

Разработка системы визуализации трёхмерных сцен для спортивных тренировок

Перед авторами была поставлена задача создания системы визуализации тренировки пользователей и отображении информации о их передвижении, пройденном расстоянии и вырабатываемой мощности. Актуальность нашего проекта заключается в том, что клиент может тренироваться в любое время и погоду. Наша система должна в реальном времени визуализировать передвижение пользователя по виртуальной местности. Необходимо получать данные о клиенте, что выполняется с помощью датчиков установленных на тренажёре (например, датчик счета мощности вращения колеса велосипеда), и, в зависимости от них, вычислить скорость движения пользователя. Для получения входных данных требуется велосипедный тренажёр с подключением по сети Bluetooth и персональный компьютер.

Для реализации нашего проекта мы использовали компьютеры с операционной системой Windows, был создан сервер для обработки нескольких пользователей, реализована обработка сигнала Bluetooth для получения данных с тренажёра, разработаны система построения столкновений для имитации реалистичного движения по рельефу и алгоритм инверсной кинематики для реалистичной анимации объектов.

Результатом работы стало создание системы трёхмерной визуализации велотренировки, позволяющей воссоздать опыт реальной поездки на велосипеде в виртуальной реальности, а возможность сетевого соединения позволяет проводить соревнования между несколькими людьми. Разработанная нами система анимации наряду с реалистичной графикой позволяет совершить полное погружение. Созданные подсистемы обнаружения столкновений, визуализации движения модели человека по опорным точкам скелета повышает уровень отображения.

В ходе проделанной работы, авторы создали полноценную систему визуализации для тренировок велосипедистов, отличительной чертой которой является простота в применении. В ближайшем будущем авторы планируют добавить возможность подключения и использования очков виртуальной реальности, добавить звуковое оформление программы, улучшить графическую составляющую, а также распространить проект.

