**Федеральное государственное автономное образовательное учреждение   
высшего образования**

**КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ИНФОРМАЦИОННЫХ СИСТЕМ**

Направление подготовки: 09.03.03 – Прикладная информатика

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**«Разработка симуляции разрушения зданий при взрывах»**

**Работа завершена:**

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

Студентка группы \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_Г. С. Съемщиков

**Работа допущена к защите:**

Научный руководитель

ст. преподаватель Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Директор Высшей школы ИТИС

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Казань – 2017 г.

**Содержание**

[Введение 3](#_Toc482996532)

[Глава 1. Анализ проблемы и постановка задачи 6](#_Toc482996533)

[1.1. Анализ предметной области; 6](#_Toc482996534)

[1.2. Обзор существующего программного обеспечения 12](#_Toc482996535)

[1.3. Формальная постановка задачи 17](#_Toc482996536)

[Глава 2. Основная часть 19](#_Toc482996537)

[2.1. Используемые технологии 19](#_Toc482996538)

[2.2. Разработка математической модели 25](#_Toc482996539)

[2.3. Практическая реализация симуляции разрушения зданий 27](#_Toc482996540)

[2.4. Работа приложения 35](#_Toc482996541)

[Заключение 41](#_Toc482996542)

[Список литературы 42](#_Toc482996543)

**Введение**

Согласно статистике, за последние 42 года в мире произошло более 113000 террористических актов, из которых почти 52000 были связаны с использованием взрывных устройств. Уровень террористической угрозы в России постоянно растет, в 2001-2010 годах произошло почти в 6 раз больше терактов, чем в 1991-2001 и их число продолжает расти (рис. 1) [1]. В связи с этим особую актуальность приобретают исследования, связанные с расчетом масштабов разрушений и потенциального ущерба, который может быть нанесен населенным пунктам при детонации взрывных устройств.

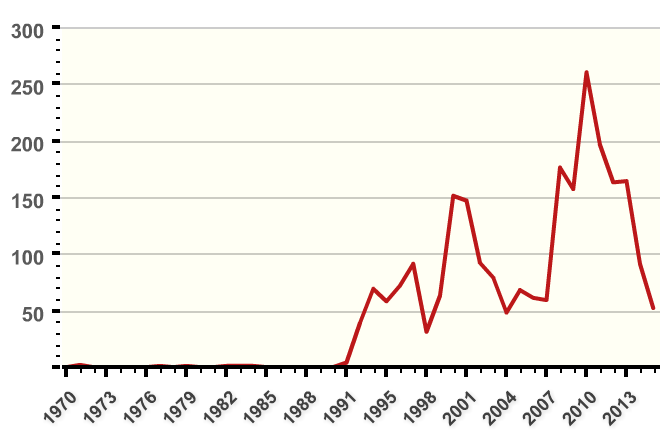


Рис. 1. Число террористических актов в России за последние годы [1].

Предметом данной работы является симуляция разрушения зданий в экстремальных ситуациях, а именно при взрывах бомб, заложенных в определенном месте населенного пункта.

В настоящее время на рынке программных продуктов в свободном доступе практически нет программ, способных рассчитать и произвести графическое моделирование взрыва бомб различной мощности в определенном районе населенной местности. Существует сервис NUKEMAP, созданный на основе Google Maps, позволяющий рассчитать последствия ядерного взрыва по заданным условиям [2]. Однако, этот сервис обладает рядом недостатков, среди которых невысокая точность расчетов, отсутствие возможности 3D моделирования взрыва бомбы в реальном времени. Кроме того, данный сервис не позволяет рассчитать последствия взрыва неядерных бомб, а получаемая с его помощью информация является интегральной по местности, то есть нельзя определить какой ущерб получит каждое здание в отдельности. Другие существующие программные продукты также являются ограниченными. Они или позволяют только рассчитать зону поражения, или только провести 3D моделирование разрушения зданий, но в свободном доступе нет программных продуктов, способных производить детальную симуляцию разрушений зданий в зависимости от типа и мощности взрывного устройства, и расчет зоны поражения одновременно. Связи с этим тема дипломной работы безусловно является практически значимой и удовлетворяет требованиям новизны.

Целью данной работы является создание программного продукта, способного производить симуляцию разрушения зданий при взрыве бомбы заданной мощности в определенном районе города.

В задачи работы входит:

* изучение способов расчета разрушения зданий и сооружений при взрывах, выбор подходящего математического аппарата для осуществления моделирования последствий взрыва бомбы;
* моделирование зданий и сооружений определенного района города с помощью сервиса OpenStreetMap;
* задание различной взрывоустойчивости зданий на основании требований государственных стандартов по безопасности;
* расчет зоны поражения на основе выбранной теоретической базы;
* изучение инструмента графического моделирования и программирования Unity 3D и реализация с его помощью симуляции разрушения зданий при взрыве бомбы заданной мощности в определенном районе города в реальном времени.

Графическое моделирование реализовано с помощью программного пакета Unity 3D. Unity 3D представляет собой современный кросс-платформенный движок для разработки 2D и 3D приложений и игр, которые поддерживают DirectX и OpenGL и работают под различными операционными системами – Windows, OS X, Android, Apple iOS, Linux, а также на игровых приставках Wii, PlayStation 3 и Xbox 360. Среди преимуществ Unity стоит выделить возможности импорта, тестирование приложения непосредственно в редакторе, кросс-платформенность, инструментарий для совместной разработки, гибкость и масштабируемость, а также ценовая доступность и наличие бесплатной версии с большим количеством функций. Для изучения Unity 3D использовались литературные источники [3-7].

Моделирование района города производилось с помощью OpenStreetMap [8]. OpenStreetMap – это некоммерческий веб-картографический проект с подробной свободной и бесплатной географической карты мира.

Математический аппарат для расчетов разрушений зданий и сооружений был выбран на основе статьи Деркачева Н.В. и соавт. [9], а механизм моделирования сколов и осколков разрушенных зданий был разработан на основе алгоритма, описанного в работе Рыженко А.А. [10].

Структура работы включает введение, две главы, заключение, список использованной литературы и приложения.

**Глава 1. Анализ проблемы и постановка задачи**

* 1. **Анализ предметной области;**

Необходимо разработать комплексное решение для симуляции разрушения зданий и местности от детонации взрывчатых устройств различной мощности. Для корректной работы приложения необходимо реализовать следующие функции и визуальный эффекты:

* на языке С# написать код программы (Explosion.cs), отвечающий за физическую составляющего взрыва. В данном скрипте будут объявлены две локальные переменные типа float: ExplRadius (хранит значение радиуса взрыва), ExplForse (хранит значение силы взрыва в условных единицах). Данный код будет работать во время проигрывания сцены. В первую очередь будет определена зона поражения и объекты, входящие в нее. Одним из вариантов решения является создание невидимой сферы с радиусом равным радиусу взрыва, который возвращает переменная ExplRadius. Второй вариант реализации заключается в использовании множества лучей определенной длины (как правило, используется значение радиуса взрыва), исходящих из одной точки. На практике оба варианта дают схожий результат, поэтому будет использован вариант со сферой, так как он является более быстрым и экономичным с точки зрения использования ресурсов компьютера. Последнее представляет собой крайне важный критерий, поскольку на следующих этапах будут выбраны более ресурсоемкие решения для достижения корректной симуляции взаимодействия различных физических объектов, присутствующих на созданной сцене, что несомненно повышает требования к вычислительным машинам. Принимая также во внимание то, что после взрыва здания будут фрагментированы на большое количество осколков, взаимодействующих между собой, для более наглядной и правдоподобной симуляции, необходимо выбирать экономичные решения в случаях, когда на общую картину значимого негативного влияния они не окажут. После инициализации объектов, входящих в зону поражения взрывного в устройства, необходимо определить удаленность каждого отдельного объекта от эпицентра взрыва, для того, чтобы рассчитать силу взрыва на нем, которая со временем t после начала взрыва будет уменьшаться. Вычислить расстояние между двумя объектами можно путем вычитания координат объекта и центра взрыва. В результате будет возвращен вектор, направленный от центра взрыва к объекту. Дистанцией будет являться модуль итогового вектора. С самим вектором работать не рекомендуется, поскольку он склонен нагружать процессор, так как при каждом вызове происходит вычисление квадратных корней. Поскольку для решения поставленной задачи достаточно знать только расстояние между объектами для сравнения, то можно ограничиться возвращением свойства sqrMagnitude. Полученное значение будет записываться в локальную переменную типа float distance и дальше будете передан в метод CanBeDestroyed();
* на языке С# написать код программы (CanBeDestroyed.cs), отвечающий за частичное или полное разрушение зданий в зависимости от мощности взрыва и порогового значения взрывоустойчивости объекта. При этом нужно учитывать то, что сила воздействия взрывной волны будет уменьшаться по мере отдаления от эпицентра взрыва. Таким образом, необходимо ввести в данный метод локальную переменную, в которой будет храниться значение расстояния между объектом и взрывным устройством. Это необходимо для того, чтобы оценивать в какую из зон поражения попадает исследуемый объект. От этого будет зависит сила воздействия на объект и, как следствие, результат симуляции. Таким образом, данный скрипт должен заменять пораженные части на фрагментированные копии, а также производить их настройку с использованием циклов.
* с использованием объекта движка Unity3d «Particle System» создать визуальную составляющую взрыва, а именно анимацию расширяющейся сферы с текстурой огня (Fireball), взрывной волны (Shokwave), а также дыма от взрыва (BaseSmoke);
* вспомогательные скрипты на языке C#, которые в автоматическом режиме во время взрыва или в процессе настройки сцены, добавляют компоненты Mesh Collider c опцией Convex и Rigidbody с активным пунктом Use Gravity.

Стоит отметить, что использование экономичного с точки зрения используемых ресурсов системы Box Collider недопустимо, поскольку при замене объекта, на его фрагментированную копию, велика вероятность того, что отдельные части объекта начнут отталкиваться друг от друга (рис. 1.1.1). Такое решение является эффективным только для реализации проектов, где физика явлений не играет важную роль.

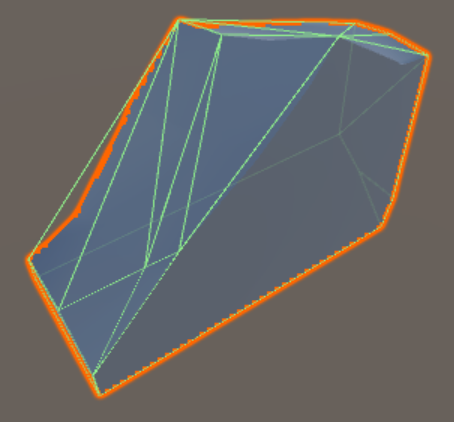
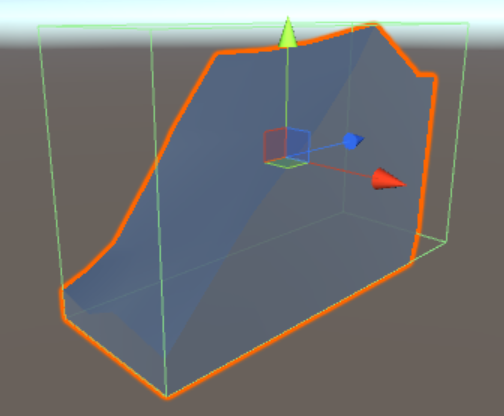
 

Рис. 1.1.1. Иллюстрация различных типов компоненты Collider. Слева Mesh Collider, справа Box Collider.

Решением является использование параметра Collider типа Mesh collider (рис. 1.1.1), который создает представление взаимодействия на основе сетки Mesh, прикрепленной к объекту, а также на основе свойств привязанной компоненты Transform. Ключевым преимуществом данного подхода является то, что он позволяет создать форму компоненты Collider в точности такой, какой является видимая сетка объекта. Как следствие, столкновения в проигрываемой сцене становятся более реалистичными. Однако, такая точность является более ресурсоемкой, нежели при использовании примитивных типов компоненты Collider. Таким образом, в случае Mesh collider фрагментированный объект в состоянии покоя будет сохранять свой внешний вид до тех пор, пока к отдельным частям объекта не будет приложена внешняя сила. Для взаимодействия фрагмента, описываемого с помощью «Mesh collider», необходимо активировать пункт «Convex». Это можно сделать как вручную в инспекторе объекта, так и с помощью С# скрипта. Сложности возникают также при взаимодействии объектов, обладающих большой скоростью перемещения. В частности, мелкие частицы, оказывающиеся в эпицентре взрыва, могут пролетать сквозь другие объекты на сцене. В этом случае также необходимо прибегнуть к более ресурсоемким решениям. В частности, проблему можно устранить отказом от дискретного режима «Collision Mode» в пользу CollisionDetectionMode.ContinuousDynamic. Такой выбор обусловлен тем, что в данном режиме столкновения как со статичными, так и с подвижными объектами на сцене рассчитываются в каждый момент времени, а не дискретно, через промежуток времени равный Time.fixedDeltaTime [11]. Таким образом исключается возможность просачивания быстродвижущегося объекта сквозь другие статичные или динамичные объекты на сцене. В предлагаемом решении указанные параметры будут устанавливаться автоматически. Необходимые настройки описаны в вышеупомянутых C# скриптах.

