Разработка системы визуализации физических моделей в реальном времени

Перед авторами была поставлена задача создания системы визуализации физических моделей в реальном времени. Актуальность нашего проекта заключается в том, что он может быть использован в различных сферах деятельности: кинематограф, компьютерные игры, компьютерное моделирование может быть использовано для наглядной демонстрации законов физики на уроках.

Наша система должна в реальном времени отслеживать столкновения твердых тел и визуализировать их движение под действием сил и импульсов в результате внешнего воздействия.

Для визуализации трехмерных сцен в реальном времени была реализована система вывода с использованием графической библиотеки DirectX 12, поддерживающая язык программирования шейдеров hlsl. Данная система позволяет модулям программы напрямую взаимодействовать с видеокартой. Для реалистичного вывода объектов нами было реализовано отложенное освещение и каскадные тени. Кроме того в проекте используется тесселяционный шейдер.

Основой нашего проекта является математически корректное моделирование движений и столкновений тел в пространстве. Моделирование движений осуществляется при помощи формул и уравнений механики.

Одна из проблем возникнувших в ходе реализации программы заключается в том, что, в отличие от реального мира, где время непрерывно, время в компьютерных симуляциях таковым быть не может. Для ее решения была создана константа «квант времени». Она определяет минимальный промежуток времени, за который с телами в системе может что-либо произойти. Таким образом, вне зависимости от технических характеристик оборудования, процессы протекают одинаково.

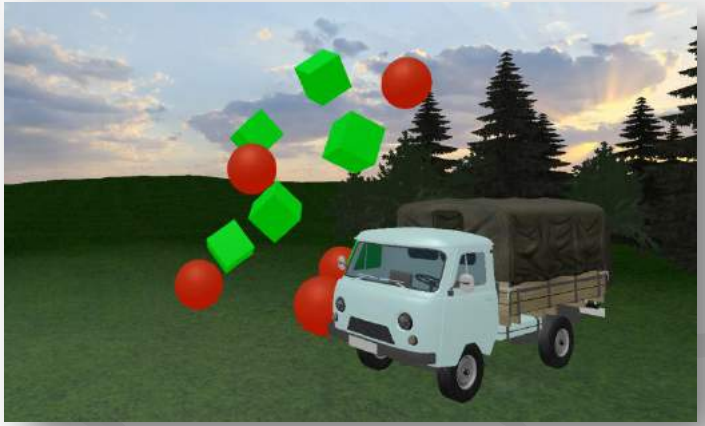
В процессе обработки физической модели необходимо в каждый момент времени знать, какие тела сталкиваются друг с другом, а также ключевые точки пространства, в которых происходит соприкосновение. Для осуществления данной задачи было реализовано несколько алгоритмов: построение Kd-Tree, GJK (Gilbert–Johnson–Keerthi algorithm) и EPA (Expanding polytype algorithm) для нахождение вектора проникновения , нахождение точек соприкосновения.

В нашем проекте присутствуют параллельные вычисления на GPU, которые осуществлены при помощи CUDA API.

В результате работы, нам удалось создать систему построения реалистичных физических процессов в реальном времени. Во время работы нами были изучены множество алгоритмов и вспомогательных исследований, связанных визуализацией, физикой и системой пересечения. Кроме того, нами был изучен метод распараллеливания вычислений на GPU с помощью CUDA API. В будущем авторы планируют развивать возможности физической составляющей проекта и оптимизировать расчет столкновений, а также добавить различные визуальные эффекты.

Разработка системы визуализации физических моделей в реальном времени

- Лазаренко Александр Николаевич 9-1
- Власов Дмитрий Вадимович 10-1
- Серков Александр Максимович 10-1
- Черепков Петр Кириллович 10-1
- Марков Максим Александрович 10-2
- Юркин Александр Сергеевич 10-2
- Иванов Святослав Игоревич 10-3



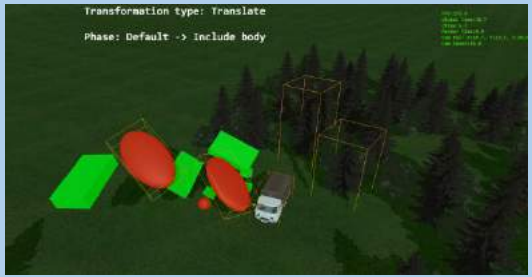
Научный руководитель: Галинский В.А.,
преподаватель информатики и программирования
СПб губернаторского физико-математического лицея № 30

Цель проекта

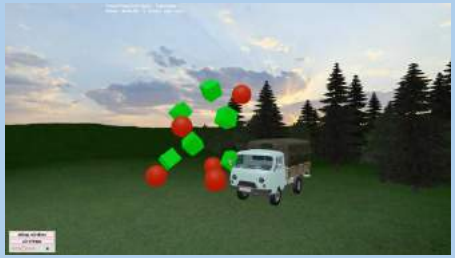
Задача: создать систему, которая в реальном времени отслеживает столкновения твердых тел и визуализирует их движение под действием сил и импульсов в результате внешнего воздействия.