Разработка системы визуализации трёхмерных сцен для спортивных тренировок

Двас Павел, 9-2 класс

Кураленок Святослав, 9-4 класс

Амбросовская Дарья, 10-2 класс

Вашпанов Александр, 10-3 класс

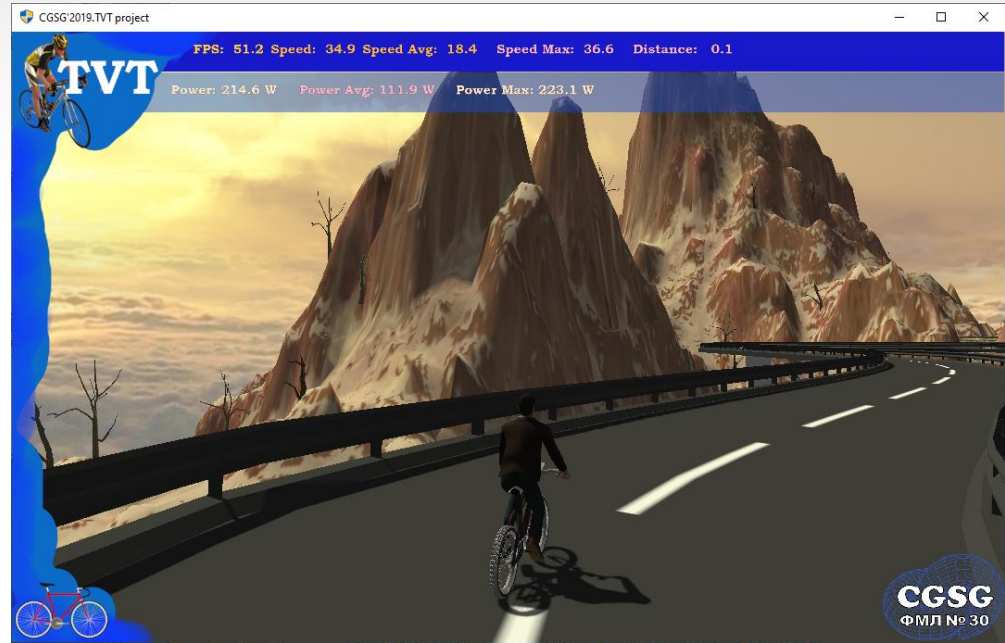
Кононов Святослав, 10-3 класс

Малахов Дмитрий, 11-3 класс

Григорович Вячеслав, 11-5 класс

Гирвиц Александр, 11-6 класс

Дмитриева Василиса, 11-6 класс



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

Computer Graphics Support Group

Санкт-Петербургский губернаторский физико-математический лицей № 30

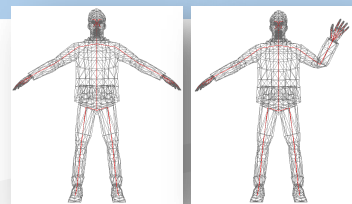
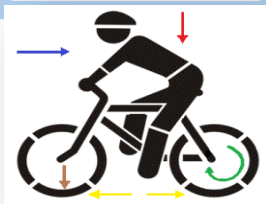
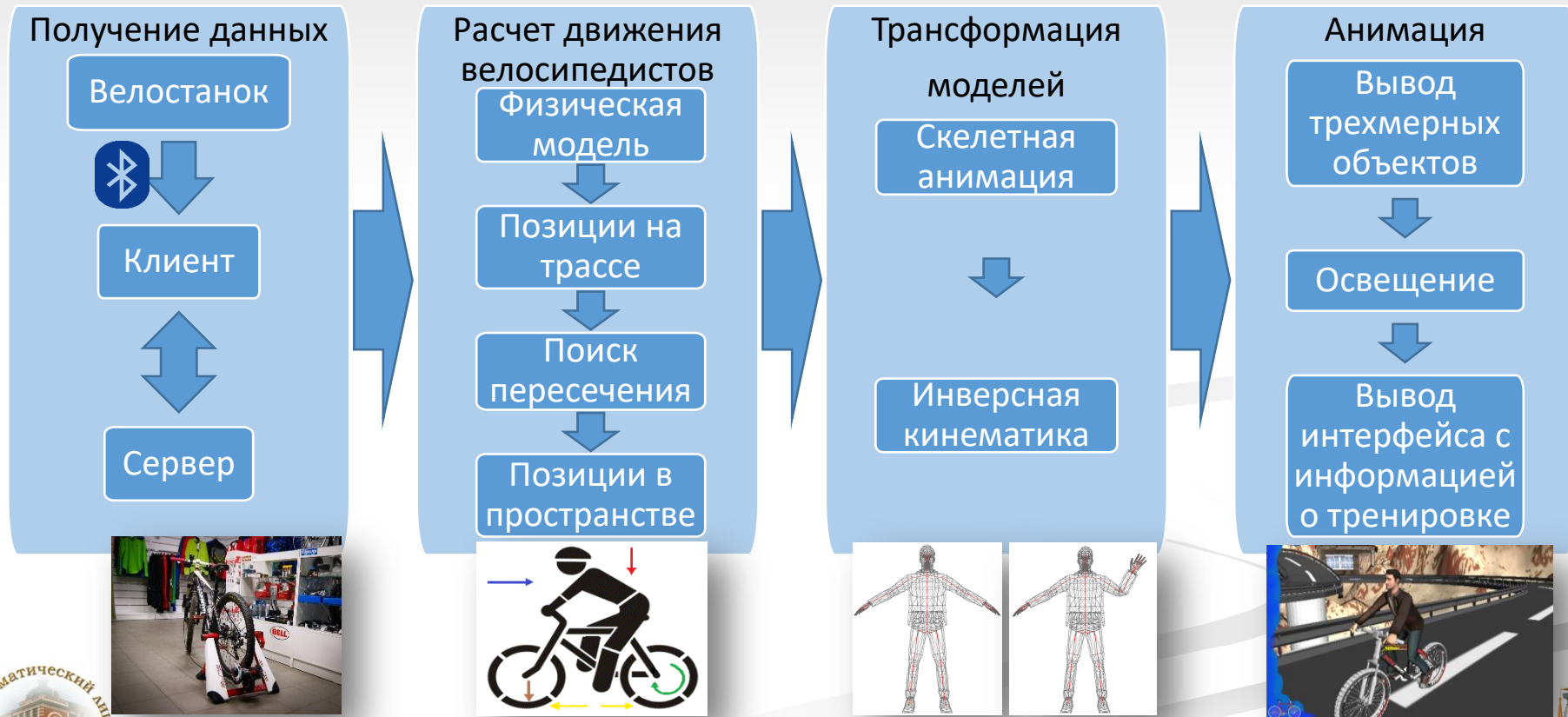