Одним из недостатков Unity3D является ограниченные возможности собственного 3D редактора. Как правило, модели для построения сцен предлагается выбирать либо из магазина дополнений «Asset store», либо в сторонних 3D редакторах: [11]

* Maya
* Cinema 4D
* 3ds Max
* Cheetah3D
* Modo
* Lightwave
* Blender
* Cheetah3D

Файлы можно импортировать в среду Unity с помощью универсальных форматов файлов, таких как .FBX или .OBJ в общем случае или в собственных форматах .Max и .Blend, если используется 3D Studio Max и Blender соответственно. При этом текстуры необходимо копировать в папку под названием Textures рядом с экспортируемой сеткой (mesh) в проекте Unity [11]. Рекомендуется использовать первый подход, который обладает рядом преимуществ:

* экспортируются только необходимые данные;
* проверяемые данные;
* как правило файлы меньшего размера;
* поддерживает модульный подход - к примеру разными компонентами для интерактивности и типов столкновений;
* поддерживает другие 3D пакеты, чьи форматы не поддерживаются в Unity напрямую.

Мощный 3D редактор необходим для более точного воссоздания зданий, которые будут тестироваться в симуляторе. К сожалению, автоматическое построение цельных прямоугольников соответствующей высоты на основе данных карт openstreetmap является некорректным, поскольку оно не позволит выявить наиболее уязвимые узлы зданий, тогда как проработка зданий в ручном режиме на основе тех же данных позволит провести более яркую демонстрацию возможностей программы.

Автоматическую фрагментацию объектов можно проводить как в 3dMax редакторе, так и в Blender. В первом случае можно использовать плагин RayFire(рис. 1.1.2), во втором, используя встроенные решения.

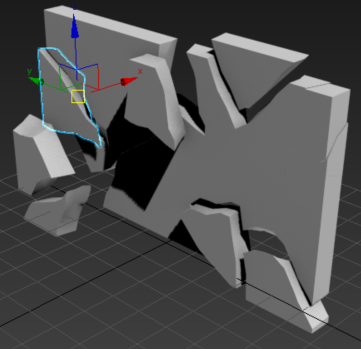
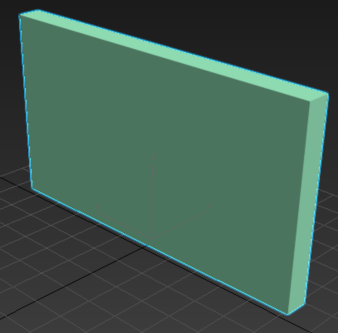


Рис. 1.1.2. Фрагментация объекта в 3d редакторе 3ds Max. Слева исходный объект, справа фрагментированный объект.

Таким образом, предлагаемая кроссплатформенная программа может быть использована обширной аудиторией, хотя основными целевыми группами являются:

* инженеры;
* архитекторы;
* сотрудники строительных компаний;
* сотрудники МЧС;
* аспиранты;
* студенты.

При работе с программой указанные группы пользователей будут иметь возможность размещать в любой точке сцене взрывное устройство, для которого в ручном режиме необходимо задать мощность взрыва. В результате пользователь увидит симуляцию в режиме «real-time» с анимацией взрыва и разрушения объектов. По окончании симуляции пользователь сможет наглядно оценить последствия взрыва зданий. Анализ результата позволит выявить наиболее уязвимые объекты на местности, а также оценить возможные риски.

Таким образом, можно заключить, что анализ предметной области помог выявить основные информационные объекты, с которыми придется работать при моделировании симулятора:

- информационные ресурсы о различных взрывчатых устройствах, описывающих зависимости радиуса поражения от мощности взрыва;

* ресурсы, описывающие государственные стандарты оценки взрывоустойчивости зданий и требования по взрывоустойчивости к основным типам построек (жилые дома разной этажности, торговые центры, офисные здания, магазины и т.д.);
* ресурсы о методах Unity3d, используемых при кодировании на языке С#.
  1. **Обзор существующего программного обеспечения**

На данный момент существует несколько готовых решений для симуляции взрывов. Наиболее распространенными из них являются NUKEMAP [2] и LS-DYNA [12, 15].

NUKEMAP представляет собой интерактивную карту, использующую программный интерфейс приложения Google Maps и данные по рассекреченным видам ядерного оружия. Подготовка к симуляции состоит из 3 шагов:

1. Выбор точки падения ядерной бомбы
2. Выбор вида ядерной бомбы
3. Выбор дополнительных опций (взрыв в воздухе, взрыв на поверхности, расчет радиоактивного облака и т.д.)

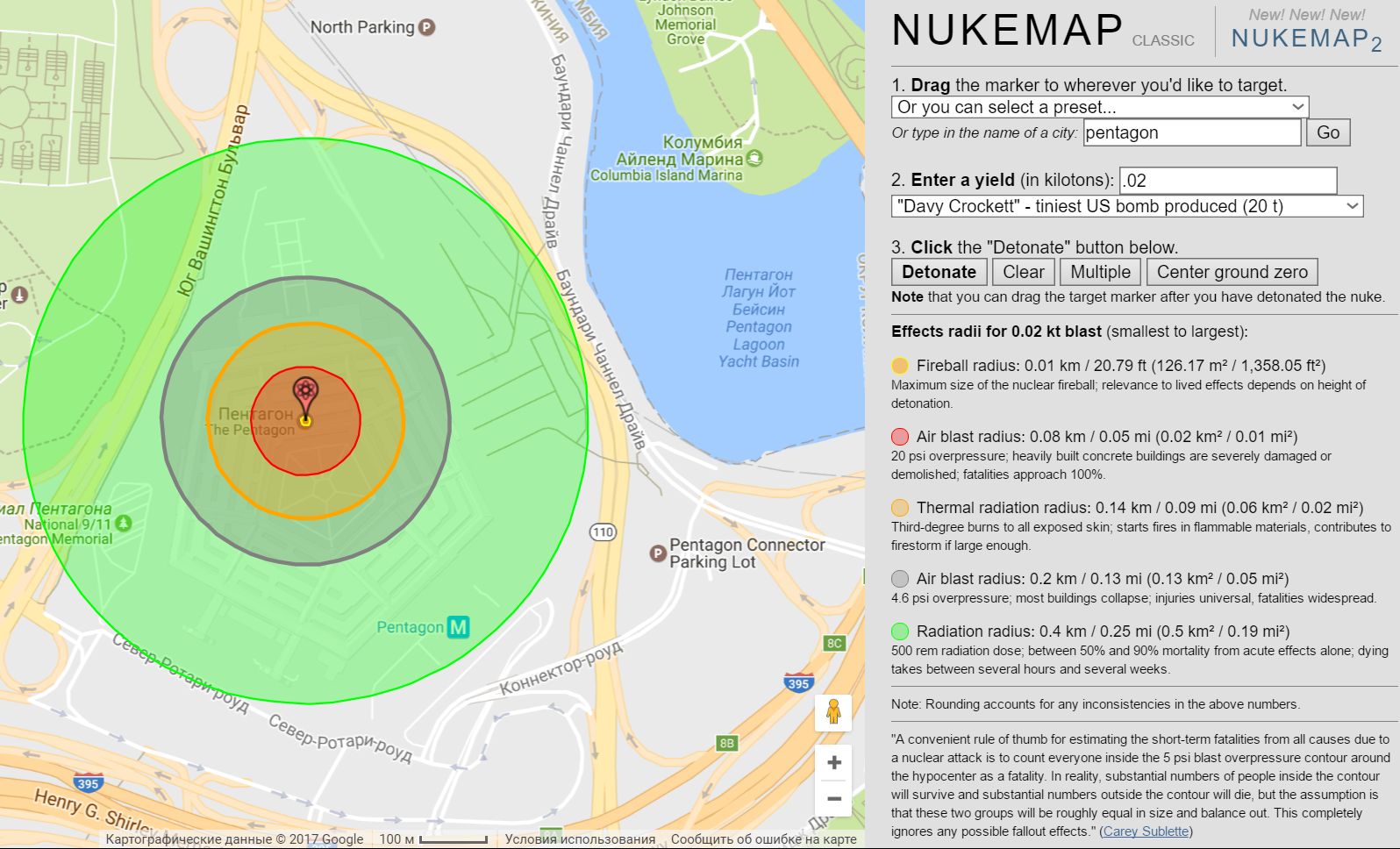


Рис. 1.2.1. Скриншот работы программы NUKEMAP.

После настройки симуляции пользователь нажимает кнопку Detonate. На карте иллюстрируются различные зоны поражения в виде окружностей разного цвета, а также приводится оценка количества пострадавших людей (рис. 1.2.1).

Схожим образом работает плагин NUKEMAP3D [13] для Google Earth, единственное различие заключается в том, что в данном случае на карте изображается объемный ядерный гриб, а также зона поражения описывается в виде полусферы, при этом здания на карте никак не разрушаются. Стоит отметить, что данный плагин сейчас неактивен в связи с решением компании Google отключить программируемый интерфейс приложения Google Earth из соображения безопасности [14].

Таким образом, можно заключить, что приложение NUKEMAP обладает рядом существенных недостатков. Ключевыми из которых являются:

1. Отсутствует возможность симуляции неядерного взрывного устройства;
2. Отсутствует возможность анализа воздействия взрыва на здания, входящие в зону поражения;
3. Отсутствует полноценная 3D анимация в режиме реального времени;
4. Оценка корректности анализа количества пострадавших людей является недостаточно благонадежной.

Более продуманным решением является программный пакет, предоставляемый компанией LSTC LS-DYNA [12,15,16]. Он позволяет моделировать взрыв с использованием взрывчаток разной мощности и исследовать воздействие взрывной волны на окружающие объекты [12].

LS-DYNA логичное продолжение программы DYNA3D, которую разработал доктор Джон О. Холквист в Ливерморской национальной лаборатории им. Лоуренса в 1976 г. Основная задачей программы состояла в симуляции воздействия ударной волны от взрыва ядерной бомбы различной мощности. На тот момент отсутствовали программы 3-хмерного моделирования, а готовые программы двухмерного моделирования давали достаточно неточные оценки результатов взрыва. Несмотря на то, что проект был отменен, разработка данного решения продолжилась параллельно с разработкой 2-х мерной версии программы DYNA2D [16].

Первую версию программы, написанную на языке Фортран, опубликовали в 1976 г. LS-DYNA предназначена для решения 3-хмерных нелинейных динамических задач механики твердого тела, ждикости и газа, а также связанных задач [16]. Она пользуется большим спросом в таких отраслях промышленности и науки, как автомобилестроение (симуляция краш-тестов), военно-промышленный комплекс (симуляция взрывов боеприпасов и их воздействие на окружающие предметы), авиа- и ракетостроение (проектирование реактивных двигателей и сопел) и так далее [16].

В LS-DYNA реализованы явный и неявный метод конечных элементов с возможностью построения лагранжевой, эйлеровой и гибридной сетки, многокомпонентная гидродинамика, бессеточный метод сглаженных частиц, бессеточный метод, основанный на методе Галеркина. Программа имеет встроенные процедуры автоматической перестройки и сглаживания конечно-элементной сетки при вырождении элементов, высокоэффективные алгоритмы решения контактных задач, широкий набор моделей материалов, возможности пользовательского программирования [12].

Практическое применение данная программа находит при исследовании механизма появления различного рода воронок после детонации взрывного устройства [17]. Для этого необходимо выбирать математическую модель грунта, позволяющую с большой точностью описывать особенности структуры почвы. Тогда результат может оказаться достаточно точным [17]. Из этого следует необходимость проводить симуляции при разных встроенных моделях грунта и дальнейший сравнительный анализ теоретических и экспериментальных данных.

Также данная программа может быть использована для симуляции подрывов различных транспортных средств, находящихся на различных участках дороги [12]. В результате, анализируются места трещины, реакции поверхности опоры, определяются уязвимые транспортные узлы. По полученным данным могут быть внесены изменения в конструкции дорог. Например, в случае моста можно добавить дополнительные демпферы или пружина, а также изменить саму структура моста [12].

Некоторые исследователи показывают, что для моделирования процесса детонации от взрыва необходимо использовать SPH метод, другие исследователи ALE (произвольный лагранжево-эйлеровый) метод. Другой особенностью программы является использование карты load\_blast\_enhanced для моделирования взрывов, которая не требует ALE и SPH методов [15]. Использование данной карты LS-DYNA позволяет значительно уменьшить время вычисления задачи (до 100 -500 раз). Однако эта карта хорошо работает лишь с известными марками взрывчаток, например, TNT [12].

