

---

предприятие (организация)

УДК \_\_\_\_\_

№ госрегистрации \_\_\_\_\_

Инв. № \_\_\_\_\_

УТВЕРЖДАЮ

---

головной исполнитель НИР

---

« \_\_\_\_\_ » \_\_\_\_\_ 2019 г.

ОТЧЁТ  
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

---

наименование и шифр НИР

---

(промежуточный)  
шифр — \_\_\_\_\_

## СОДЕРЖАНИЕ

Введение.....	4
<b>1 Аналитический раздел.....</b>	<b>6</b>
1.1 Анализ предметной области .....	6
1.2 Обзор и анализ существующих решений, обоснование необходимости разработки .....	8
1.3 Выбор, обоснование метода моделирования и алгорит- ма .....	12
2 Конструкторский раздел.....	13
3 Технологический раздел.....	14
4 Экспериментальный раздел.....	15
Заключение.....	16
Список использованных источников.....	17
Приложение А Картинки.....	18

## РЕФЕРАТ

Отчет содержит 18 стр., 2 рис., 5 источн., 1 прил.

## ВВЕДЕНИЕ

На протяжении десятилетий взрывы были самыми динамичными и визуально привлекательными спецэффектами в кино и видеоиграх. Они стали настолько заметными в боевиках и приключенческих фильмах, что кажется необычным, когда его нет в фильме. Каким был бы фильм "Звездные войны" без финального взрыва Звезды Смерти?

Традиционно взрывные эффекты создаются перед камерой, а не в компьютере. Либо строится уменьшенная модель и взрывается перед высокоскоростными камерами, либо используются настоящие взрывчатые вещества.

Правда в кинематографе, важна эффектность, поэтому все заранее планируется в мельчайших подробностях.

Если нужно взорвать настоящее здание, пусть даже специально для этого построенное, его предварительно готовят: подпиливают рамы так, чтобы нужная часть их осталась на месте, а нужная – разлетелась на части. Также это делается для того, чтобы здание развалилось при минимально необходимом заряде. Все части, что должны развалиться, делаются из легких материалов с тем, чтобы если какие-то куски случайно упадут на голову кинематографистам, ущерб был бы минимален.

Если при этом в кадре должны быть персонажи, с людьми скрупулезно отрабатываются все их передвижения, фиксируется время, необходимое для того, чтобы они достигли безопасного расстояния до того, как произойдет главный взрыв. [1]

Существует множество веских причин для использования компьютеров для создания взрывных эффектов вместо более традиционных практических методов. Основной мотивацией, конечно, забота о безопасности актеров. Когда взрыв происходит полностью внутри компьютера, нет никаких шансов на то, что кто-то случайно попадет в зону взрыва. Также компьютерные взрывы дешевле и быстрее, чем точное масштабирование и размещение детонаторов, а также огнестойкость существующих

конструкций или создание специальных миниатюр. Когда режиссеры снимают практический взрыв, они настраиваются на несколько дней, чтобы получить несколько ракурсов на один единственный взрыв. С помощью компьютерных эффектов режиссеры могут посмотреть на промежуточный результат и попросить что-то изменить, чтобы более точно отразить свое творческое видение.

Целью проекта является создание максимально приближенной модели взрыва большого числа частиц, при столкновении с телом, имеющим больший размер с использованием графического редактора систем частиц. Моделирование основано на физическом явлении взрыва взрыва и возникающих побочных эффектов в заданном пространстве за заданное время и взаимодействующих с окружающей средой.

Для достижения поставленной цели необходимо решить следующие задачи:

- а) Определить понятие системы частиц.
- б) Создать движок для работы с частицами.
- в) Изучить физическое явление - взрыв.
- г) Смоделировать взрыва большого числа частиц, при столкновении с телом

## 1 Аналитический раздел

Целью работы является создание максимально приближенной модели взрыва большого числа частиц. Объекты в сцене представлены в виде твердых частиц, которые приводятся в движение различными силами.