Постановка задачи

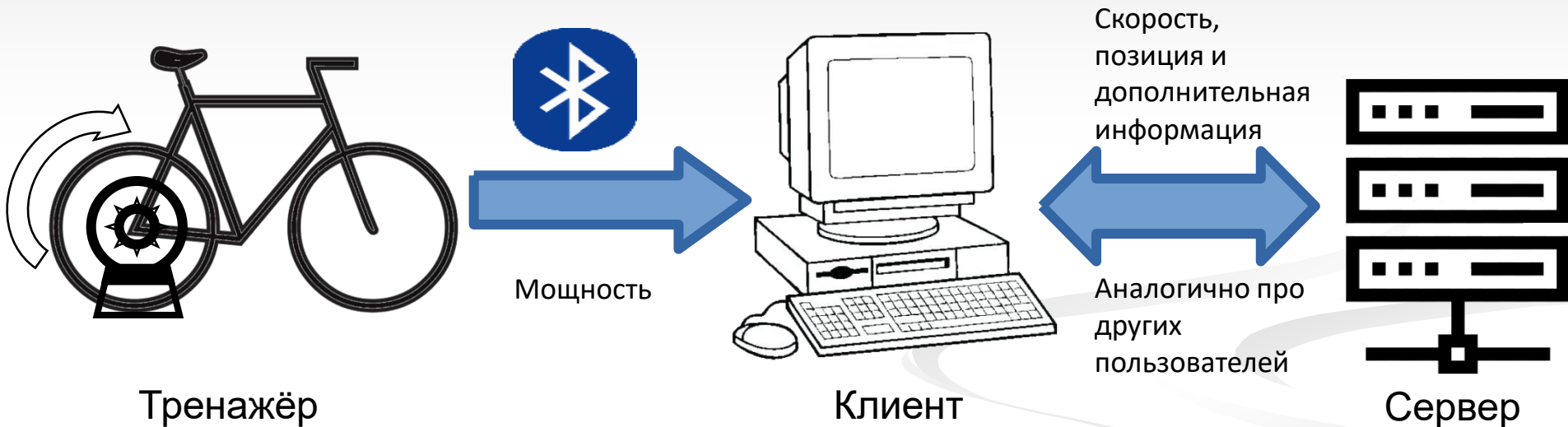
Разработать программное обеспечение, позволяющее проводить тренировки:

- 1) с использованием технологий виртуального погружения
- 2) с возможностью взаимодействия по сети с другими игроками
- 3) предоставляющее пользователю результаты тренировки для дальнейшего анализа

Структура проекта



Получение данных с тренажёра



Получение данных с тренажёра

Пользователь получает с тренажёра по протоколу Bluetooth low energy мощность в ваттах и передаёт ее на обработку физической модели для подсчёта скорости. Потом скорость вместе с другими параметрами, такие как позиция и , на сервер, при этом получая с него аналогичную информацию о других пользователях.