Добавление объектов в сцену через текстовый редактор

Редакторы объектов



Система вывода

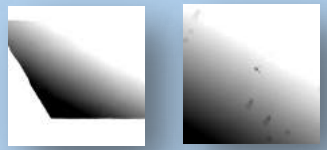


Определение пересечения объектов и применение физической модели

Рендер

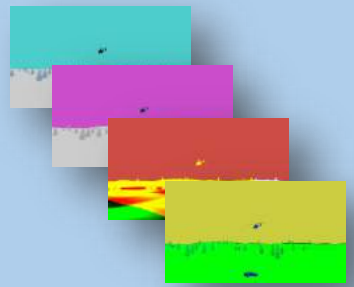
Объекты отправляются на рисовку

Карты теней

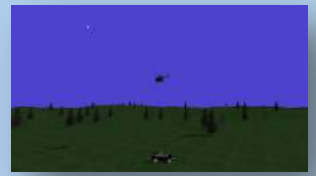


Цвет без освещения

G-Buffers



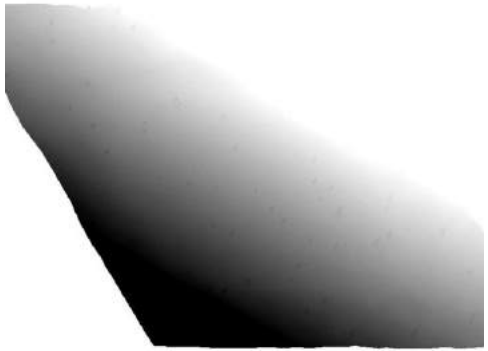
Освещение

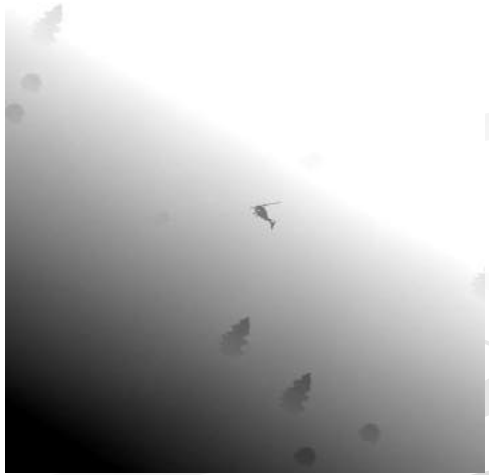


Отрисовка остальных объектов

Рендер. Каскадные тени

В проекте реализован метод создания теней через несколько каскадов.





Рендер. Отложенное освещение

Для отложенного освещения объектов была реализована модель Блинна-Фонга. Доступны различные виды источников света, такие как точечный, направленный и прожекторы.

