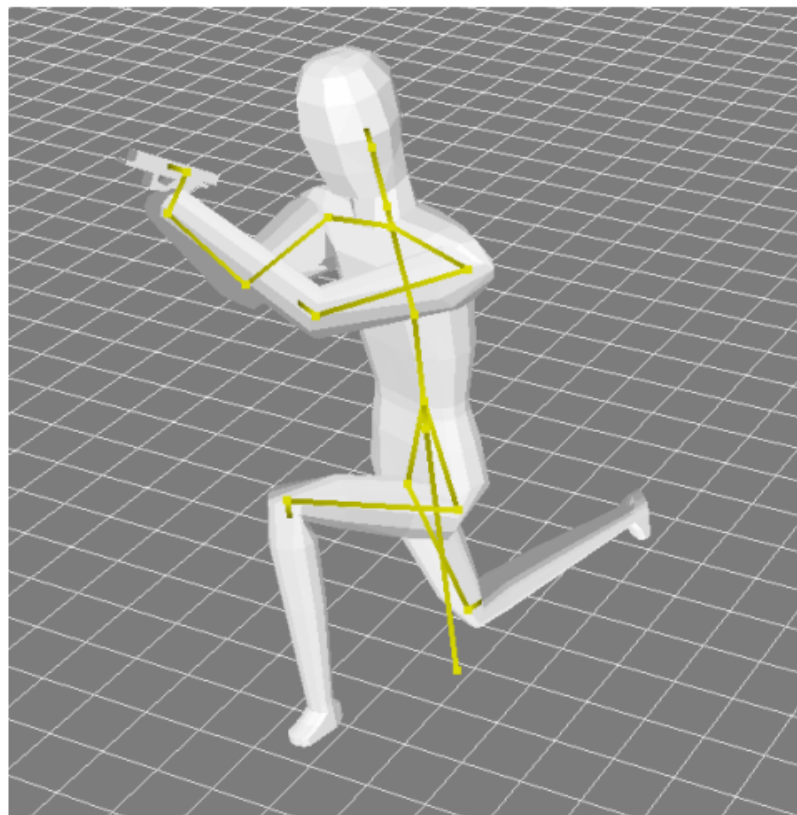
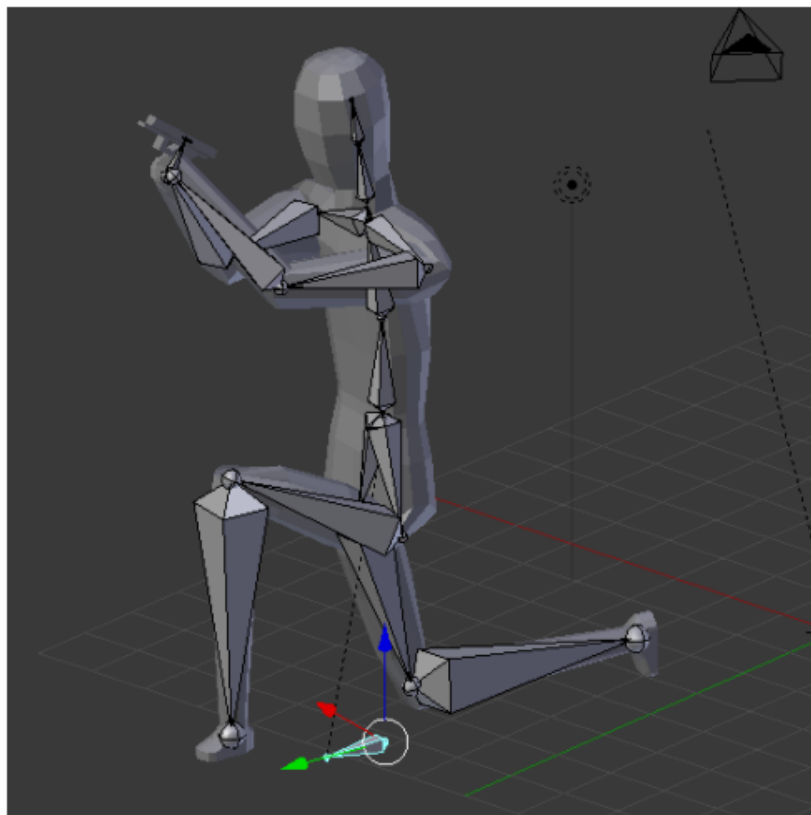