Физическая модель

$$F_{Grav} = m * g * \sin(\alpha)$$

$$F_{Roll} = m * g * \cos(\alpha) * C_{rr}$$

$$F_{Air} = 0.5 * \rho_{air} * S * (v_{bike} + v_{wind})^2$$

$$F_{Res} = F_{Grav} + F_{Roll} + F_{Air}$$

$$A = (P * \mu - F_{Res} * v_{bike}) * \Delta t$$

$$v_{new} = \sqrt{v_{bike}^2 + 2 * A / m}$$

α — угол склона

C_{rr} — коэффициент сопротивления вращению

ρ_{air} — плотность воздуха

S — фронтальная площадь велосипеда

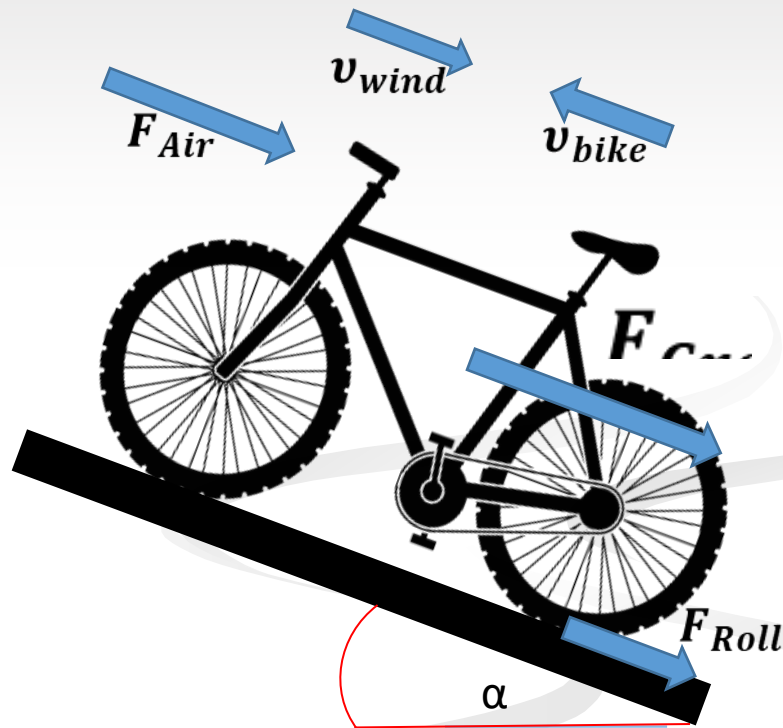
P — вырабатываемая мощность

μ — КПД велосипеда

v_{bike} — текущая скорость велосипеда

v_{wind} — скорость встречного ветра

v_{new} — новая скорость велосипеда

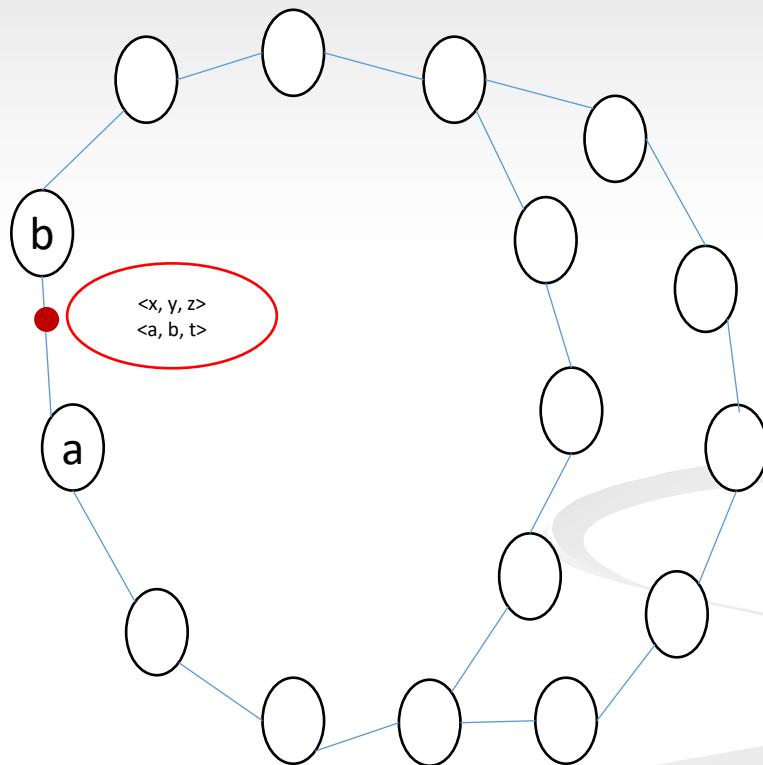


Физическая модель

В расчёте скорости велосипеда на сцене используется модель изменения кинетической энергии за счёт работы, совершённой кручением педалей велотренажёра.

При переводе мощности, вырабатываемой пользователем на велотренажёре, в мощность, вырабатываемую на велосипеде на сцене, должны учитываться силы сопротивления, такие как притяжения, трения и воздуха. На слайде приведены расчётные формулы для нахождения сил сопротивления, совершённой полезной работы и конечной скорости велосипеда.

Трасса



Трасса

Трасса представляет из себя граф, вершины которого соединены между собой.

Для большей детализации берётся большое количество вершин путем интерполяции.

Для оптимизации при загрузке графа сохраняются расстояния между соединёнными вершинами.

При движении велосипедиста по графу высчитывается позиция, в которой он окажется, зная вычисленную ранее скорость велосипедиста с помощью физической модели, его текущую позицию и направление движения.

Для передачи позиции пользователя на сервер с дальнейшей возможностью рассчитать его положение наперёд используются два номера вершин графа, между которыми находится велосипедист, и соотношение позиции пользователя к длине отрезка между двумя данными графами.

Алгоритм поиска пересечений

Разбиение пространства на воксели
(прямоугольные параллелепипеды)



Построение луча



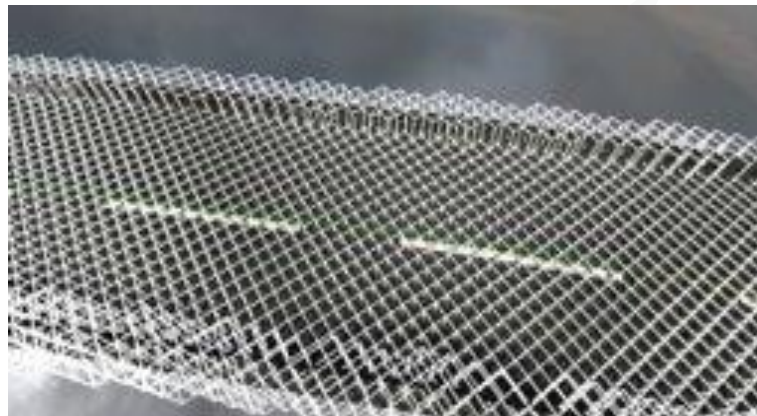
Нахождение вокселей, в которых
проходит луч



Поиск пересечения луча с треугольниками,
входящими в найденные воксели



Постановка объекта на рельеф



Алгоритм поиска пересечений

Система поиска пересечений используется для реалистичного отображения движения объектов по сложному рельефу. Для разбиения пространства на воксели нами был разработан оптимизированный алгоритм нахождения пересечения между треугольником и прямоугольным параллелепипедом, а для того, чтобы найти те воксели, в которых проходит луч, был создан алгоритм, сложность которого не зависит от количества треугольников, благодаря дискретному разбиению пространства. Когда луч построен и найдено пересечение с рельефом, начинается процесс постановки объекта.