Часто LS-DYNA применяется для исследования воздействия взрывной волны на различные объекты, например, воздействие на корпус корабля [12].

Моделирование взрыва позволяет создать методику для проектирования новых безопасных конструкций, например, кораблей, танкеров и т.п. Это касается как грузовых судов, так и пассажирских кораблей, поскольку новые методы проектирования, обусловленные применением программ математического моделирования, являются более надежными и спроектированная конструкция позволит выдержать большие нагрузки при чрезвычайных ситуациях. Главная цель моделирования в этом случае в предсказании разрушения или образование трещин, возникающих в конструкции. В большинстве случаев критерием разрушения в объемно-конечном моделировании принимается эквивалентная пластическая деформация.

Как правило, взрывное воздействие имеет сложный вид и использование моделирования позволяет более адекватно проектировать большие структуры, например, корабли.

Разрушение объектов в данном программном пакете происходит путем замены реального объекта на модель, состоящую из набора пластичных элементов. Во время моделирования анализируется эквивалентная пластическая деформация. Данный метод нельзя исследовать в случае, когда разрушение происходит под воздействием в нескольких направления [12]. Тогда как в случае разрушения зданий фрагменты стен, двигаясь под действием силы притяжения, силы действия взрывной волны могут сталкиваться друг с другом в результате чего направление движение объектов должно меняться при упругом столкновении.

В целом теоретический расчет в случае моделирования взрыва в условиях городской застройки обладает низкой точностью. Данное программное обеспечение специализируется на симуляции в конечно-элементных программах. При этом необходимо определенным образом подготавливать среду симуляции.

Также стоит отметить, что данный продукт является платным и, в первую очередь, рассчитан для продажи крупным организациям. Лицензия на продукт в основном продается лишь на год [15]. Таким образом такие группы пользователей, как студенты, аспиранты, а также сотрудники небольших организаций скорее всего не смогут позволить себе пользоваться данным предложением.

* 1. **Формальная постановка задачи**

С целью изучения возможных рисков при детонации взрывчатого устройства в условиях реальной среды создается программа, основанная на кроссплатформенном движке Unity3d с использованием скриптов, написанных на объектно-ориентированном языке С#. К симулятору взрывов предъявляются следующие требования:

- симуляция процесса взрыва в режиме реального времени;

- наличие анимации взрыва;

- наличие анимации разрушения объектов;

- возможность устанавливать взрывчатое устройство в произвольной точке на сцене;

- возможность задавать мощность взрыва в пользовательском интерфейсе.

Для реализации поставленных целей необходимо воссоздать взрывное устройство и провести симуляцию различных сценариев взрыва с использованием примитивных объектов движка Unity3d (куб, шар, плоскость и т.д.) разной массы.

После успешного завершения первой фазы проектирования необходимо приступить к симуляции частичного или полного разрушения стены в зависимости от мощности взрыва под действием взрывного устройства. Для этого необходимо создать и экспортировать в формат .FBX модель стены и фрагментированную копию данной модели в редакторе 3ds Max. Далее написать соответствующие скрипты на языке C# для реализации разрушения стены от взрыва. Так же необходимо учесть, что фрагменты стены имеют произвольный объем. Соответственно присвоение компоненты Rigidbody без указания массы объекта является некорректным, поскольку фрагменты объекта разного объема в этом случае будут иметь одинаковый показатель инертности. Как следствие, симуляция взаимодействия фрагментов окажется недостоверной.

Несмотря на то, что расчет объема и массы каждого фрагмента приведет к дополнительной нагрузке на процессор, учитывая количество объектов, он будет реализован в предлагаемом симуляторе.

Для демонстрации возможностей симулятора будет смоделирован случайно выбранный двор в Кировском районе г. Казань в редакторе 3ds MAX (рис. 1.3.1).

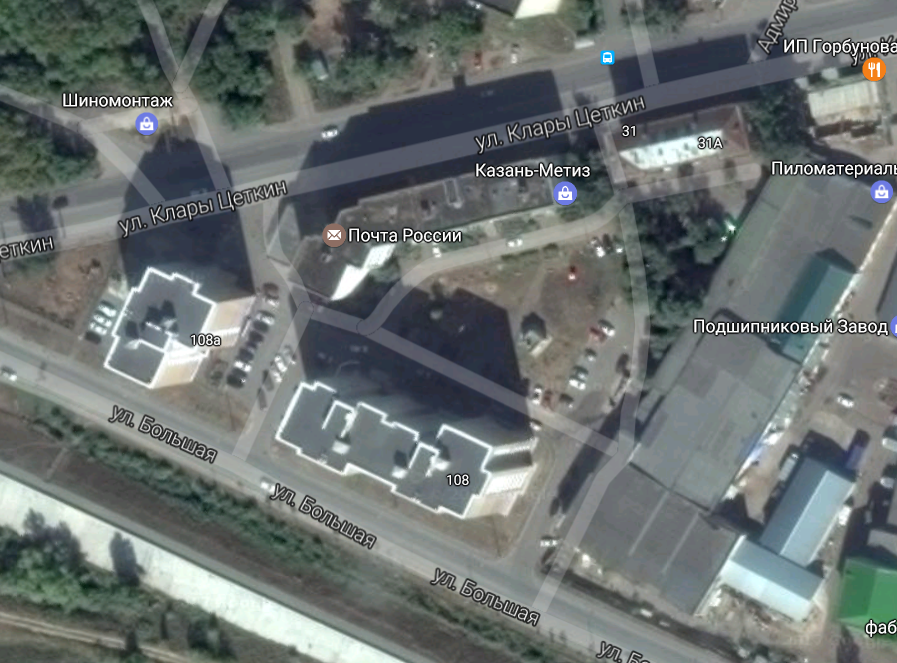


Рис. 1.3.1. Карта двора, выбранная для моделирования в редакторе 3ds MAX.

**Глава 2. Основная часть**

**2.1. Используемые технологии**

Залогом успешного и реалистичного моделирования результатов взрыва, в следствии детонации взрывного устройства или утечки газа, является высокая детализация моделей зданий и прочих объектов, подлежащих исследованию. В частности, здания должны быть представлены не в качестве обычных полых или сплошных блоков, а как минимум, сложные модели с учетом несущих стен и межэтажных плит. При этом необходимо использовать действительные планы зданий. Моделирование должно вестись либо с использованием реальных данных о габаритах отдельных элементов в СИ или любых других системах единиц. Важным является сохранение пропорций размеров элементов относительно друг друга. Таким образом, автоматическое создание зданий на основе карт openstreetmap видится некорректным решением, поскольку данные открытые карты хранят информацию об этажности зданий, при этом высота этажей не указана. Помимо этого, отсутствует информация о несущих стенах, колонах, ребрах жесткости и т.д. Более того, заявленная ранее целевая аудитория данного симулятора будет заинтересована в многостороннем изучение объектов своих исследований. В случае с утечкой природного газа можно поставить вопрос о расположении кухни относительно ключевых опорных сооружений для каждой отдельной квартиры на этажной площадке. Таким образом можно будет выбрать такие расположения, при которых разрушения от взрыва будут минимальным. Для этого пользователь должен обладать свободой моделирования объектов исследования.

Моделировать элементы сцены можно как во внутреннем 3D редакторе Unity, так и в сторонних. К сожалению, собственный 3D редактор Unity является достаточно ограниченным и неудобным. Он не приспособлен для моделирования сложных составных объектов. Основной его целью является выстраивание композиции из уже готовых моделей. Поэтому особое внимание разработчики уделили возможностям импорта 3D решений из сторонних сред разработки [11]. В частности, для таких всемирно признанных и популярных 3D редакторов как 3ds Max и Blender есть возможность импорта моделей из файлов с собственными форматами. В остальных же случаях разработчикам предлагается пользоваться форматом .FBX. Несмотря на то, что он является проприетарным недокументированным форматом [18], данный тип файлов обрел широкую популярность благодаря FBX Extensions SDK, доступному на языках C++ и Python. FBX может хранится на диске в виде бинарных или ASCII данных. FBX SDK поддерживает оба формата. Оба формата не задокументированы, однако формат ASCII имеет древовидную структуру с чёткими обозначениями идентификаторов [18].

Таким образом, пользователь может пользоваться любым привычным ему 3D редактор и в дальнейшем импортировать собственный модели в предлагаемый симулятор. Для целей демонстрации возможностей симулятора был выбран 3D редактор 3ds Max 2017, поскольку он обладает удобными и эффективными встроенными функциями, а также множеством полезных сторонних плагинов, таких как RayFire, GhostTown и т.д.

Моделирование начинается с создания плоскости, на которой выставляется, выбранный ранее кусок карты (рис. 1.3.1.). В первую очередь будет создана 3D модель дома №108, а после противоположный дом на ул. Клары Цеткин. На финальной стадии проектирования для данных моделей будут созданы фрагментированные копии с разным уровнем фрагментации, для демонстрации влияния количества осколков на реалистичность и точность разрушения. Для создания 3D модели дома №108 необходимо расположить карту на плоскости соответствующим образом (рис. 2.1.1.). Далее задачу моделирования можно решить двумя способами. Первый заключается в том, чтобы контур здания, то есть фасадные стены отметить на плоскости с помощью линий, которые должны замыкаться в определенной точке. Тогда пространство между двумя замкнутыми площадями можно заполнить и поднять на определенную высоту. Таким образом будут созданы несущие стены.

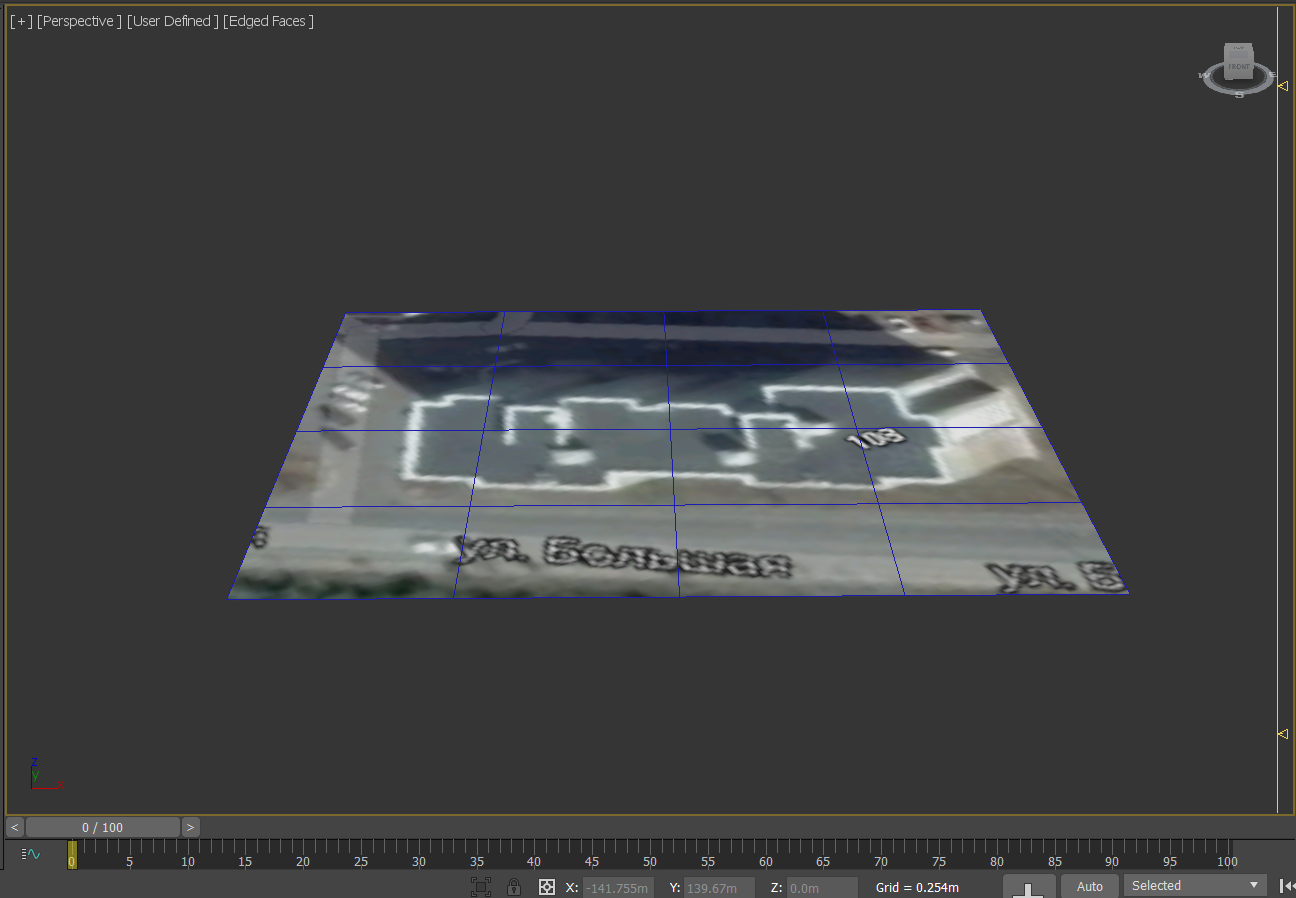


Рис. 2.1.1. Рабочая область программы 3ds Max 2017 c объектом «Плоскость».