### 1.1 Анализ предметной области

**Система частиц** – широко используемый в компьютерной графике метод представления объектов, не имеющих четки геометрических границ. Облака, туманности, дым, взрыв, снег – все эти объекты моделируются с помощью систем частиц. Системы частиц создают объем частиц с индивидуальными свойствами, а не поверхность с текстурой. При таком подходе возможно изменять внешний вид каждой отдельной частицы с течением времени и, таким образом, создавать визуализации, которые реалистично отображают огонь, дым, воду, взрывы и с некоторыми изменениями даже волосы и мех. Они также используются для представления неестественных явлений, таких как магические эффекты. [2]

Уже в начале 1980-х годов Уильям Т. Ривз использовал системы частиц для моделирования огня, поглощающего планету, в фильме «Звездный путь II: Гнев Хана». Ривз считается изобретателем систем частиц, и в 1983 году он опубликовал статью, объясняющую, как это было сделано. Ривз продолжил развивать эту идею и с тех пор вносил большой вклад в область систем частиц. Разработка продолжалась, и сегодня большинство игровых движков содержат систему частиц. [3]

Эта техника имеет много полезных приложений, будь то большие взрывы, пыль, пожары или большие толпы людей, спецэффекты необходимы во многих фильмах, анимациях, съемках или компьютерных играх. Фильмы, показывающие массивные взрывы и разрушенные здания, могут имитировать эти эффекты, а не создавать их по-настоящему. В

этой работе мы сосредоточимся на визуализации взрыва большого числа частиц, при столкновении с телом, имеющим больший размер.

Взрыв в реальности состоит из мелких частиц, которые движутся независимо друг от друга, однако, поскольку частицы будут взаимодействовать в основном одинаковыми силами, они будут вести себя одинаково.

Визуализация взрыва частиц может быть описана по-разному в зависимости от источника силы, порождающий взрыв. В данной работе, источником является шарообразное тело, врезающееся в систему более мелких частиц.

На сцене располагается неподвижная, в начальный момент, группа частиц, расположенных вместе, шарообразное подвижное тело, точечный источник света.

В начальный момент времени, есть возможность скорректировать размеры тела, поменять расположение камеры.

После нажатия на кнопку, произойдёт симуляция.

На рисунке 1.1 изображена **функциональная модель**, отображающая структуру и функции системы.

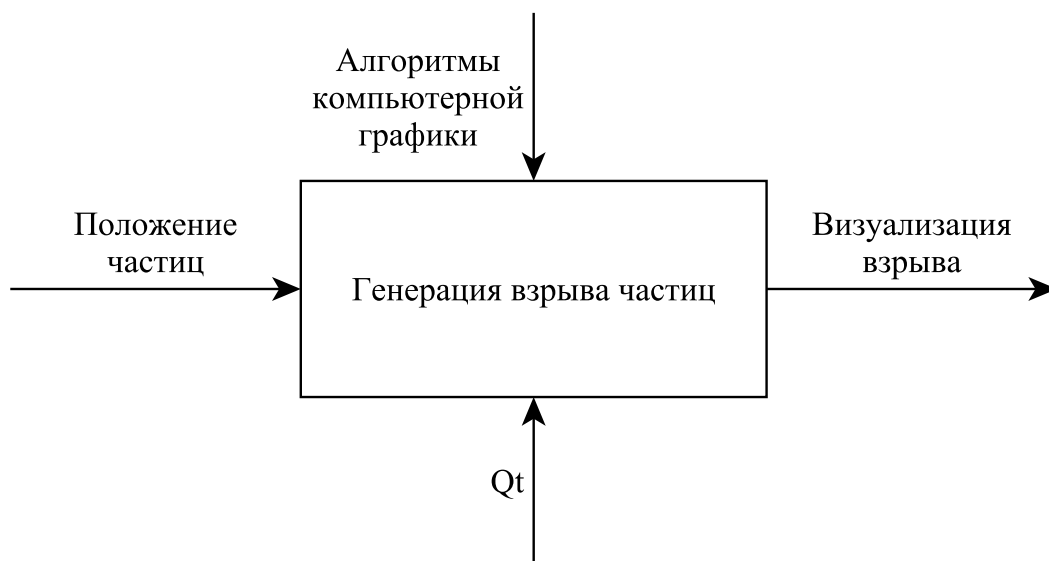


Рисунок 1.1 — Функциональная модель процесса визуализации взрыва частиц

На вход принимаются частицы с различными параметрами (положение, масса, размер) и положение шарообразного врезающегося тела, с идентичными параметрами. На выходе получаем визуализацию взрыва мелких частиц при столкновении с телом.

В рамках данного исследования существует возможность, менять параметры частиц.

Ставится задача получения визуализации взрыва в режиме реального времени.