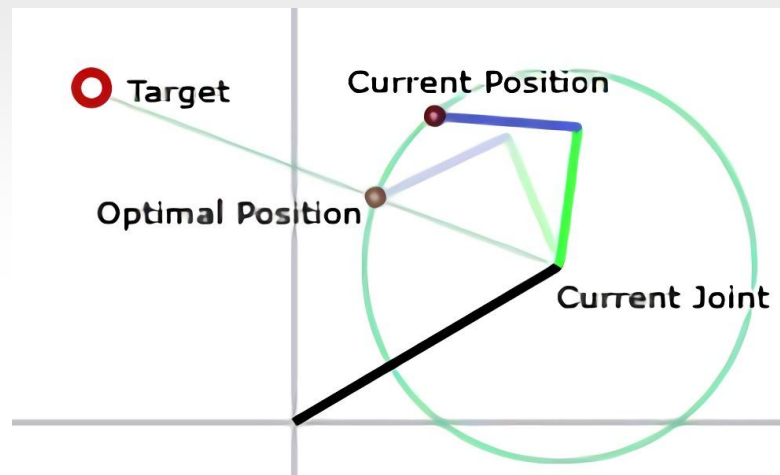
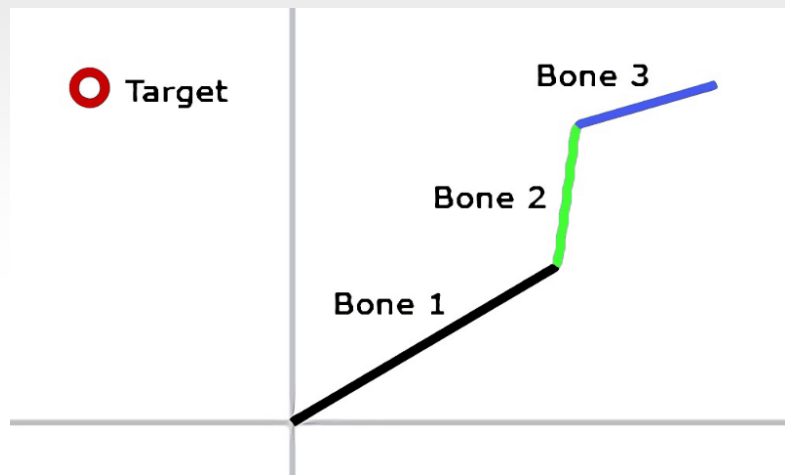
Постановка объектов на рельеф



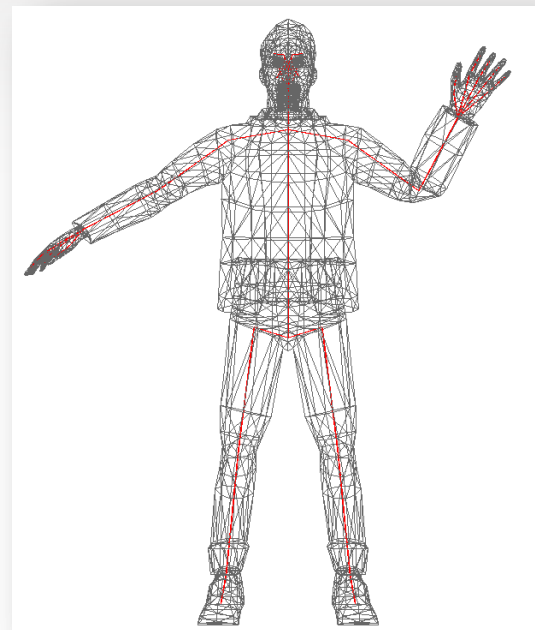
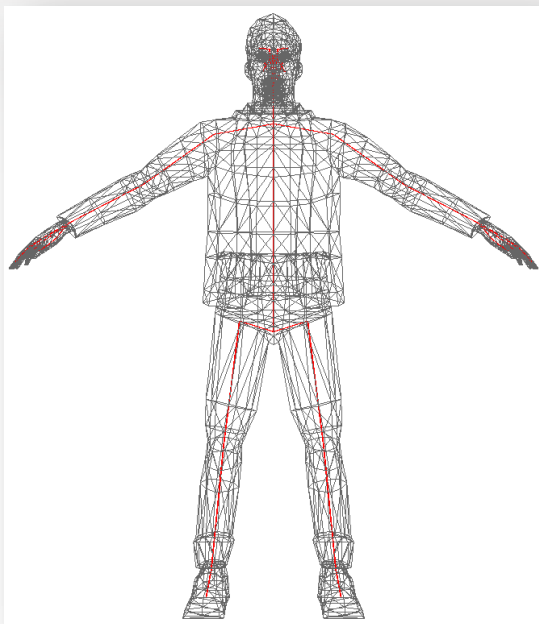
Постановка объектов на рельеф

Алгоритм поиска пересечений используется нами для размещения деревьев на ландшафте и для реалистичной имитации движения велосипеда по трассе. Чтобы поместить велосипед на трассу, мы строим два вертикальных луча через центры его колёс и находим их пересечение с рельефом, на основе полученных двух точек осуществляется постановка велосипеда.