Рендер. Отложенное освещение

1. Отрисовка объектов без освещения

2. Применение освещения

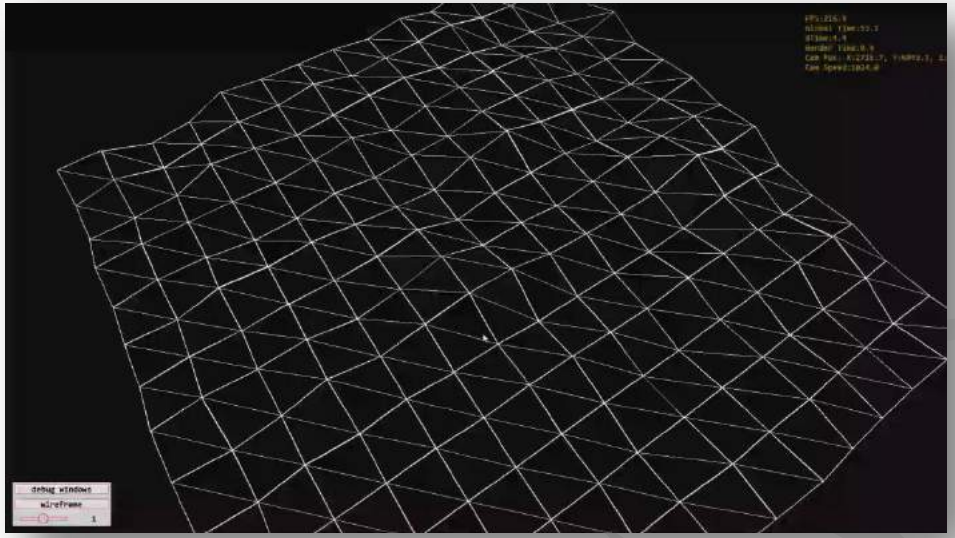
Рендер. Отложенное освещение

3. Отрисовка объектов, не требующих освещения

4. Отрисовка Sky Box’а

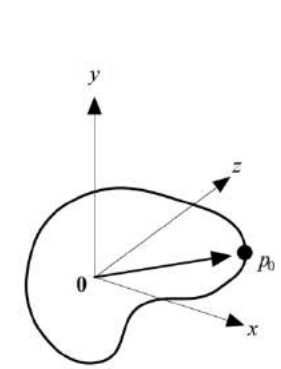
Рендер. Тесселяционный шейдер

Тесселяционный шейдер позволяет разбивать геометрию моделей на большее количество треугольников

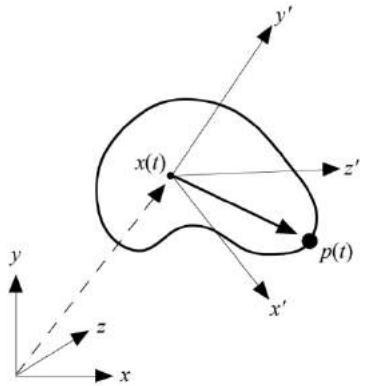


Физика

Положение тела в пространстве однозначно задаётся координатами его центра масс в некоторой условно неподвижной системе координат и матрицей поворота этого тела.



Система координат тела



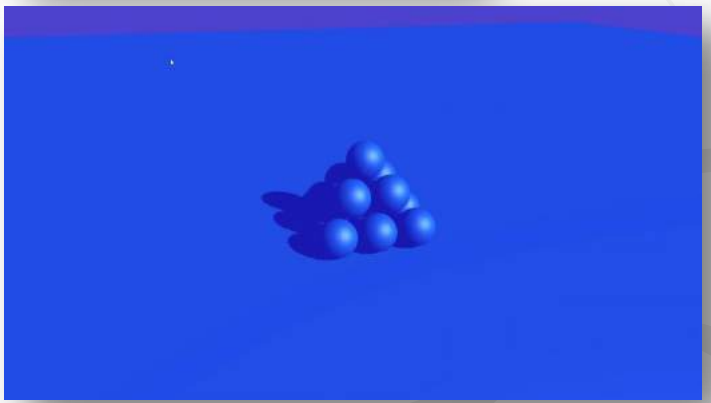
Мировая система координат

Физика

$$\vec{r} = \vec{r_0} + \vec{v_0}t + \frac{\vec{a}t^2}{2}$$

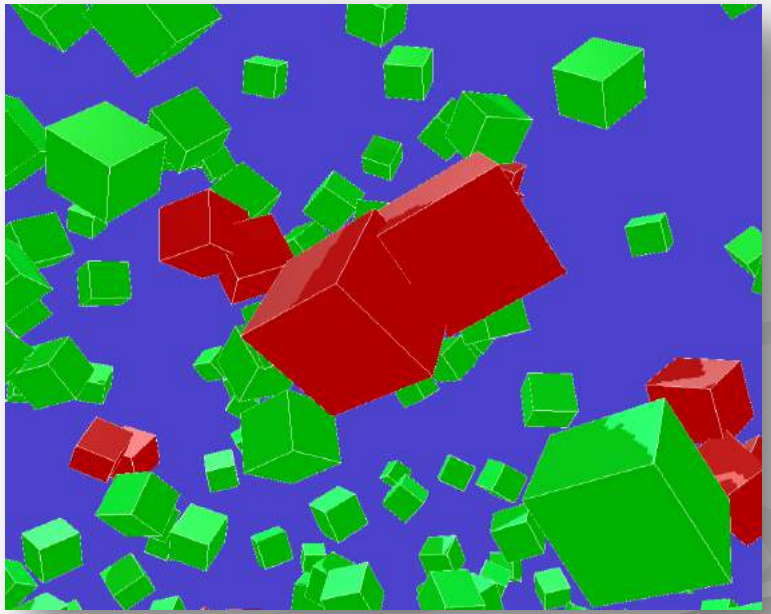
$$\vec{\alpha} = \vec{\alpha_0} + \vec{\omega_0}t + \frac{\vec{\varepsilon}t^2}{2}$$

Все движения раскладываются на поступательные и вращательные составляющие. Поступательное движение определяется характеристиками, такими как скорость и ускорение, а вращательное — угловой скоростью и угловым ускорением.



