлен как одна комната 2x2 метра.

Программная система при заполнении настроек дома добавляет линк-соединение между комнатами, на основании которого строятся двери между комнатами в случайной общей стене. Первый вариант реализации крыш осуществлен на основе этажа с выходом на крышу. Для второго вида крыши использован алгоритм Straight skeleton [15], который может работать с любыми двумерными полигонами, например, обычными и с внутренними полостями. Это открывает возможности создавать различные варианты крыш, которые будут выглядеть довольно реалистично. Также была добавлена возможность комбинирования крыш различных типов, а именно каскадного и плоского типов. Примеры

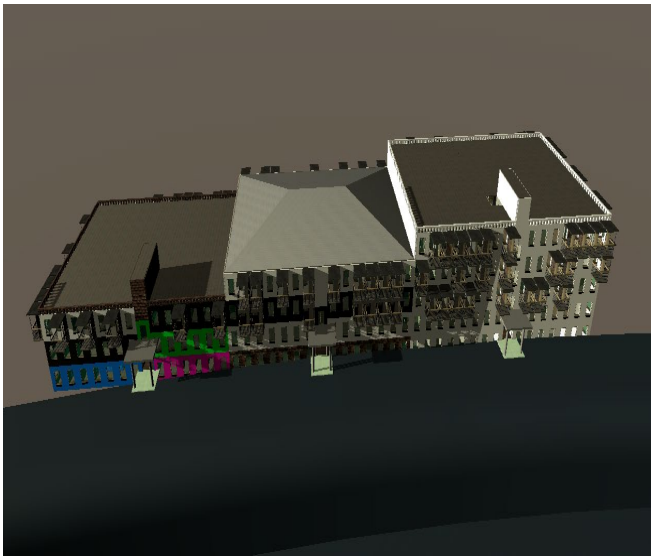

Рис. 1. Отображение комбинирования крыши

визуализации работы алгоритма изображены на рисунке 1.

Сейчас на рынке 3D моделирования существует несколько популярных решений, среди которых выделяют 3Dmax и 3D редактор Blender, применение которого обусловлено наличием открытого кода, а также современные инструменты для создания очень сложных 3D моделей. После окончания процесса построения 2D плана помещения происходит отражение в 3D. Программная система на предыдущем этапе генерирует положение планов, комнат, квартир у стены, конца стены, центра стены в 2D координатах, однако процесс 3D-отображения имеет ряд нюансов. Для корректного отображения в 3D все модели при импорте в Unity должны быть ориентированы в одном направлении, поскольку в зависимости от направления стены ее необходимо поворачивать на 90, 180, 270 градусов соответственно. Корректный импорт модели должен иметь следующее расположение световых векторов: синий (forward) вектор направлен во внутреннюю сторону модели, красный (right) вектор направлен справа от модели, зеленый (up) вектор направлен вверх модели. На рисунке 2 изображена импортируемая модель.

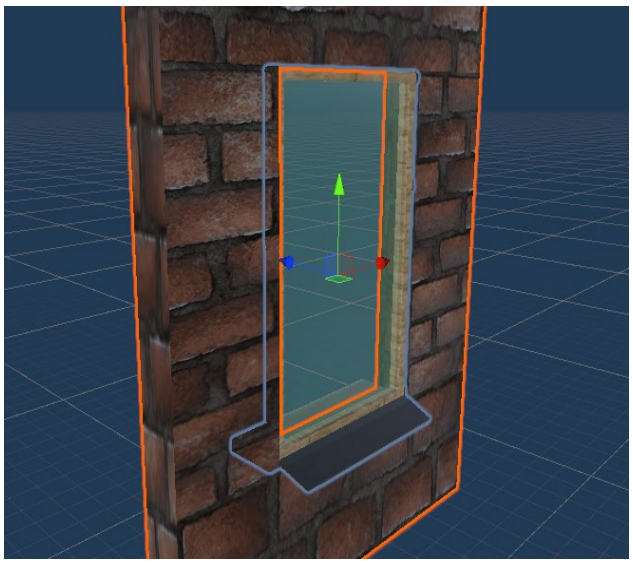

Рис. 2. Корректный импорт модели с правильным расположением векторов модели

Расположение лестницы и продолжение этой лестницы на следующих этажах имеет свои особенности и ограничения. Во-первых, размер модели должен подчиняться критериям: 2 метра в ширину, 6 метров в длину и 5 метров в высоту. Данное ограничение необходимо для того, чтобы лестница могла состыковаться при добавлении ее на следующий этаж. Проблема расположения лестницы в комнате решается благодаря нахождению центра комнаты и расположения лестницы в этой точке. Этот элемент также модульный, как и стены, и может быть изменен по усмотрению пользователя.