Инверсная кинематика



Движение ног осуществляется при помощи алгоритма ccd(Cyclic Coordinate Descent(Метод координатного спуска)). Начиная с предпоследней точки, угол между направлением на цель и направлением на конечную точку стремится к минимуму с учетом ограничений.



Skinning

Разрабатывая систему мы столкнулись с проблемами нереалистичного перемещения. Для решения проблемы нереалистичного движения велосипедиста авторы использовали подход скелетной анимации. При этом подходе каждая вершина модели относится к своему суставу(joint), благодаря чему можно передвигать только суставы(joints). У каждой вершины есть свой вес, что помогает определить на сколько данный сустав(joint) будет влиять на вершину.

Skinning

$$v_{out} = \sum_{i=0}^n ((v \cdot BSM) \cdot IBM_i \cdot Tr_i) \cdot w \quad w_s = \sum_{i=0}^n w = 1$$

v_{out} - итоговая вершина

v - начальная вершина

n - количество joints

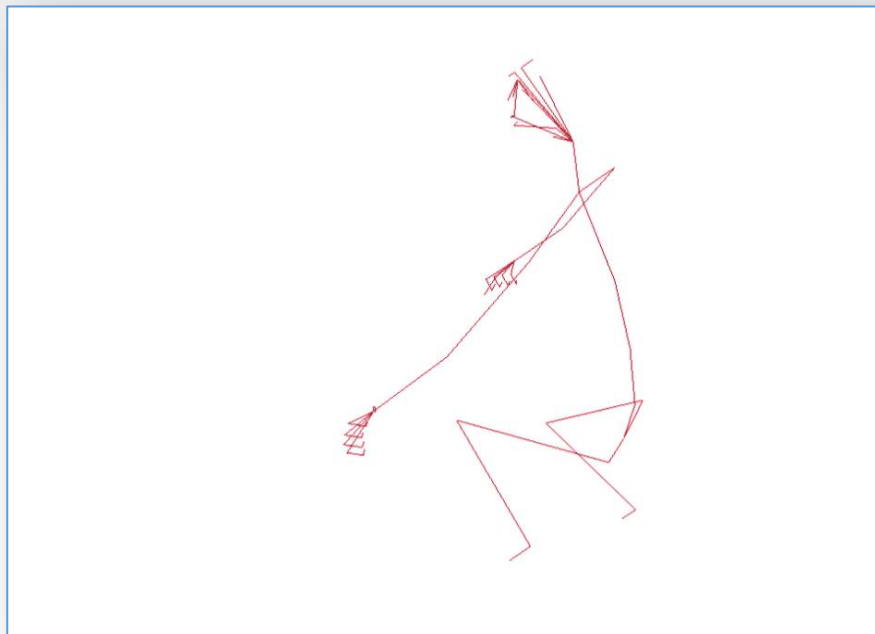
BSM - начальная матрица, на которую умножаются все вершины

IBM_i - обратная матрица позиции для сустава (joint) на i позиции
(возвращает вершину из локальных координат в мировые)

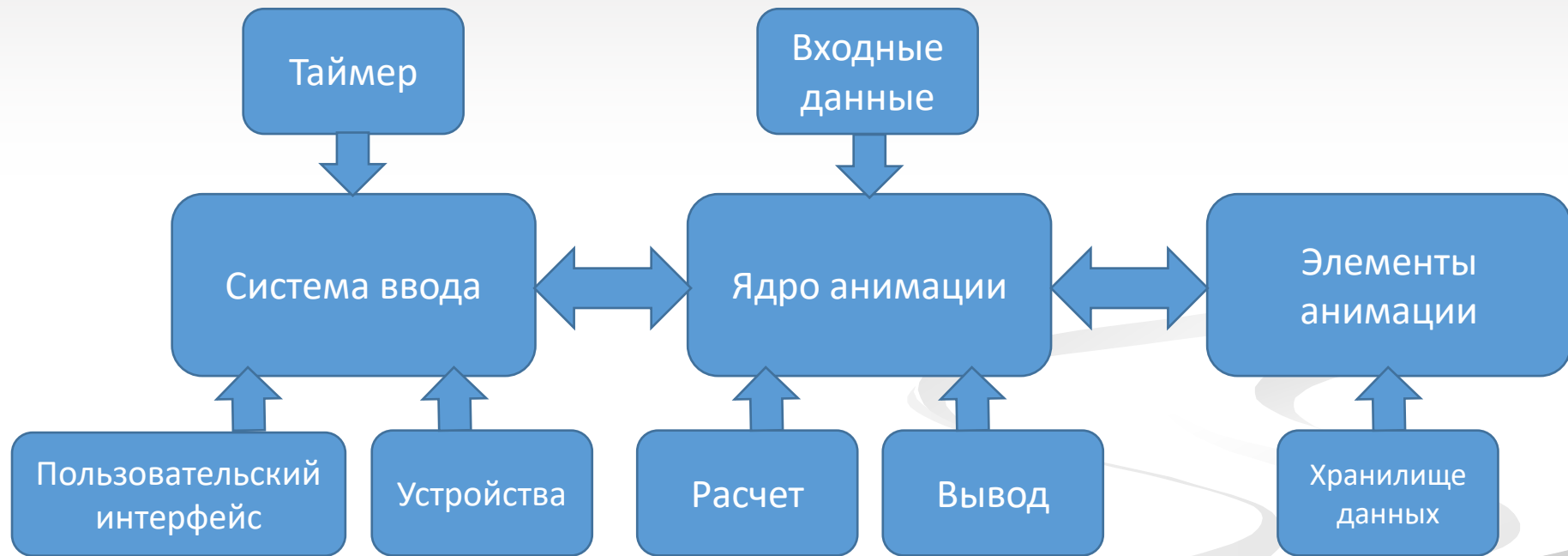
Tr_i - матрица трансформации для сустава (joint) на i позиции