Второй способ заключается в использовании плоскости, которую необходимо расположить в одном из углов здания. Тогда можно с помощью зажатой кнопки Shift создавать новую плоскость, выходящую из ребра предыдущей под любым углом. В этом случае замыкать плоскости в конечном итоге нет необходимости. После того, как схема здания будет воспроизведена на карте, необходимо преобразовать готовые плоскости в формат Editable Poly и в окне Modifier List выбрать функцию Extrude, которая превратит плоскость в куб или параллелограмм нужной высоты в метрах, если в 3ds Max выбрана система единиц СИ. Это можно сделать в окне Customize. После ряда несложных операций в редакторе создана основа модели здания (рис. 2.1.2.). Высота стен равна 3 метрам. Таким образом, это модель одного этажа, для которой требуется создать межэтажное перекрытие.

Необходимо отметить, что использование карты как подложки, не только облегчает процедуру моделирования, но также позволяет сохранить отношения габаритов зданий их отдельных элементов. В данном случае расстояния на плоскости как видно из рис. 2.1.1 задаются в метрах. В таком случае остальные параметры в симуляторе, в частности силу взрыва будет рассчитываться в реальных единицах. В противном случае, необходимо масштабировать силу взрыва и прочие параметры.

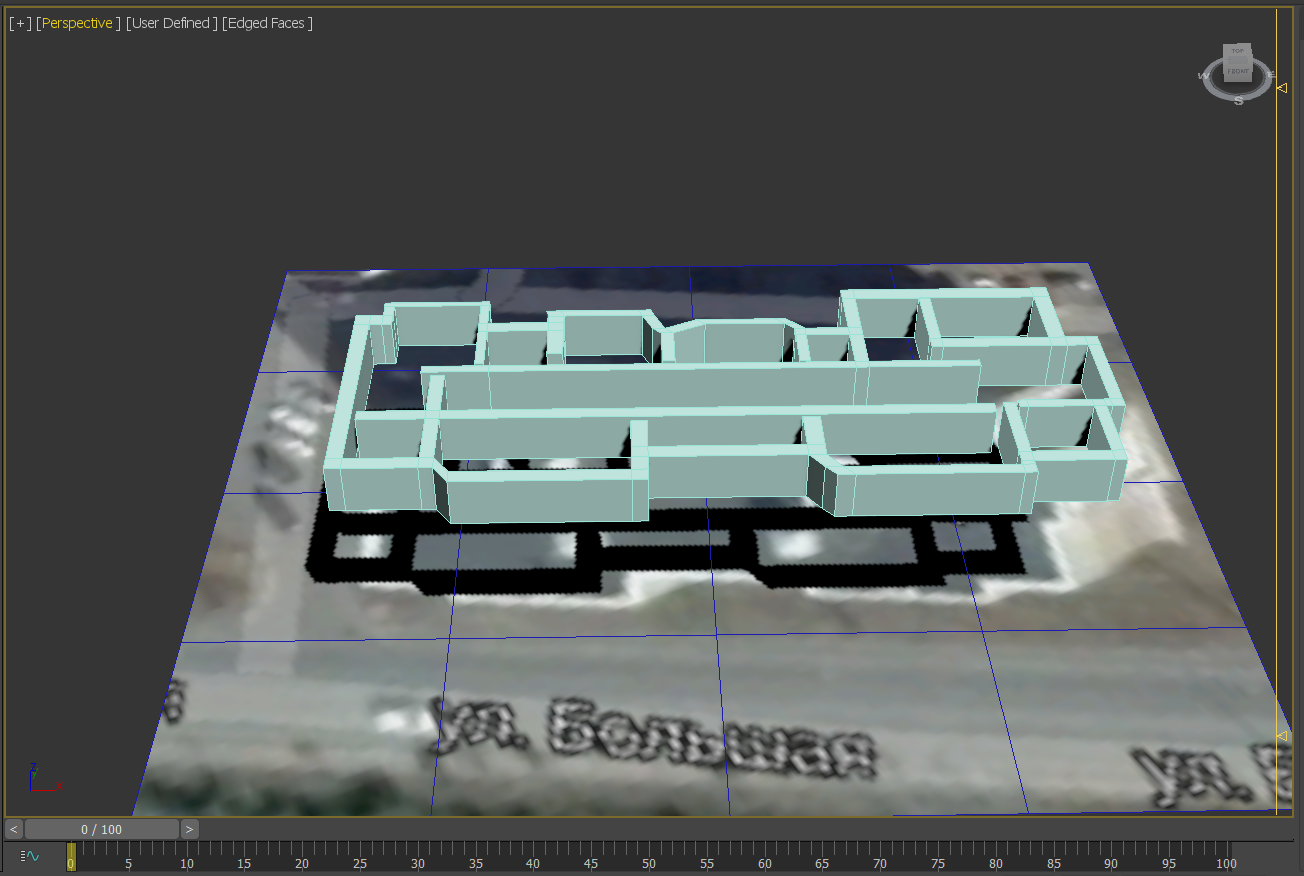


Рис. 2.1.2. Смоделированные фасадные и несущие стены здания в 3D редакторе 3ds Max.

Для создания межэтажного перекрытия используем объект «Линия». После того, как замкнутый контур межэтажного перекрытия будет нарисован, будет создана плоскость в пределах этого контура. На следующем этапе, как и прежде, будет использована функция extrude. В итоге будет создана вторая ключевая единица, необходимая для моделирования целого здания рис. 2.1.3. Для облегчения работы с окончательной моделью здания можно объединить множество разрозненных объектов с помощью встроенного плагина Compound objects. Тип объекта выбрать ProBoolean. В появившемся окне нажать на кнопку Unit, после успешной настройки необходимо приступить к выбору объектов, которые требуется объединить, предварительно кликнув на кнопку Start Picking.

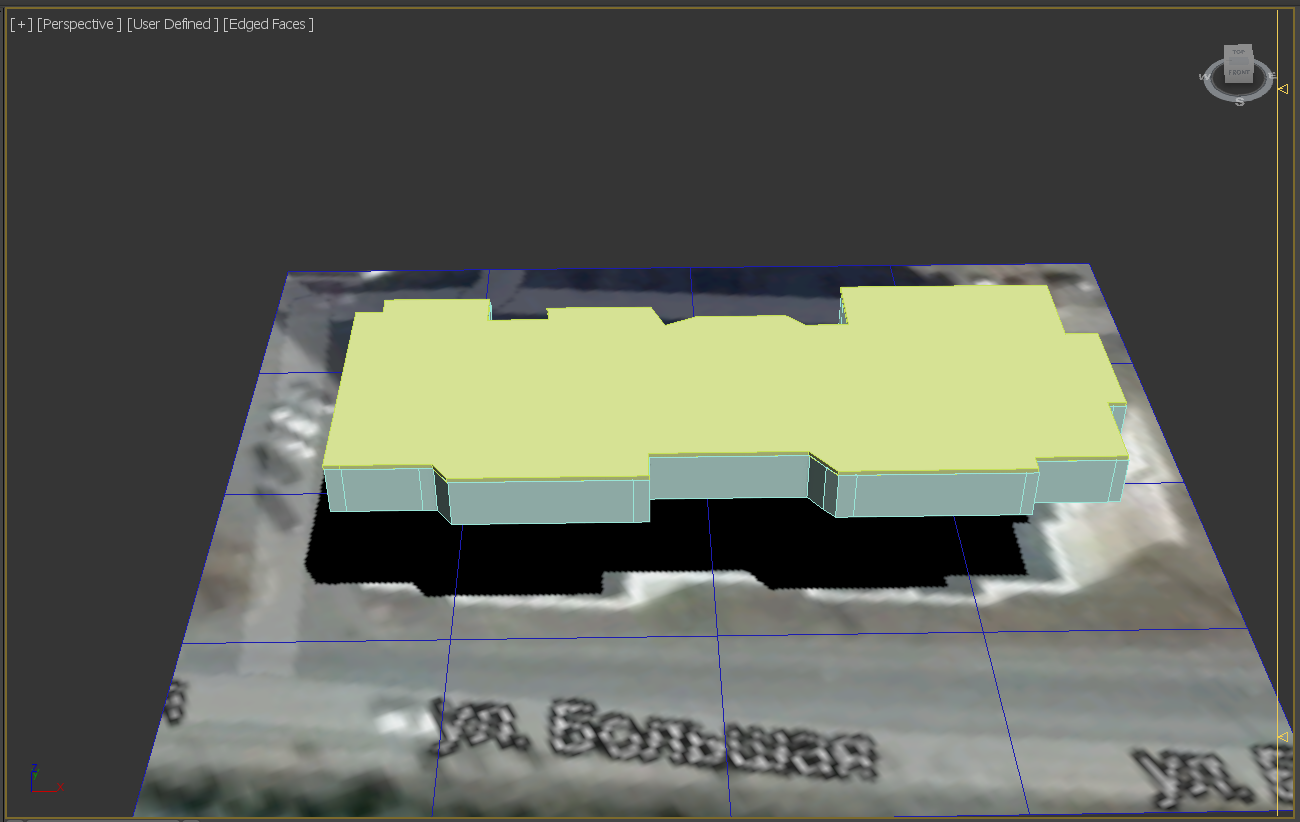


Рис. 2.1.3. Готовая модель одного этажа здания с учетом межэтажного перекрытия в рабочем окне редактора 3ds Max

Таким образом 3ds Max позволяет достаточно быстро моделировать здания. Использования схем или карт, облегчают процесс создания моделей, а встроенный набор плагинов позволяют проводить различные сложные трансформации в несколько шагов. Принимая во внимание тот факт, что .FBX является собственностью компании Autodesk и широкие возможности импорта из данного 3D редактора в сцену Unity, можно заключить, что выбор 3ds Max является рациональным решением.