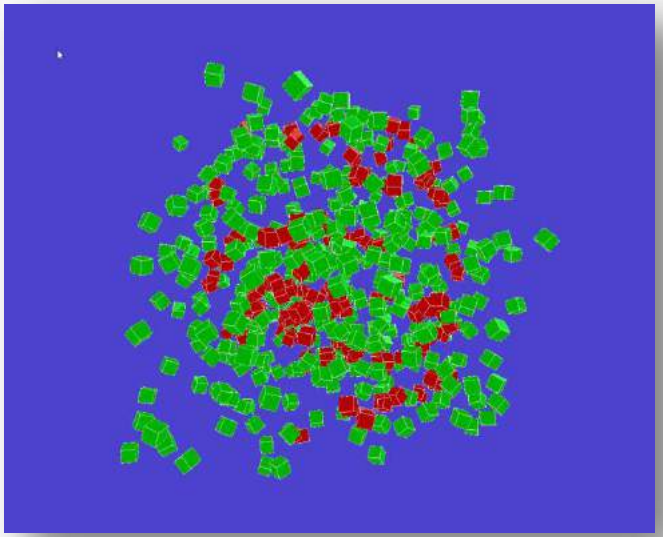
Коллизия

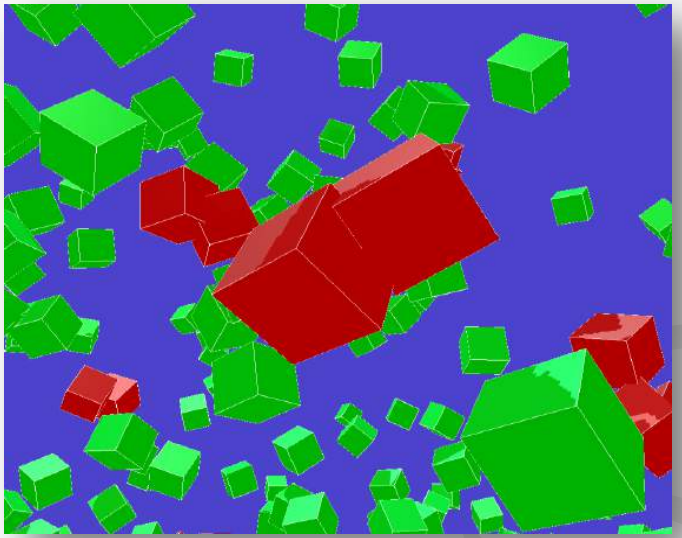
Для работы физической системы была реализована коллизия – система определения точек пересечения объектов физической модели.



Коллизия. Построение Kd-Tree

Структура Kd-Tree необходима для уменьшения количества проверок тел на соприкосновения.

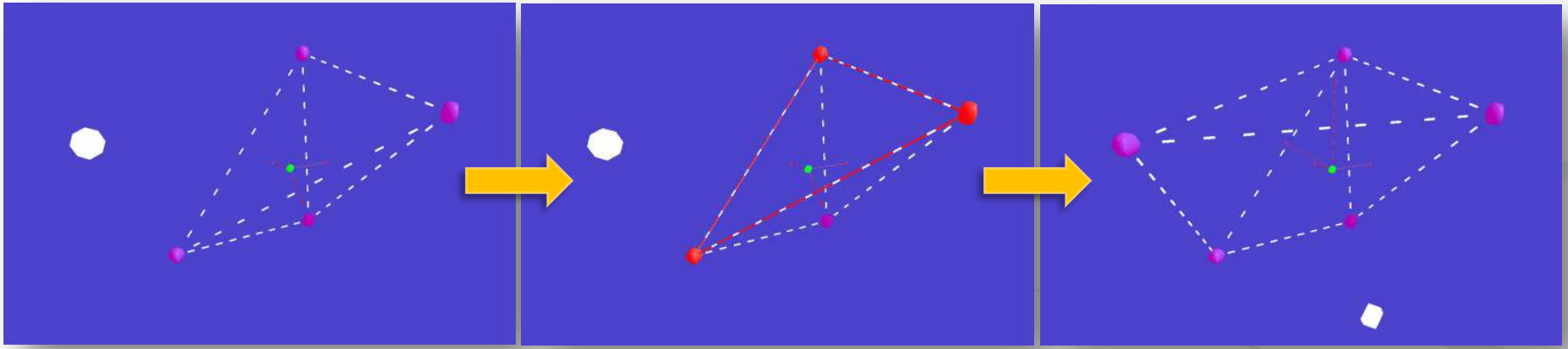




(Kd-Tree позволяет работать с большим количеством объектов)