w - вес данного сустава (joint) на вершину v

Skinning и инверсная кинематика



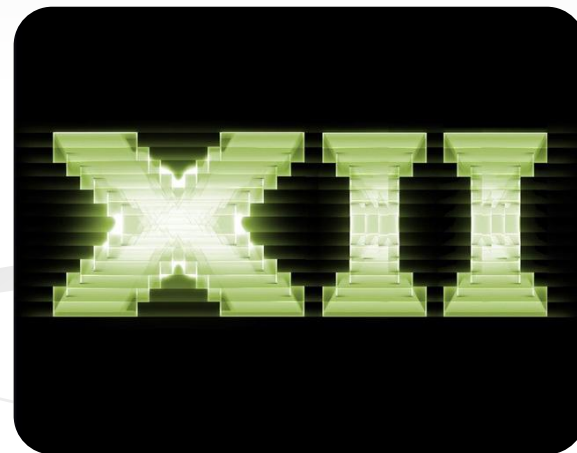
Пример работы инверсной
кинематики и Skinning



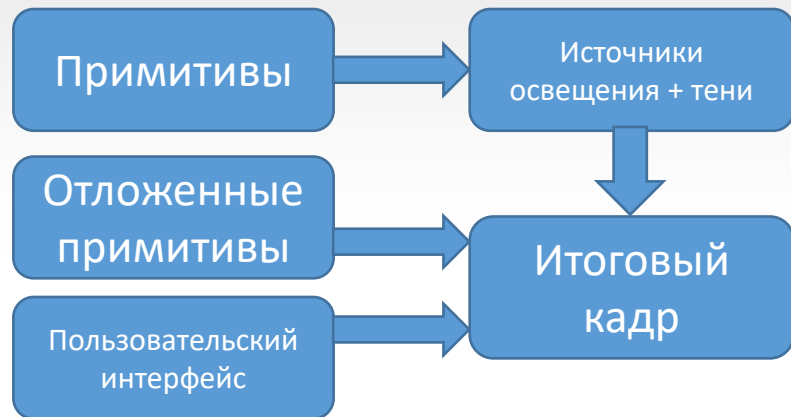
DirectX 12

DirectX 12 – это современная графическая библиотека. Одна из главных особенностей – увеличенные возможности программы и лучшая производительность из-за уменьшения нагрузки на драйвер.

Отрисовка объектов происходит с помощью составлений списков команд, которые можно записывать на разных потоках и выполняются параллельно на видеокарте. На видеокарте все обрабатывается при помощи шейдеров – параллелизуемых микропрограмм.