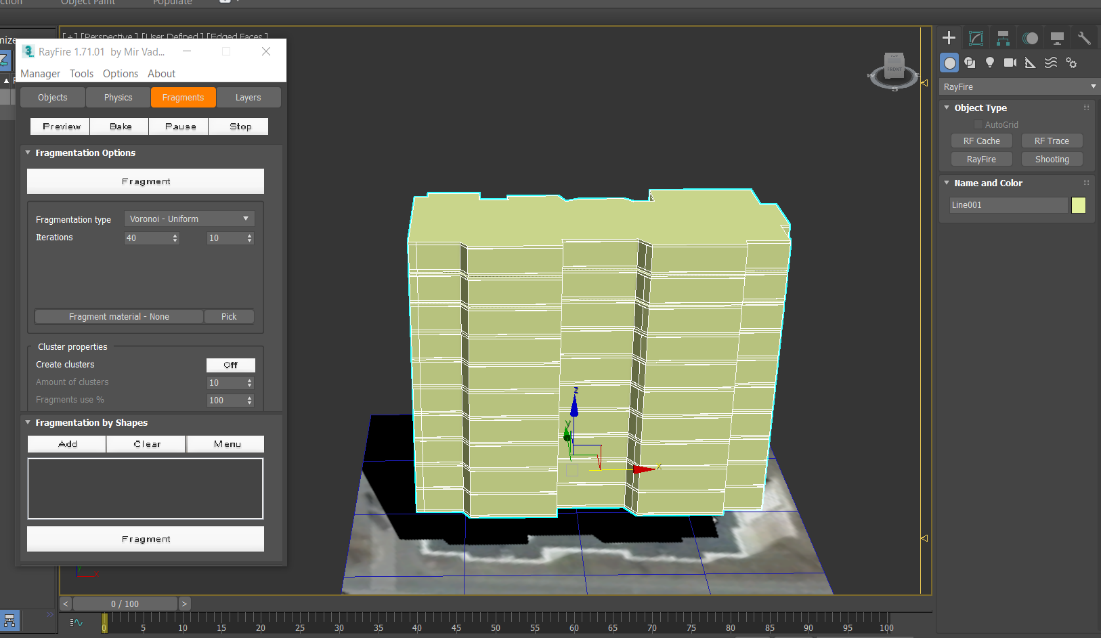
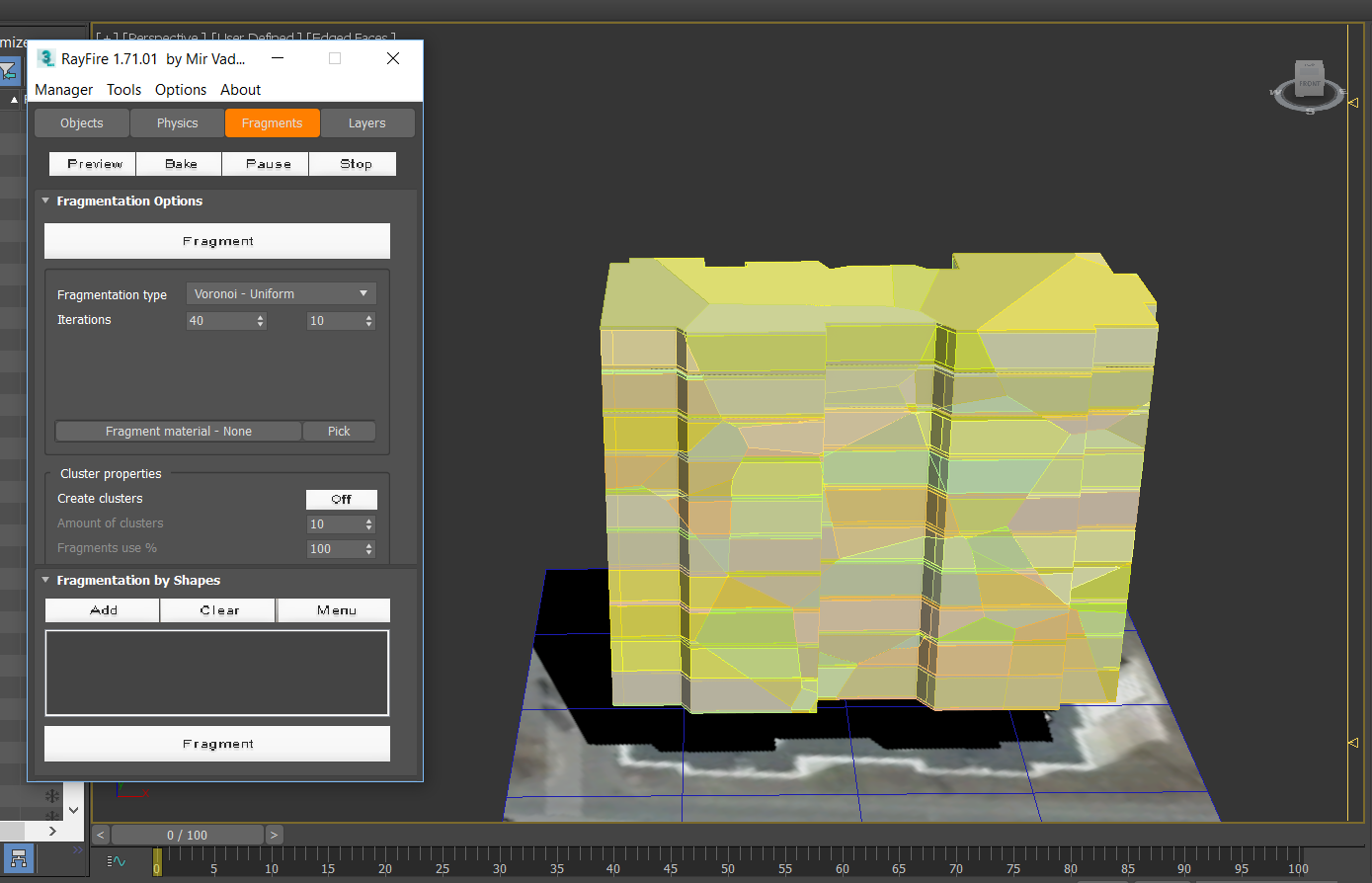
 Разрушение происходит путем замены модели здания на фрагментированную копию. Процесс замены одного объекта на другой во время симуляции будет описан в следующих разделах. Фрагментировать модель в 3ds Max можно без особого труда с помощью плагина RayFire. В окне Object данного плагина нужное внести объекты, подлежащие фрагментации. После чего перейти в окно Fragments. Выбрать количество фрагментов, форму, размеры и т. д. Как только все необходимые настройки будут проведены, нужно запустить процесс фрагментации кликнув на кнопку Fragment. В результате вместо исходного объекта будет отображаться фрагментированный.

Рис. 2.1.4. Пример процесса фрагментации с помощью плагина RayFire. Сверху до фрагментации, снизу после.

**2.2. Разработка математической модели**

Важнейшей компонентой предлагаемого симулятора является скрипт на языке C# Explosion.cs, описывающий физическую составляющую взрыва. Движок Unity располагает удобными решениями для реализации проекта, например, взаимодействие объектов сложной формы. Однако, для написания скрипта для исследовательских целей необходимо разработать математическую модель, основанную на приказе Ростехнадзора от 11.03.2013 N 96 «Об утверждении Федеральных норм и правил в области промышленной безопасности «Общие правила взрывобезопасности для взрывопожароопасных химических, нефтехимических и нефтеперерабатывающих производств» [19]. Для расчетов зон поражения существует две методики:

1. методика оценки зон поражения, основанная на "тротиловом эквиваленте" взрыва опасных веществ;
2. методика, учитывающая тип взрывного превращения (детонация/дефлаграция) при воспламенении ТВС.

Согласно Приложению «Расчет последствий взрыва и критерии взрывоустойчивости зданий» приказа Ростехнадзора [19] расчет зон поражения при взрывах твердых и жидких химически нестабильных соединений, а также для приближенного расчета последствий взрыва ТВС внутри замкнутых объемов (помещений) следует проводить согласно методике, основанной на "тротиловом эквиваленте" [19].

Тротиловый эквивалент является мерой энерговыделения высокоэнергетических событий, выраженная в количестве тринитротолуола (ТНТ), выделяющем при взрыве равное количество энергии [20]. Удельная энергия взрывного разложения тринитротолуола в зависимости от условий проведения взрыва варьирует в диапазоне 980—1100 кал/г. Для сравнения различных видов взрывчатых веществ условно приняты значения 1000 кал/г и 4184 Дж/г. Тротиловый эквивалент рассчитывается по формуле:

где - масса твердых и жидких химически нестабильных соединений, определяемая по их содержанию в технологической системе, блоке, аппарате, кг;

- удельная энергия взрыва твердых и жидких химически нестабильных соединений, кДж/кг;

- удельная энергия взрыва тринитротолуола (далее - ТНТ), кДж/кг.

Зоной разрушения считается площадь с границами, определяемыми радиусами R, центром которой является место расположение взрывчатого вещества. Границы каждой зоны характеризуются значениями избыточных давлений на фронте падающей ударной волны и соответственно безразмерным коэффициентом K. Классификация зон разрушения приводится в таблице 2.1.1.

Методика для определения радиуса зон поражения при взрыве, приведенная в ФНП, использует формулы 1 и 2, полученные путем аппроксимации границ различных степеней разрушения кирпичных зданий в результате бомбардировок Великобритании во время Второй мировой войны [9].

При массе вещества менее 5000 кг.

где K - безразмерный коэффициент, характеризующий воздействие взрыва на объект.

При массе вещества более 5000 кг.

Таблица 2.2.1. Классификация зон разрушения типовых зданий и оборудования

|  |  |  |  |
| --- | --- | --- | --- |
| Класс зоны разрушения | K | , кПа | Возможные последствия, характер повреждений зданий и сооружений |
| 1 | 3,8 | 100 | Полное разрушение зданий с массивными стенами |
| 2 | 5,6 | 70 | Разрушение стен кирпичных зданий толщиной в 1,5 кирпича; перемещение цилиндрических резервуаров; разрушение трубопроводных эстакад |
| 3 | 9,6 | 28 | Разрушение перекрытий промышленных зданий; разрушение промышленных стальных несущих конструкций; деформации трубопроводных эстакад |
| 4 | 28 | 14 | Разрушение перегородок и кровли зданий; повреждение стальных конструкций каркасов, ферм |
| 5 | 56 | 2 | Граница зоны повреждений зданий; частичное повреждение остекления |

Основной недостаток данного подхода в том, что модель «тротилового эквивалента» при большой массе вещества, участвующего во взрыве, показывает зоны поражения с «запасом», так как не учитывает агрегатное состояние вещества, характеристики окружающего пространства и положение точки инициирования взрывоопасного облака. Однако при оценке ликвидации последствий взрывного воздействия такой подход представляется более приемлемым, не требующим сложных вычислительных алгоритмов [9].

Таким образом, создаваемый симулятор будет рассчитывать зоны поражения согласно Федеральным стандартам [19]. Для каждой из указанных зон будет расписан свой сценарий действия. В частности, от этого зависят настройки отдельных фрагментов и условия их разрушения.

**2.3. Практическая реализация симуляции разрушения зданий**

Разработка решения делится на разработку двух основных модулей программы:

1. Физический модуль – в основном это скрипты C#, отвечающие за реалистичную симуляцию взрыва и разрушение зданий с учетом выбранной математической модели.
2. Модуль взаимодействия пользователя с приложением – в первую очередь речь идет о грамотном выстраивании пользовательского интерфейса программы, а также о согласовании и оптимизации работы отдельных скриптов таким образом, чтобы снизить время необходимое на настройку симуляции. Иными словами, необходимо разработать множество отдельных небольших скриптов, отвечающих за передачу значений переменных из одних методов в другие с учетом обстоятельств симуляции. Также необходимо уделить особое внимание механизмам проверки данных на случай, если пользователь будет в ручном режиме менять значения переменных, во избежание ненужных ошибках. Сведения об ошибках необходимо записывать и отображать в журнал событий движка Unity3d.

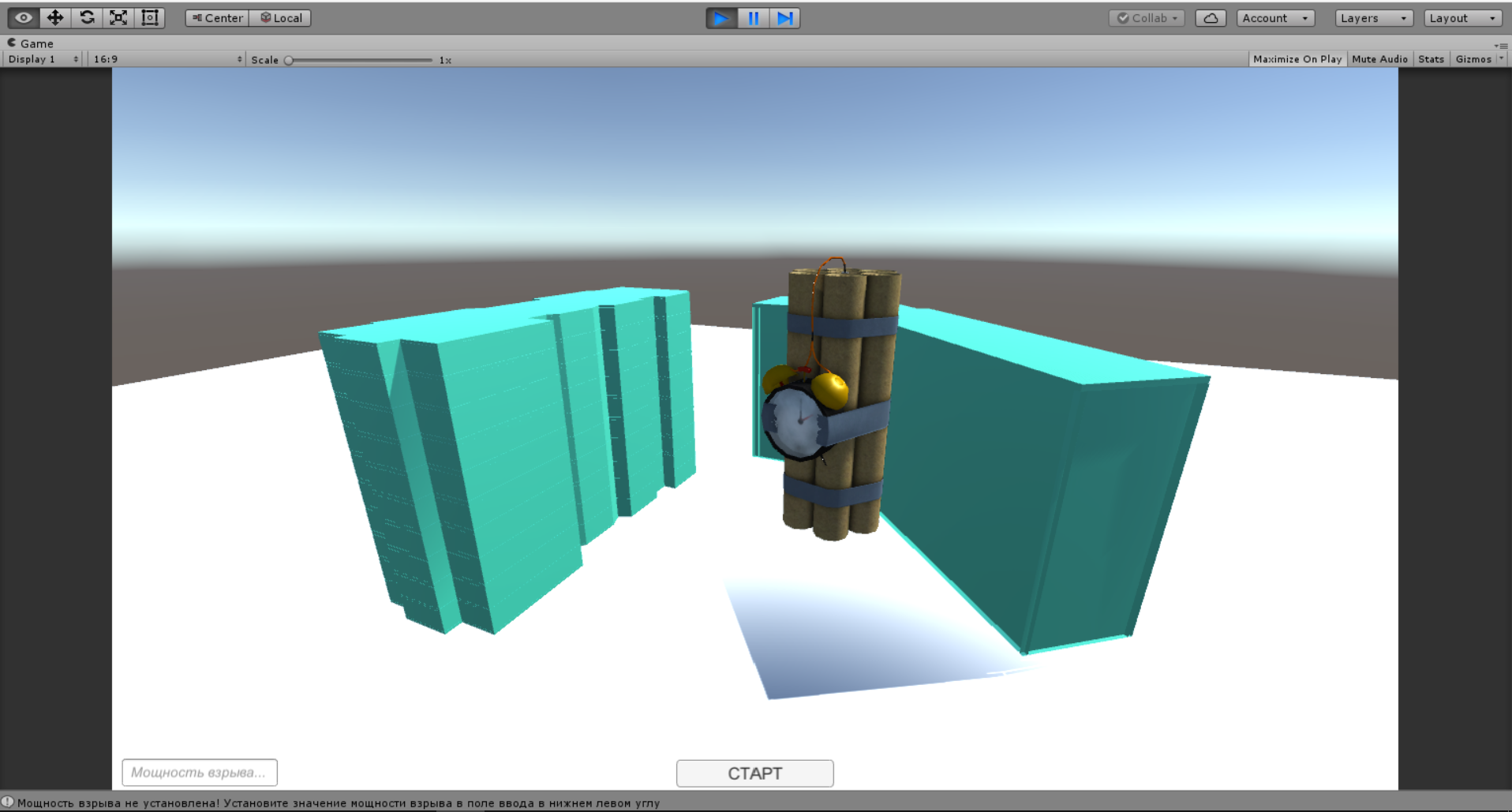


Рис. 2.3.1. Интерфейс запущенного симулятора взрывов.