## 1.2 Обзор и анализ существующих решений, обоснование необходимости разработки

**В настоящее время при 3D моделировании объекты часто создают в основном только двумя способами. [4]**

а) Либо с помощью плоских полигонов — тем самым будет создана полая модель без внутреннего наполнения.



б) Либо с помощью частиц, которые полностью заполняют внутренности модели, где каждая частица представляется как материальная точка с дополнительными атрибутами, такими как скорость, цвет, ориентация в пространстве, угловая скорость и т. п.

Ввиду того, что полигональные модели пусты по своей природе, очень трудно моделировать их поведение в 3D мире, как, например, всплески воды. Если же воду моделировать через систему частиц, то всё становится гораздо проще, так как вся вода от поверхности океана и до дна состоит из атомов, которые можно рассматривать, как набор отдельных частиц.

**Далее рассматриваются алгоритмы удаления невидимых поверхностей. [5]**

- а) Алгоритм, использующий z-буфер
- б) Алгоритм разбиения области Варнока
- в) Метод трассировки лучей

#### **Алгоритм удаления поверхностей с Z-буфером.**

Алгоритм предложен Эдом Кэтмулом и представляет собой обобщение буфера кадра. Обычный буфер кадра хранит коды цвета для каждого пиксела в пространстве изображения. Идея алгоритма состоит в том, чтобы для каждого пиксела дополнительно хранить еще и координату Z или глубину. При занесении очередного пиксела в буфер кадра значение его Z-координаты сравнивается с Z-координатой пиксела, который уже находится в буфере. Если Z-координата нового пиксела больше, чем координата старого, т.е. он ближе к наблюдателю, то атрибуты нового пиксела и его Z-координата заносятся в буфер, если нет, то ни чего не делается. Таким образом, после выполнения этих операций в каждой точке экрана будет находиться пиксель, соответствующий грани, находящейся ближе всего к картинной плоскости в данной точке.

Этот алгоритм наиболее простой из всех алгоритмов удаления невидимых поверхностей, но требует большого объема памяти. Время

работы алгоритма не зависит от сложности сцены. Многоугольники, составляющие сцену, могут обрабатываться в произвольном порядке.

Основной недостаток алгоритма с Z-буфером - дополнительные затраты памяти. Другие недостатки алгоритма с Z-буфером заключаются в том, что так как пиксели в буфер заносятся в произвольном порядке, то возникают трудности с реализацией эффектов прозрачности или просвечивания и устранения лестничного эффекта.

### **Алгоритм разбиения области Варнока.**

Алгоритм работает в пространстве изображения и анализирует область на экране дисплея (окно) на наличие в них видимых элементов. Если в окне нет изображения, то оно просто закрашивается фоном. Если же в окне имеется элемент, то проверяется достаточно ли он прост для визуализации. Если объект сложный, то окно разбивается на более мелкие, для каждого из которых выполняется тест на отсутствие и/или простоту изображения. Рекурсивный процесс разбиения может продолжаться до тех пор пока не будет достигнут предел разрешения экрана (1 пиксель).

Можно выделить 4 случая взаимного расположения окна и многоугольника (рис. 1.2):

- а) многоугольник целиком вне окна,
- б) многоугольник целиком внутри окна,
- в) многоугольник пересекает окно,
- г) многоугольник охватывает окно.

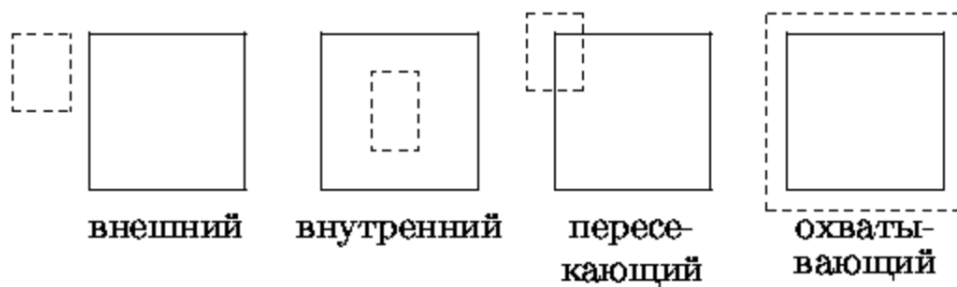


Рисунок 1.2 — Соотношения между окном экрана (сплошная рамка) и многоугольником (штриховая рамка)

В четырех случаях можно сразу принять решение о правилах закраски области экрана:

а) все многоугольники сцены - внешние по отношению к окну. В этом случае окно закрашивается фоном;

б) имеется всего один внутренний или пересекающий многоугольник. В этом случае все окно закрашивается фоном и затем часть окна, соответствующая внутреннему или пересекающему окну закрашивается цветом многоугольника;

в) имеется единственный охватывающий многоугольник. В этом случае окно закрашивается его цветом.