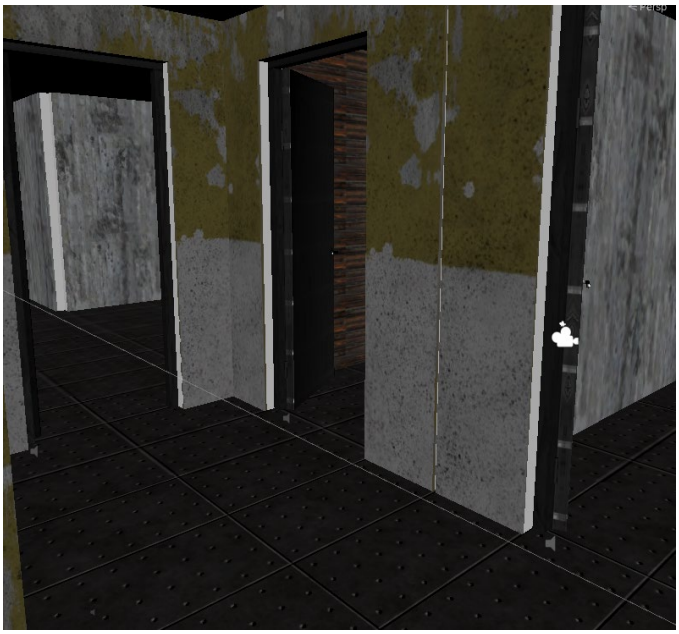

Рис. 3. Сгенерированный объект – коридор с дверьми, ведущими в комнату

В результате была разработана система процедурного генерирования дома, которая способна генерировать помещения с достаточно большой вариативностью. При использовании программной системы ее параметры можно изменять, начиная от самых общих настроек (количество этажей,

площадь, количество подъездов) до мельчайших параметров, таких как настройка высоты комнаты, материалы этажа, квартиры, комнаты, а также возможность изменять все компоненты (стены, лифт, лестницы, двери). Процесс генерирования здания занимает некоторое время, что может очень сильно ударить по быстродействию игры, поэтому разработанная программная система больше подходит для применения в режиме редактирования игры. Сгенерированное помещение изображено на рисунке 3.

Вопросы оптимизации, связанные с реализацией процедурной генерации здания с помощью модульных стен, решаются с помощью двух эффективных подходов. В первом варианте применяют объединения меш-сетей объектов. С помощью этого метода все модели стен, окон, полов объединяются в одну меш-сетку с сохранением материалов на них. Использование этого подхода при большом количестве объектов позволило увеличить частоту кадров почти в 8 раз. Дополнением к оптимизации меш-сетей является правильное моделирование 3D объектов, без лишних точек на модели и с минимизацией плоскостей на ней. Вторая проблема связана с быстродействием создания здания, решение которой связано с применением партерную проектирования Object pool. Сущность его заключается в хешировании некоторого количества объектов в оперативной памяти с последующим использованием в случае необходимости. Данный подход является эффективным в случае, когда к жесткому диску уходит довольно много запросов, поскольку операции с диском вызывают длительные задержки по времени из-за загрузки центрального процессора. Этот вариант будет эффективно работать, поскольку меш-сети реальных объектов, которые были созданы перед объединением, не уничтожаются и используются повторно.