На рис. 2.3.1. приведен интерфейс разрабатываемого симулятора. Пользовательский интерфейс представляет собой панель Canvas с установленным режимом рендеринга Screen Space Overlay. Для более четкого и сглаженного рендеринга объектов на панели Canvas можно активировать пункт Pixel Perfect. Далее на данную панель были добавлены два объекта: Button, InputField. Стоит отметить, что для корректного отображения объектов панели Canvas вне зависимости от разрешения экрана необходимо использовать специальные якоря Anchors, доступные в инспекторе объектов в графе Rect Transform, иначе при разных режимах экрана и их разрешения положение кнопок будет различным. В данном случае для элемента Button выбран якорь, устанавливающий его всегда в нижней точке экрана по центру, а для InputField якорь установлен в нижнем левом углу экрана. Таким образом, вне зависимости от ситуации данные элементы всегда будут отображаться одинаково. Также в центре обзора камеры, оказывается взрывное устройство, которое пользователю нужно будет перетащить в интересующую его точку. Нужно отметить, что данный элемент имеет длину 0.5 м. Это сделано для того, чтобы пользователь, учитывая масштаб зданий мог легче находить объект на сцене.

Перед запуском симуляции, нажатием кнопки СТАРТ, пользователю необходимо внести значение мощности взрывного устройство в тротиловом эквиваленте. Введенное значение сохраняется в публичной переменной типа float TNTequivalent, объявленной в скрипте InputScript.cs, которая привязана к объекту InputField. В данном скрипте учтены несколько важных вещей. Во-первых, значение сохраняется в переменную только после того, как пользователь закончит редактирование, то есть нажмет на клавишу Enter или кликнет мышкой в любую точку окна программы за исключением самого поля. Это описывает строка кода input.onEndEdit.AddListener(SubmitName). Класс SubmitName, указанный в скобках, является приватным. В нем описана структура данных, которые он должен сохранить, как только событие «конец редактирования» произойдет. Причем изначально данные записываются как строка, поскольку в поле InputField можно указывать любые символы. В связи с последним фактом, необходимо рассматривать два случая:

* пользователь вводит некоторое число, используя только цифры. Тогда значение указанное в строке конвертируется из типа string в тип float и сохраняется в переменной TNTequivalent.
* пользователь помимо цифр, вводит другие символы. Тогда необходимо оповестить пользователя об ошибке в строке журнала событий, которая находится под сценой (рис. 2.3.1.).

В данном случае такое поведение программы обеспечивается использованием конструкции if else, где условием будет float.TryParse(arg, out TNTequivalent). TryParse возвращает значение типа bool. То есть либо true, если конвертация string arg в float TNTequivalent прошла успешно, либо false, если конвертация невозможна. В последнем случае в журнале событий выводится сообщение "ОШИБКА! Введите значение мощности взрыва в тротиловом эквиваленте". Полный код данного скрипта приведен на рис. 2.3.2.

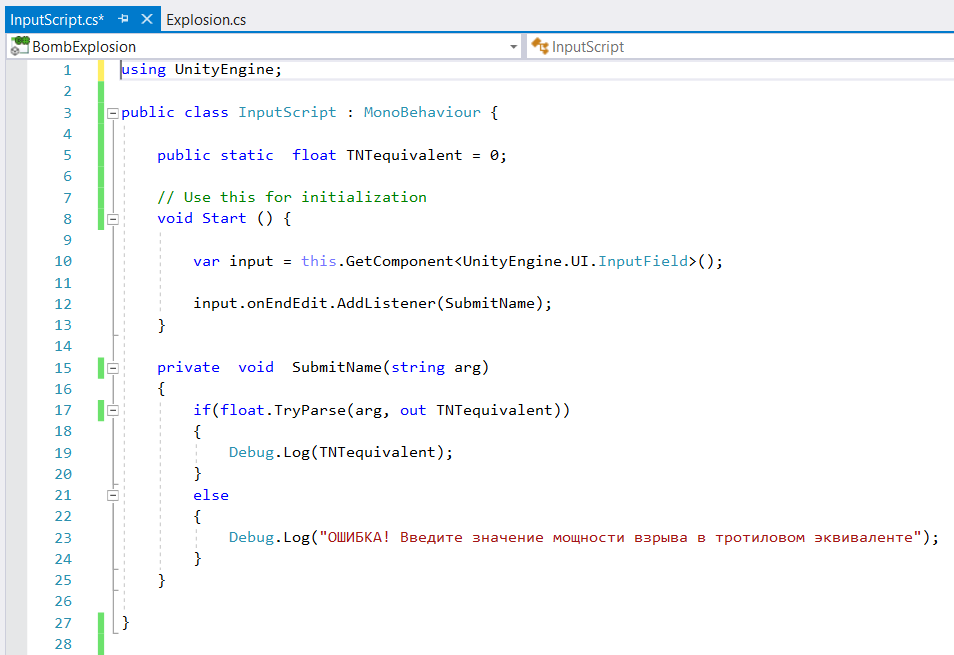


Рис. 2.3.2. Код скрипта InputScript.cs, прикрепленный к объекту InputField.

В случае корректного ввода данных и установления взрывчатого вещества в определенной точке. Пользователь может запустить симуляцию кликнув на кнопку СТАРТ. Для данной кнопки был создан обработчик событий, который вызывает публичный метод StartButtonClicked(). Указанный обработчик создается в инспектора объекта кнопки. В графе Button (Script) необходимо выбрать режим работы кнопки: Editor and Runtime и Runtime Only. В первом случае кнопка будет работать как в режиме редактирования, так и в режиме симуляции, во втором только в режиме симуляции. После необходимо выбрать объект, к которому прикреплен скрипт с вызываемым методом и указать метод в поле function. Таким образом организован вызов метода StartButtonClicked() из класса Explosion. Вызов указанного метода приводит к том, что значение переменной типа bool isStartButtonClicked, изначально равной false, заменяется на true. Значение мощности взрыва, введенного пользователем в InputField, сохраняется в локальную переменную TNTequivalent (рис. 2.3.3.). Стоить отметить, что одноименная переменная сохраненная в скрипте InputScript, является статичной, иначе передача значения из одного класса в другой без ссылки на объект будет невозможной.

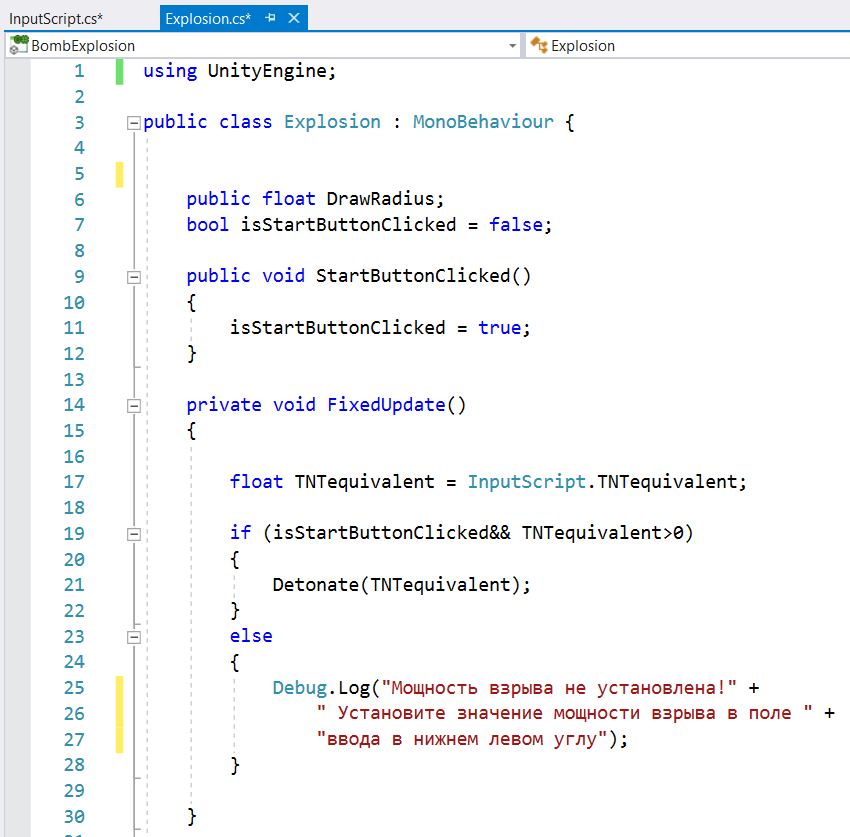


Рис. 2.3.3. Часть кода скрипта Explosion.cs, прикреплённого к объекту Exploder.

Перед запуском расчетов зон, реализации симуляции проводится проверка входных параметров. В частности, проверяется значение переменной TNTequivalent. Если она равна нулю, то выдается сообщение об ошибке «Мощность взрыва не установлена! Установите значение мощности взрыва в поле ввода в нижнем левом углу». Запуск симуляции происходит тогда и только тогда, когда выполняются следующие условия:

* переменная isStartButtonClicked принимает значение true
* значение переменной TNTequivalent больше нуля

На рис. 2.3.3. можно заметить публичную переменную DrawRadius типа float. Это переменная необходима для того, чтобы рисовать сферу заданного радиуса, исходящую из центра объекта, к которому привязан скрипт Explosion.cs. В данном случае сфера прикреплена к взрывному устройства. Таким образом у пользователя появляется дополнительный инструмент с помощью, которого можно мерить габариты объектов на сцене в режиме редактирования. Поскольку линии Gizmos не отображаются в режиме воспроизведения.

«Физическим ядром» симулятора является метод Detonate и класс CanBeDestroyed. В указанном методе, в первую очередь, рассчитываются радиусы зон поражения согласно методике, описанной в параграфе 2.2. Далее создается массив hitColliders, состоящий из объектов типа Collider, в который заносятся все объекты, которые входят в зону поражения. После чего создается цикл, который перебирает каждый объект из массива hitColliders. Внутри этого цикла вычисляется расстояние от объекта Exploder (взрывчатое вещество) до i-го объекта hitColliders (объект, входящий в зону поражения) для того, чтобы определить в какую именно зону поражения он входит, согласно классификации, приведенной в таблице 2.2.1, от этого зависит сценарий разрушения. Например, если объект находится в зоне поражения 1-го класса, то он должен испытывать избыточное давление больше 100кПа, что приведет к полному разрушению здания. В случае попадания в зону поражения 5-го класса здание получит незначительные разрушения.

После того, как зона поражения была предварительно определена, вызывается метод Dead() из класса CanBeDestroyed, который прикреплен к объекту, который должен взаимодействовать с взрывом. Если к зданию не прикрепить данный скрипт, то во время симуляции оно не будет реагировать на взрыв. При прикреплении скрипта к зданию, необходимо в инспекторе объекта указать фрагментированную копию здания. Метод Dead во время запуска сцены удаляет целый объект и в место него создает фрагментированную копию. Далее для каждого фрагмента здания устанавливается компонент MeshCollider со свойством convex равным true. После в зависимости от зоны поражения, типа элементов, входящих в зону поражения устанавливаются Rigidbody. Например, в случае попадания объекта в зону класса 1 и радиуса этой зоны, компонента Rigidbody устанавливается всем фрагментам. В других случаях, для фрагментов, которые не входят в зону поражения и не могу быть разрушены взрывной волной, устанавливается Rigidbody со свойством Kinematic. Для фрагментов, которые должны под действием силы взрыва приобрести начальную скорость и ускорение скрипт в компоненте Rigidbody включает учет силы гравитации (Use Fravity), а collisionDetectionMode выставляется равным ContinuousDynamic. Далее движок Unity каждому фрагменту, к которому прикладывается определенное значение силы, начинает присваивать значение скорости и ускорения в каждом кадре, а также просчитывает взаимодействия фрагмента с другими фрагментами. Как уже было отмечено ранее, выбранная методика проводит расчет с запасом мощности взрыва, поэтому реальный объем повреждений не будет превышать, полученных результатов симуляции.

В скрипте Explosion также объявлены три публичные переменные типа GameObject, в которых хранятся объекты FireBall, Shockwave и BaseSmoke. Это сделано для того, чтобы была возможность управлять анимацией. Только после запуска процесса детонации запускается визуальная часть путем вызов объекта к которому она прикреплена gameObject.GetComponent<ParticleSystem>().Play(). Также в самом скрипте есть приватная переменная Delay. Это время определяет время детонации после запуска симуляции пользователем.