г) имеется несколько различных многоугольников и хотя бы один из них охватывающий. Если при этом охватывающий многоугольник расположен ближе остальных к наблюдателю, то окно закрашивается его цветом.

В любых других случаях процесс разбиения окна продолжается.

Алгоритм работает с многоугольниками.

### **Алгоритм трассировки лучей.**

Суть метода заключается в том, что для каждой точки на проектируемой плоскости строится луч, начало которого совпадает с положением наблюдателя, а угол луча по отношению к направлению наблюдения определяется характеристиками наблюдателя (максимальный угол зрения относительно горизонтальной и вертикальной плоскостей) и точкой плоскости проецирования (картинная плоскость), для которой строится луч.

После построения луча определяется его пересечение со всеми гранями, из которых состоит сцена и выбирается та грань, расстояние от которой до наблюдателя минимально.

Далее в данной точке картинной плоскости рисуется пиксель цветом, определяемым в точке пересечения луча и выбранной грани.

Этот метод является довольно простым и позволяет совместить определение видимости с расчётом цвета соответствующего пикселя. Ещё одним преимуществом метода является то, что сцена может состоять не из треугольников, а быть задана набором геометрических примитивов (например, поверхностями второго порядка). В этом случае пересечение луча с геометрическим примитивом вычисляется аналитически (точное решение задачи построения геометрического примитива), в отличие от разбиения произвольного геометрического примитива (например, сферы) на треугольники и решения более простой задачи пересечения луча с треугольником.

Могут генерироваться новые лучи внутри сцены для корректного отображения отражений, преломлений, затенений – алгоритм обратной трассировки лучей более эффективен.

Может быть модифицирован для отображения общего освещения сцены.

Тем не менее для реализации метода требуется довольно большое количество вычислений и следовательно большие временные затраты.

**Актуальность темы** исследования обусловлена тем, что за последние несколько лет технология виртуальной реальности совершила огромный скачок в развитии и расширении сфер применения. Если раньше эта технология в основном применялась в военной промышленности и компьютерных играх, то сейчас виртуальная реальность проникает практически во все сферы деятельности человека: медицину, образование, архитектуру, рекламу и прочее. Эта технология имеет огромный потенциал и поэтому она так активно развивается.

### **1.3 Выбор, обоснование метода моделирования и алгоритма**

На основании предложенных методов моделирования, в данной работе будет использоваться система частиц (а именно сфер).

Для удаления невидимых поверхностей будет реализован алгоритм обратной трассировки лучей, несмотря на вычислительную сложность, алгоритм предоставляет возможность работать непосредственно со сферическими поверхностями и отображать освещение.

#### 1.4 Вывод

В данной работе ставится задача визуализации взрыва системы частиц при столкновении с шарообразным телом, используя алгоритм обратной трассировки, как наиболее оптимальный для поставленной задачи.

На сцене располагается неподвижная, в начальный момент, группа частиц, расположенных вместе, шарообразное подвижное тело, точечный источник света.

В начальный момент времени, есть возможность скорректировать размеры тела, поменять расположение камеры.

После нажатия на кнопку, произойдёт симуляция.

## 2 Конструкторский раздел

### 3 Технологический раздел

## 4 Экспериментальный раздел



## ЗАКЛЮЧЕНИЕ

В результате проделанной работы

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Взрывы в кино: что за кадром? — Режим доступа: <http://www.mir3d.ru/vfx/950/> (дата обращения: 10.10.2019).
2. Простая система частиц? — Режим доступа: <http://grafika.me/node/496> (дата обращения: 16.11.2019).
3. Reeves William T. Particle Systems – A Technique for Modeling a Class of Fuzzy Objects. — Computer Graphics, том 17, №3, с. 359-376, 1983.
4. И. Е. Жигалов И. А. Новиков. Программирование трехмерной компьютерной графики. — Изд-во ВлГУ, Владимир, 2016.
5. Удаление скрытых линий и поверхностей. — Режим доступа: <http://algolist.ru/graphics/delinvis.php> (дата обращения: 16.11.2019).

ПРИЛОЖЕНИЕ А

КАРТИНКИ