Коллизия. Нахождение вектора проникновения

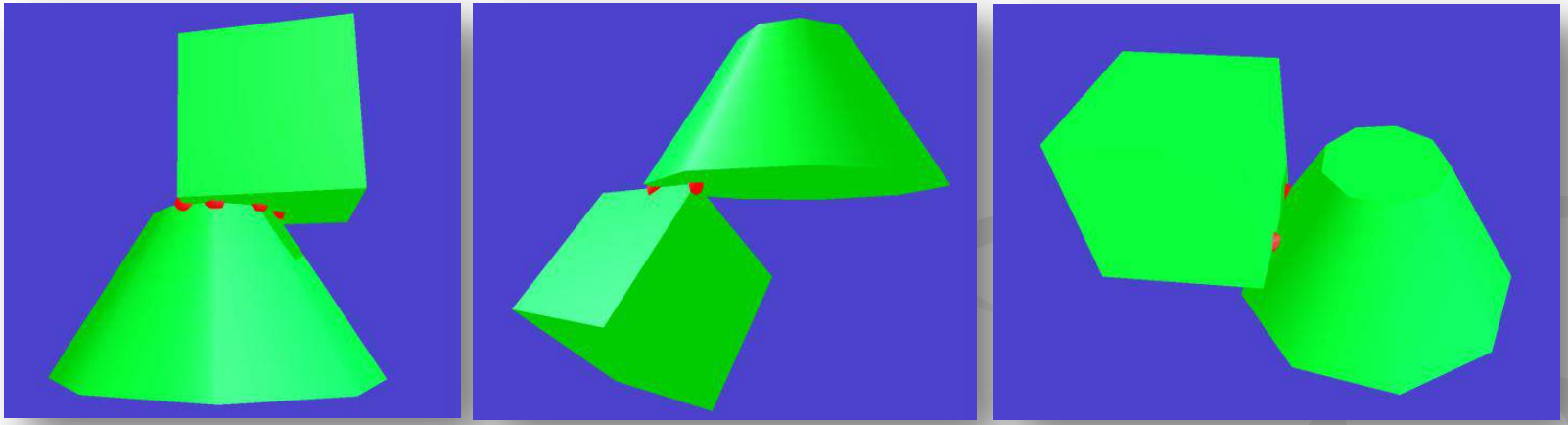
Вектор проникновения используется для определения расстояния между объектами, а также при поиске точек коллизии.



(Визуализация работы алгоритмов GJK и EPA)

Коллизия. Нахождение точек соприкосновения

Точки соприкосновения тел задействуются при применении физической модели.



CUDA

- Реализована система контейнеров, которая позволяет оптимизировать время копирования памяти с GPU и обратно.
- Используется для оптимизации нахождения пересечений объектов.

Вывод

В результате работы, нам удалось реализовать:

- система построения реалистичных физических процессов в реальном времени
- система определения пересечения твердых тел

В будущем планируется:

- развивать физические возможности проекта
- оптимизировать расчет коллизии
- добавить различные визуальные эффекты

Разработка кода

- Авторов: 7
- Языки: C++/HLSL/CUDA
- Среда разработки: Visual Studio 2019
- Время разработки: январь 2021 - март 2021
- Размер кода: 1.07 MiB
  (56 x *.cpp – 0.5 MiB , 66 x *.h – 0.5 KiB, 21 x *.hlsl – 76 KiB)
- Строк кода: 36100
  (*.cpp – 18500, *.h - 17600, *.hlsl – 2500)

- **Лазаренко Александр, 9-1 класс**
  Оптимизация на GPU с помощью CUDA API
- **Власов Дмитрий, 10-1 класс**
  Коллизия, система анимации, редактор моделей
- **Серков Александр, 10-1 класс**
  Физическая модель, система ресурсов, интерфейс
- **Черепков Петр, 10-1 класс**
  Физическая модель, интерфейс
- **Марков Максим, 10-2 класс**
  Построения геометрии объектов, визуальные эффекты
- **Юркин Александр, 10-2 класс**
  Система визуализации, взаимодействие с GPU
- **Иванов Святослав, 10-3 класс**
  Система освещения