Таким образом, от пользователя требуется импорт моделей зданий в Unity. После чего пользователь сможет провести неограниченное количество симуляций. Для этого необходимо указывать мощность взрыва и положение взрывного устройства. Далее предлагаемое решение произведет симуляцию в режиме реального времени. Также в самом коде есть дополнительные инструменты влияния на процесс для продвинутых пользователей. Более того к симулятору прикладывается скрипт SetCollider, который устанавливает коллайдре указанного типа зданию или множеству зданий на сцене. Демонстрация реализации симуляции приведена в следующем параграфе.

* 1. **Работа приложения**

Для демонстрации возможностей программы был выбран район, указанный на рис. 1.3.1. В частности, дом №108 на ул. Большая и дом №33 на ул. Клары Цеткин. В первую очередь программа была отлажена на прямоугольной стене. На рис. 2.4.1-2.4.3. приведены итоги симуляции взрыва в зависимости от положения. В данном случае стены оказывались в зоне класса 3-4. Из рисунков видно, что в тех случаях, когда в зону поражения, не являющейся зоной 1-го класса, входит верхняя половина стены то, стена разрушается частично. Если же подрывается основание стены, как показано на рис. 2.4.3., то стена разрушается полностью.

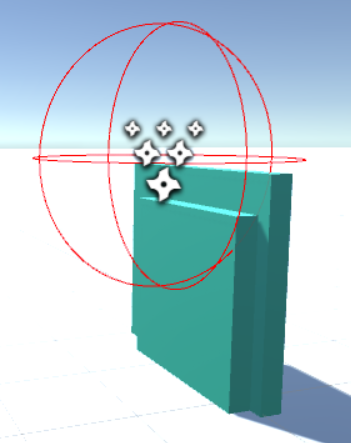
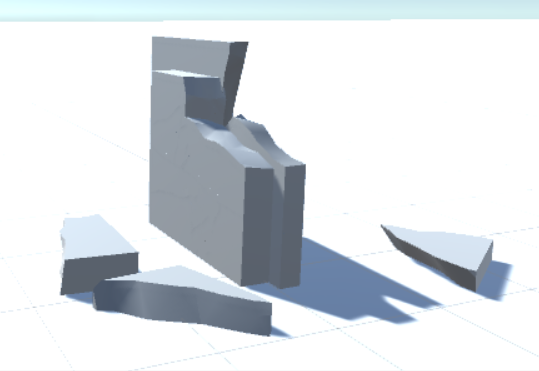
 

Рис. 2.4.1. Сцена подрыва верхней части стены. Слева – сцена до взрыва (красная сфера, описывает зону поражения), справа – сцена после взрыва.

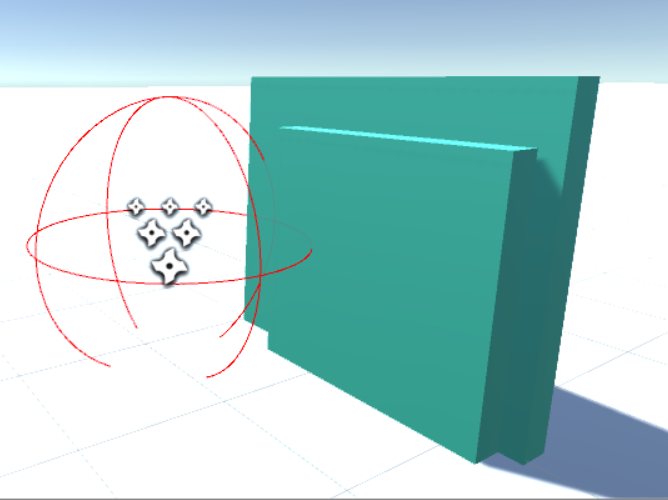
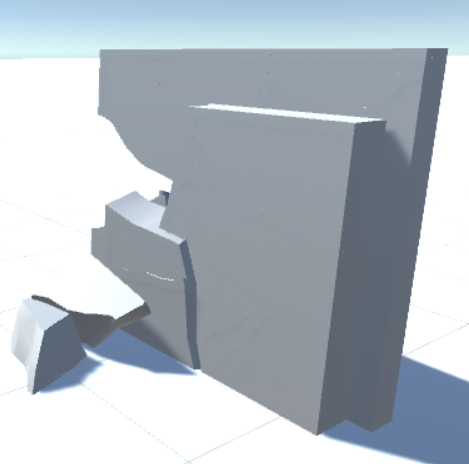
 

Рис. 2.4.2. Сцена подрыва боковой части стены. Слева – сцена до взрыва (красная сфера, описывает зону поражения), справа – сцена после взрыва.

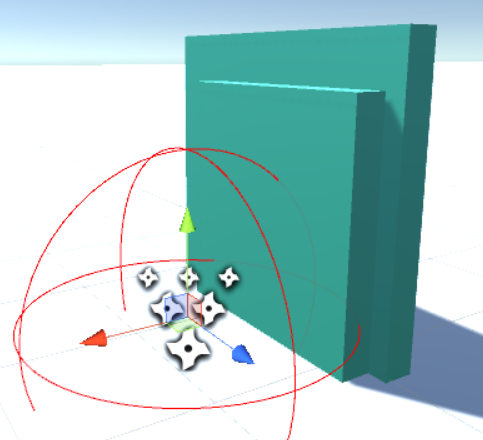
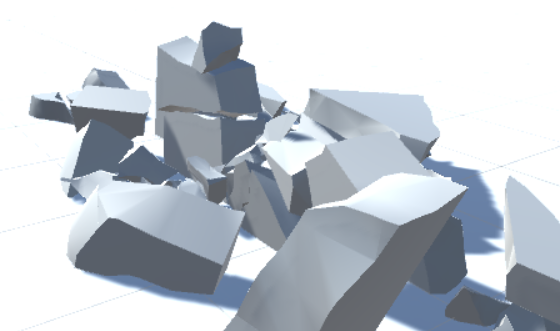
 

Рис. 2.4.3. Сцена подрыва основания стены. Слева – сцена до взрыва (красная сфера, описывает зону поражения), справа – сцена после взрыва.

Таким образом, физика процесса была отработана на этой простой модели. Далее испытания проводились на сцене с моделями зданий. В первую очередь с зданием №108 рис. 2.4.3. Взрывное устройство располагалось ближе к дальней части дома. Расстояние от взрывного устройства до центра масс здания равно 50 метрам, до ближайшей точки около 7 метров. Сила взрыва равна 50 кг в тротиловом эквиваленте. Радиус зоны поражения пятого класса чуть больше 52 метров, третьего класса 9 метров, четвертого класса 26 метров. Урон зданию нанесен колоссальный. При таком взрыве смогла устоять задняя лицевая стена и ближняя часть дома. Перегородки между этажами разрушены практически полностью. Если применить взрывное устройство подобной мощности ближе к центру здания, тогда здания разрушается полностью.

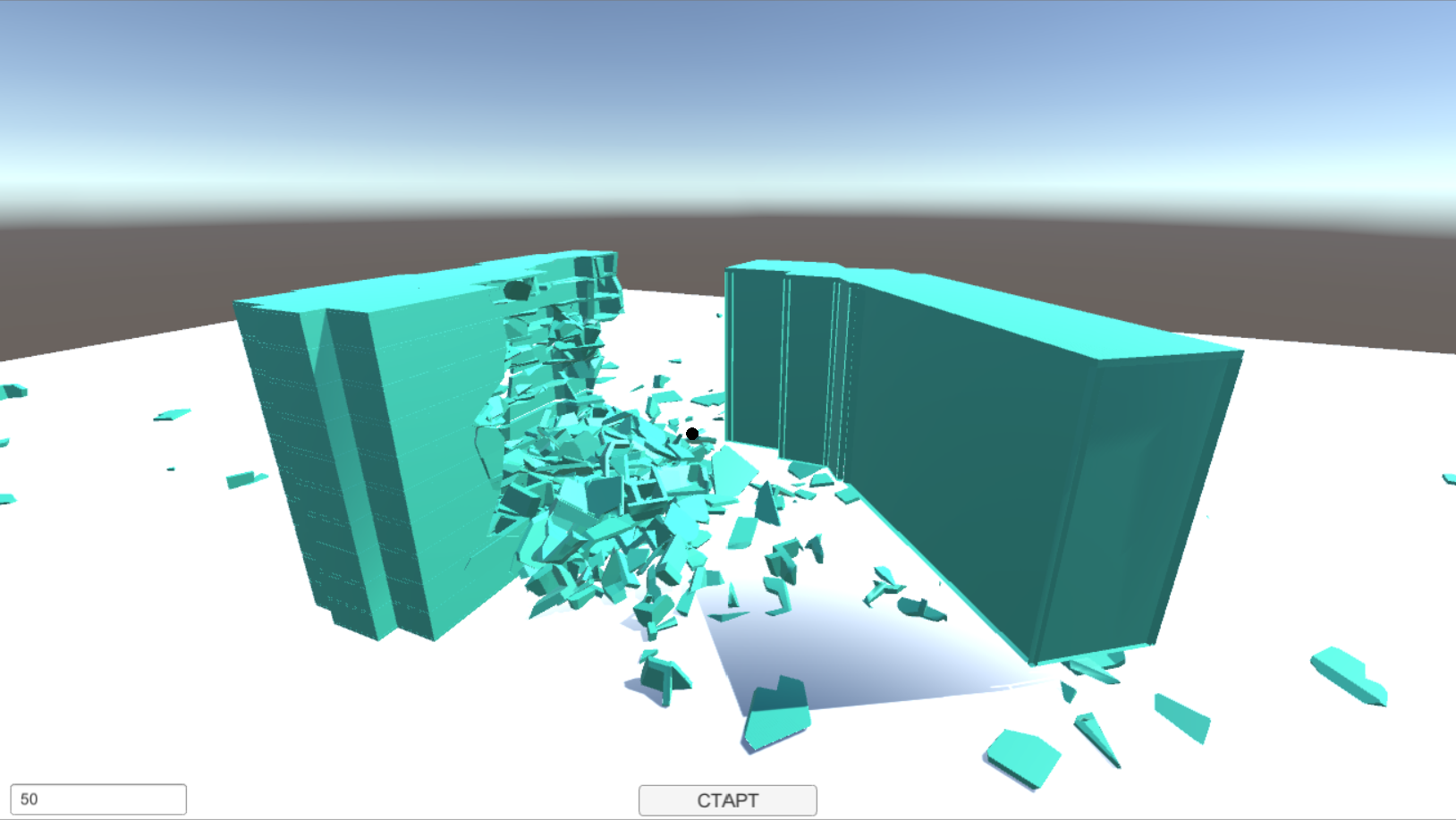


Рис. 2.4.3. Результат симуляции взрыва здания №108. Сила взрыва в 50 кг. в тротиловом эквиваленте. Черная точка соответствуют центру взрыва.

Стоит отметить, что симулятор очень требователен к ресурсам компьютера, поскольку приходится рассчитывать взаимодействия большого количества маленьких фрагментов с одной стороны, с другой стороны программа производит настройку фрагментов в процессе реализации. Поэтому исследовать более двух зданий с высоким уровнем детализации на бюджетном компьютере может быть проблематичным. Оптимизация вычислительных процессов и использование параллельных вычислений может быть одной из следующих задач развития проекта. Также остается сложность с вычислениями расстояний до объекта. В частности, при определении расстояния между объектами стандартными методами возвращается объект типа Vector3. Свойство magnitude данного объекта есть расстояние. У этих методов есть ряд недостатков. Во-первых, для вычисления расстояния требуется вычисление корней. Частое вычисление корней сильно перегружает процессор. Во-вторых, расстояние рассчитывается путем вычитания из координат центра масс одного объекта координат центра масс других объектов. То есть полученное расстояние не совсем корректно использовать для определения класса зоны поражения в который попадает объект. Простейшим решения данного вопроса заключается в создании пяти сфер с радиусами, соответствующими рассчитанным радиусам зон поражения разных классов. Определение классов зон поражения для объекта можно выбирать с помощью метода Physics.OverlapSphere(Vector3 position, float radius), но этот вопрос также может быть в дальнейшем доработан, например, путем вычисления длины луча испускаемого из центра взрыва.