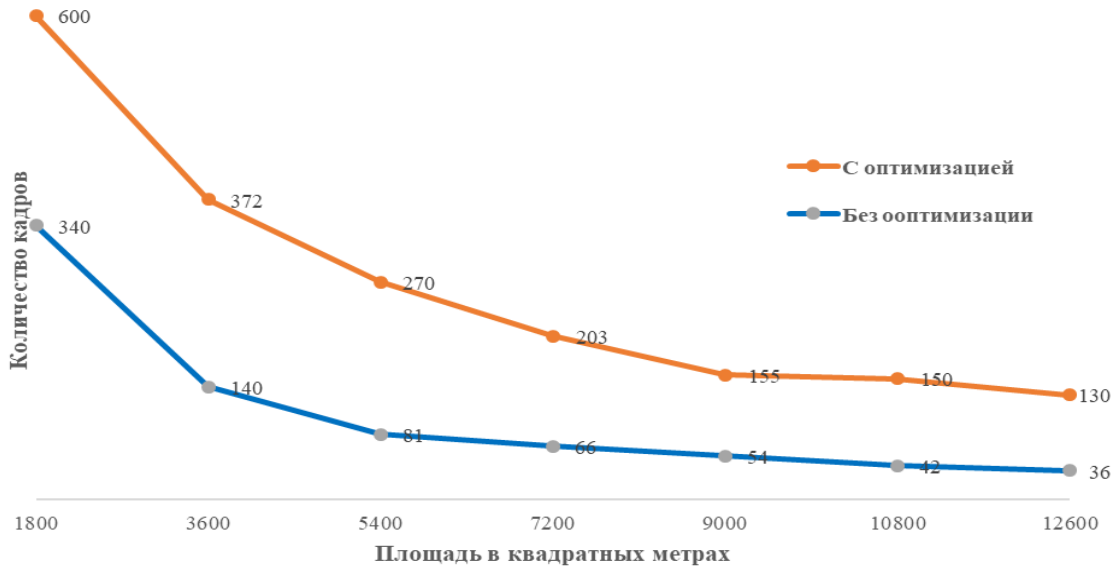

Рис. 4. Зависимость количества кадров от площади здания с учетом меш-сети

Результат оптимизации построения процедурно созданного здания приведен на рис. 4. В случае объединения элементов здания в единую меш-сетку наблюдается существенное увеличение частоты кадров.

Выводы

Проблема процедурного генерирования, которая представлена в работе – процедурное генерирование помещений. Основная идея заключается в том, что программа имеет возможность генерировать помещения с большим интервалом изменения настроек, начиная от количества этажей, изменения типа крыши до

настройки генерирования плана здания, изменения моделей стен, окон и прочее. Это поможет увеличить вариативность зданий, как внешне (формы зданий), так и внутренне (план помещения), а также уменьшить потраченные человеко-часы при создании более детализированного игрового объекта здания. Был проведен анализ предметной области, на основе которого разработана программная система, способная генерировать здания с внутренней планировкой этажей. На основе этих данных в будущем возможна разработка игр, например, для генерирования процедурно созданного города или создания моделей городов.

Список использованной литературы

1. Togelius J., Shaker N., Nelson M. Procedural Content Generation in Games: A Textbook and an Overview of Current Research, Berlin: Springer, 2014. 142 p.
2. Browne C. Evolutionary Game Design, Berlin: Springer, 2011. 122 p.
3. Шорт Т. Х., Адамс Т. Процедурная генерация в гейм-дизайне, М.: ДМК Пресс, 2020. 344 с.
4. Меженин М.Г. Обзор систем процедурной генерации игр. *Вестн. ЮУрГУ. Сер. Выч. матем. информ.*, 2015, т. 4, вып. 1, с. 5–20.
5. Процедурная генерация уровней. URL: https://habr.com/post/418685/ (дата обращения: 08.08.2021).
6. Алгоритм процедурной генерации подземелий. URL: https:// https://www.progamer.ru/dev/procedural-dungeon-generation.htm (дата обращения: 08.08.2021).
7. Dykeman I. Procedural Worlds from Simple Tiles. URL: https://ijdykeman.github.io/ml/2017/10/12/wang-tile-procedural-generation. html (дата обращения: 08.08.2021).
8. Procedural Map Generation. URL: https://www.gridsagegames.com/ blog/2014/06/procedural-map-generation/ (дата обращения: 08.08.2021).
9. Andersson S. Detailed Procedurally Generated Buildings: master's work / Sweden. Linköping University, 2019. 109 p.
10. Pádua L. Procedural Modeling of Buildings Composed of Arbitrarily-Shaped Floor-Plans: Background, Progress, Contributions and Challenges of a Methodology Oriented to Cultural Heritage. *Journals Computers*. Vol. 8, Issue 2, 2019. P. 1-10.
11. Kužel V. Procedural building reconstruction from building outlines: master's work / Prague. Charles University, 2018. 98 p.
12. Camazzato D. A. Method for growth-based procedural floor plan generation: master's work / Rio Grande. Pontifical Catholic University of Rio Grande do Sul, 2015. 132 p.
13. Silveira I., Camozzato D., Marson F. Real-time Procedural Generation of Personalized Façade and Interior Appearances Based on Semantics. *Conference 14th Brazilian Symposium on Computer Games and Digital Entertainment.* (Brazilian, 2015). Brazilian, 2015. C 1-8.
14. Robert E., Wyk V. Christopher J. An O(n log log n)-time algorithm for triangulating a simple polygon, *SIAM Journal on Computing.* 1988. Vol. 17, № 1. P. 143–178.
15. Aichholzer O., Aurenhammer F., Alberts D. A novel type of skeleton for polygons. *Journal of Universal Computer Science*. 1995. №1. P. 752–761.