Отложенное освещение



Result



+



+



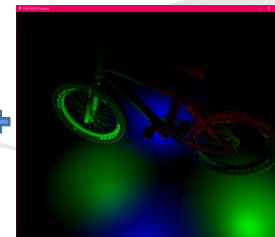
+



+



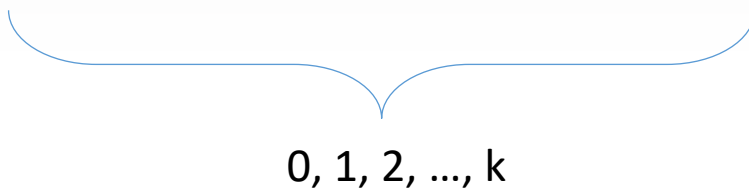
+



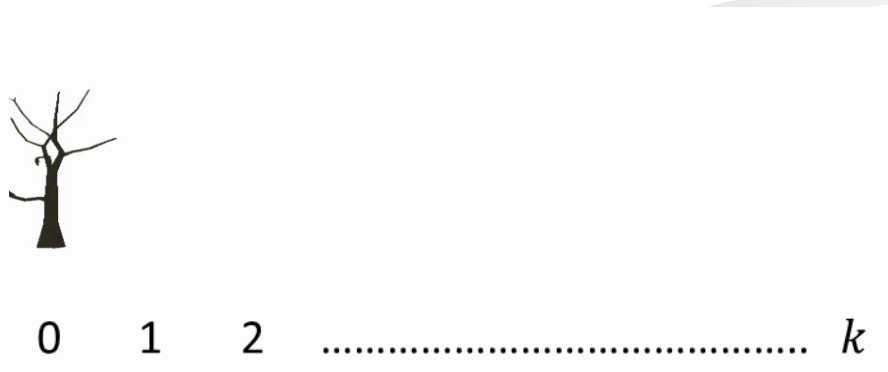
G-buffers

Instanting

Вершины модели



Matrix pool



Computer Graphics Support Group

Санкт-Петербургский губернаторский физико-математический лицей № 30

Instancing

В результате добавления мы столкнулись с проблемой множества объектов. Для того чтобы уменьшить количество данных, которые хранятся при рисовке, мы придумали Instancing. Его суть в том, что вместо того, чтобы хранить все вершины одинаковых моделей, мы храним один набор вершин одной модели и матрицы в `matrixpool` (место хранения матриц для моделей).

Карты теней

Карты теней – алгоритм для визуализации теней.

Создается текстура с расстояниями от источника света к ближайшим объектам и при отрисовке объекта расстояние до него сравнивается с расстоянием в текстуре.

Для создания необходимо адаптивно выбирать область видимости текстуры.

Каскадные тени – разновидность алгоритма с несколькими картами теней на разные области видимости (в зависимости от дальности)



Environment



Computer Graphics Support Group

Санкт-Петербургский губернаторский физико-математический лицей № 30

Environment

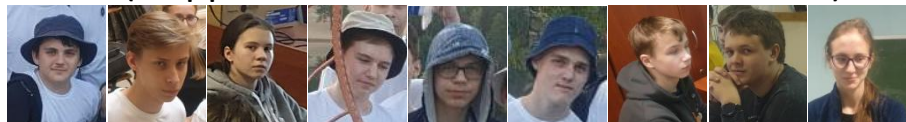
Для создания окружения были совмещены алгоритмы Instancing и поиск пересечений. При помощи поиска пересечений ищется точка, куда поставить объект, по ней строится матрица для того, чтобы вывести его при помощи Instancing. Это помогает создать более реалистичную картинку. Так же для создания окружения был реализован Skybox (или же небо).

Заключение

- Создана система для велосипедных тренировок
- В ходе проделанной работы были решены задачи рендеринга сцены, анимирования моделей, постановки объектов на трассу, получения данных с велотренажера, сетевого взаимодействия
- В будущем планируется добавить поддержку очков виртуальной реальности
- Авторы планируют популяризовать проект

Разработка кода

- Авторы: 9
- Языки: C++/HLSL
- Время разработки:
ноябрь 2019 - апрель 2020
- Размер кода: 1938 KiB
(60 x *.cpp – 471 KiB, 98 x *.h – 1377 KiB,
23 x *.hlsl – 90 KiB)
- Строк кода: 28000
(* .cpp – 16000, *.h - 14000, *.hlsl – 3700)



- **Двас Павел, 09-2 класс**
система ввода, структурирование данных, полученных от тренажера по протоколу Bluetooth Low Energy, система клиентов для передачи данных на сервер
- **Кураленок Святослав, 09-4 класс**
подсистемы вывода и освещения
- **Амбросовская Дарья, 10-2 класс**
подсистемы вывода, тени и низкоуровневое программирование видеоадаптера
- **Вашпанов Александр, 10-3 класс**
система поиска пересечений лучей с объектами, постановка объектов на рельеф с использованием системы поиска пересечений, подсистема анимации
- **Кононов Святослав, 10-3 класс**
вывод звука, обработка данных тренажера
- **Малахов Дмитрий, 11-3 класс**
система ввода, низкоуровневое общение с устройством по протоколу Bluetooth Low Energy для возможности получения данных, сервер для передачи данных между пользователями
- **Григорович Вячеслав, 11-5 класс**
инверсная кинематика, модуль математики с использованием SSE инструкций, генерация ландшафта по шуму и прокладывание трека по нему
- **Гирвиц Александр, 11-6 класс**
подсистема работы с иерархией моделей, работа со скелетной анимацией для реалистичных движений, работа с окружением сцены с использованием поиска пересечений
- **Дмитриева Василиса, 11-6 класс**
подсистема вывода инстансированных объектов с помощью matrixpool, работа с matrixpool, создание конвертора для нового формата моделей и их загрузка