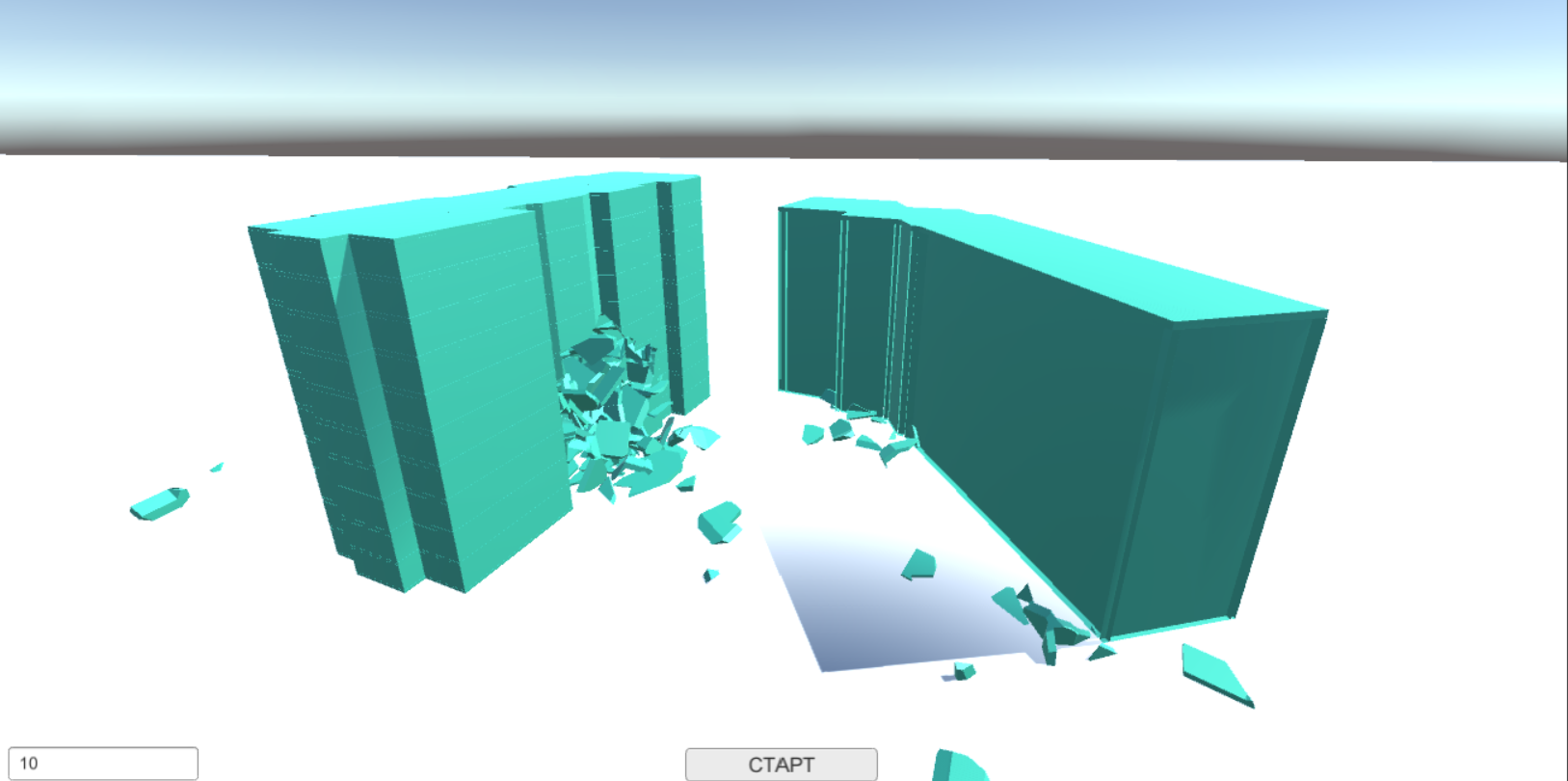


Рис. 2.4.4. Сцена взрыва дома №108. Сила взрыва 10 кг тротилового эквивалента.

На рис. 2.4.4. приведен подрыв дома №108 с использованием взрывчатого устройства. Мощность взрыва - 10 кг тротилового эквивалента. Зоны поражения приведены в табл. 2.4.1. Радиус зон поражения для всех классов растет нелинейно. Симуляции при значениях мощности более 50 кг в тротиловом эквиваленте разрушали здания полностью находясь между двумя зданиями.

На рис. 2.4.5 приведен результат подрыва дома №108. Мощность взрыва 100 кг в тротиловом эквиваленте. В результате подобного взрыва указанный дом разрушен полностью. Дом напротив получил значительные повреждения. Фасадная стена на дальней части дома разрушена, пострадали также бетонные перекрытия между этажами. Также заметны трещины на сохранившейся стороне. Таким образом, можно заключить, что симулятор работает корректно в соответствии с заявленными возможностями.

Табл. 2.4.1. Результаты расчета радиусов зон поражения для разных классов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Мощность взрыва в тротиловом эквиваленте, кг | Радиус зоны поражения, м | | | | |
| Класс 1 | Класс 2 | Класс 3 | Класс 4 | Класс 5 |
| 1 | 0.3 | 0.4 | 0.7 | 1.9 | 3.8 |
| 5 | 0.8 | 1.1 | 1.9 | 5.6 | 11.1 |
| 10 | 1.2 | 1.8 | 3 | 8.8 | 17.7 |
| 50 | 3.5 | 5.2 | 8.9 | 25.8 | 51.7 |
| 100 | 5.6 | 8.2 | 14.1 | 41 | 82 |

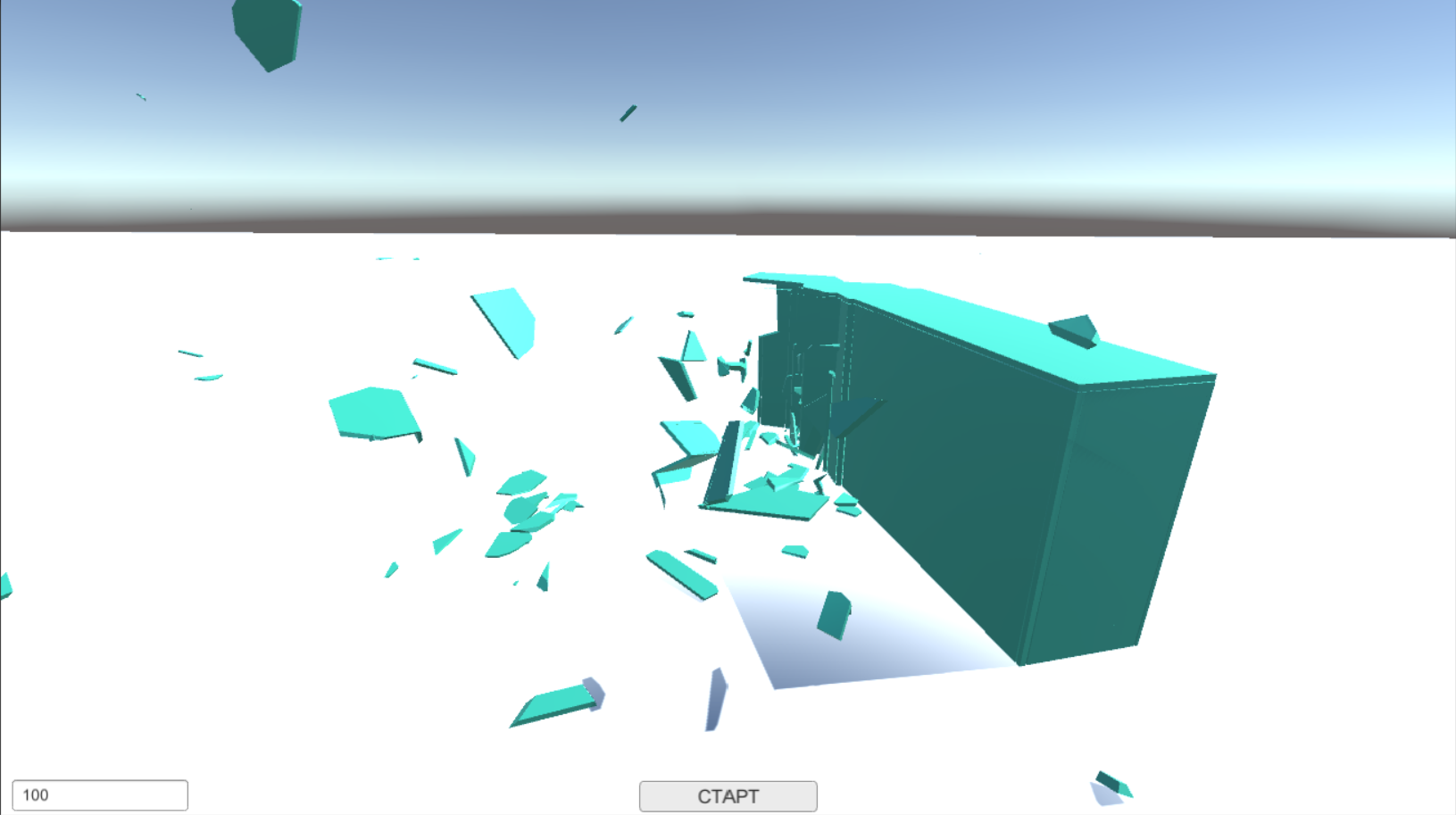


Рис. 2.4.5. Сцена взрыва дома №33. Сила взрыва 100 кг тротилового эквивалента. Взрывчатое устройство было расположено у здания №108, которое в следствие взрыва разрушено полностью.

**Заключение**

В ходе выпускной квалификационной работы был создан симулятор взрывов, основанный на современном движке Unity3d. При реализации данного проекта были учтены федеральные стандарты расчета взрывоустойчивости зданий. Для удобства пользователей разработан простой и удобный интерфейс. Также обеспечен вывод данных в журнал событий для возможности дальнейшего анализа. В соответствии с заданными требованиям симуляция происходит в режиме реального времени. Для тестирования данной системы в 3D редакторе 3ds Max были созданы 3D модели реальных зданий. С использованием созданных моделей были проведены симуляции для разных значений мощности взрыва. Таким образом основная задача выпускной квалификационной работы выполнена успешно. Предлагаемая кроссплатформенная программа может быть использована обширной аудиторией, хотя основными целевыми группами являются инженеры, архитекторы, сотрудники строительных компаний, сотрудники МЧС, аспиранты и студенты.

**Список литературы**

1. Global terrorism database [Электронный ресурс] // https://www.start.umd.edu/gtd/
2. Nukemap [Электронный ресурс] // https://nuclearsecrecy.com/nukemap/
3. Unity 3D. Базовый курс [Электронный ресурс] // <http://www.realtime.ru/courses/unity-3d/>
4. Лапин А.И. Руководство по Unity 3D. - Интернет-издательство, 2013. — 77 с.
5. Miles Jere. Unity 3D and PlayMaker Essentials: Game Development from Concept to Publishing. – CRC Press, 2016. – 507 p.
6. Creighton R.H. Unity 3D Game Development by Example. Beginner's Guide. - Packt Publishing, 2010. - 384 с.
7. Unity [Электронный ресурс] // <https://unity3d.com/ru>
8. OpenStreetMap [Электронный ресурс] // https://www.mapbox.com/blog/mapping-3d-buildings/
9. Деркачев Н. В., Деркачев В. И., Быльев Ю. В., Медведева А. Н., Афанасьев Р. В. Расчет зон разрушений зданий и сооружений при взрывах на опасных производственных объектах / Проблемы современной науки и образования, 2015 - № 10 (40). – С. 42-46
10. Рыженко А.А. Механизм моделирования сколов и осколков разрушенных объектов сложной формы / Труды Кольского научного центра РАН, 2014. - №5 (24). – С. 204-214
11. Руководство Unity [Электронный ресурс] // <https://docs.unity3d.com/ru/current/Manual/UnityManual.html>
12. LS-DYNA [Электронный ресурс] // <http://www.dynaomd.ru/explosion.htm>
13. NUKEMAP3D [Электронный ресурс] // <http://nuclearsecrecy.com/nukemap3d/>
14. Announcing deprecation of the Google Earth API [Электронный ресурс] // <https://maps-apis.googleblog.com/2014/12/announcing-deprecation-of-google-earth.html>
15. LS-DYNA [Электронный ресурс] // <http://www.lstc.com/products/ls-dyna>
16. LS-DYNA [Электронный ресурс] // https://ru.wikipedia.org/wiki/LS-DYNA
17. Valcartier C.R. Numerical Study of Soil Modelling Approaches using LS-DYNA: Part 2 / Defence R&D Canada - Valcartier Contract Report DRDC, 2009.
18. FBX [Электронный ресурс] // https://ru.wikipedia.org/wiki/FBX
19. Приказ Ростехнадзора от 11.03.2013 N 96 «Об утверждении Федеральных норм и правил в области промышленной безопасности «Общие правила взрывобезопасности для взрывопожароопасных химических, нефтехимических и нефтеперерабатывающих производств» [Электронный ресурс] // <http://www.consultant.ru/cons/cgi/online.cgi?req=doc&base=LAW&n=194477&rnd=261745.556910475&from=145465-37#0>
20. Тротиловый эквивалент [Электронный ресурс] // <https://ru.wikipedia.org/w/index.php?title=Тротиловый_эквивалент&oldid=83070840>