References

1. Togelius, J., Shaker, N. & Nelson, M. (2014). Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Berlin: Springer.
2. Browne, C. (2011). Evolutionary Game Design. Berlin: Springer.

3. Short, T. & Adams, T. (2020). Short T. H., Adams T. Protsedurnaya generatsiya v geym-dizayne. (Procedural generation in game design). M.: DMK Press.
4. Mezhenin, M.G. (2015). Obzor sistem protsedurnoy generatsii igr. (Review of procedural game generation systems). *Vestn. SUSU. Ser. Subtract. mat. Inform*., **1,** 4, 5–20.
5. Protsedurnaya generatsiya urovney (Procedural level generation). Retrieved from: URL: https://habr.com/post/418685/.
6. Algoritm protsedurnoy generatsii podzemeliy (Algorithm for procedural dungeon generation). Retrieved from: URL: https:// https://www.progamer.ru/dev/procedural-dungeon-generation.htm.
7. Dykeman, I. Procedural Worlds from Simple Tiles. Retrieved from: URL: https://ijdykeman.github.io/ml/2017/10/12/wang-tile-procedural-generation.html.
8. Procedural Map Generation. Retrieved from: URL: https://www.gridsagegames.com/blog/2014/06/procedural-map-generation/.
9. Andersson, S. (2019). Detailed Procedurally Generated Buildings [master's work]. Sweden: Linköping University.
10. Pádua, L. (2019). Procedural Modeling of Buildings Composed of Arbitrarily-Shaped Floor-Plans: Background, Progress, Contributions and Challenges of a Methodology Oriented to Cultural Heritage. *Journals Computers*. **2,** 8, 1-10.
11. Kužel, V. (2018). Procedural building reconstruction from building outlines [master's work]. Prague: Charles University.
12. Camazzato, D. (2015). Method for growth-based procedural floor plan generation [master's work]. Rio Grande: Pontifical Catholic University of Rio Grande do Sul.
13. Silveira, I, Camozzato, D. & Marson F. (2015). Real-time Procedural Generation of Personalized Façade and Interior Appearances Based on Semantics. *Conference 14th Brazilian Symposium on Computer Games and Digital Entertainment*. (Brazilian, 2015). Brazilian. pp. 1-8.
14. Robert, E., Wyk, V. & Christopher J. (1988). An O(n log log n)-time algorithm for triangulating a simple polygon, *SIAM Journal on Computing.* **1,** 17, 143–178.
15. Aichholzer, O., Aurenhammer, F. & Alberts, D. (1995). A novel type of skeleton for polygons. *Journal of Universal Computer Science*. **1**, 752–761.

Михайлуца Елена Николаевна – к.т.н., доцент, доцент кафедры программного обеспечения автоматизированных систем Запорожского национального университета, e-mail: elenamikhaylutsa7@gmail.com, ORCID: 0000-0003-2935-7997

Пожуев Андрей Владимирович – к.ф.-м.н., доцент, заведующий кафедры общеобразовательных дисциплин Запорожского национального университета, e-mail: scorpio6828@gmail.com, ORCID: 0000-0002-4083-5139

Волик Сергей Александрович – магистр кафедры программного обеспечения автоматизированных систем Запорожского национального университета