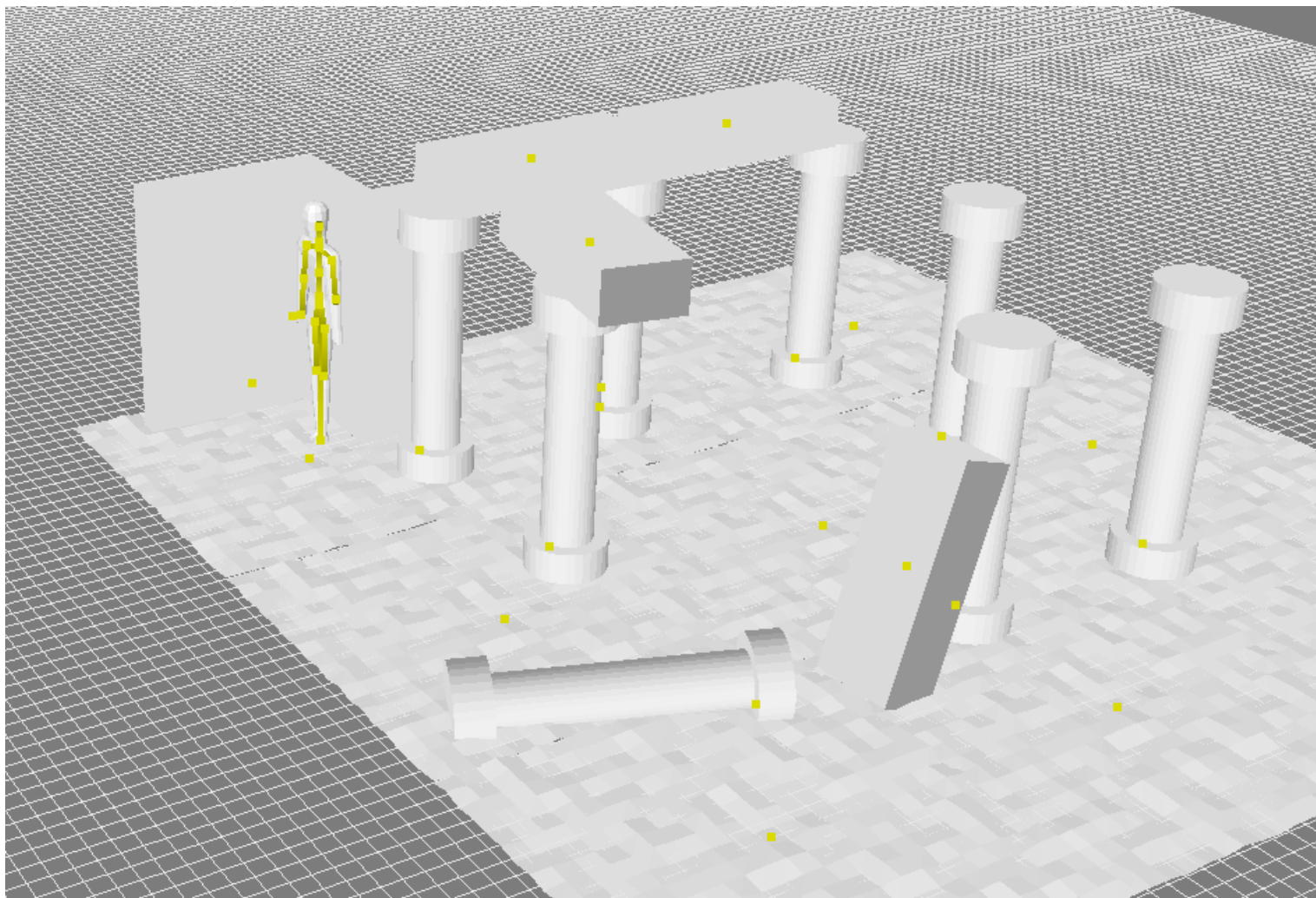
Конвертер трехмерных моделей



Ключевые особенности

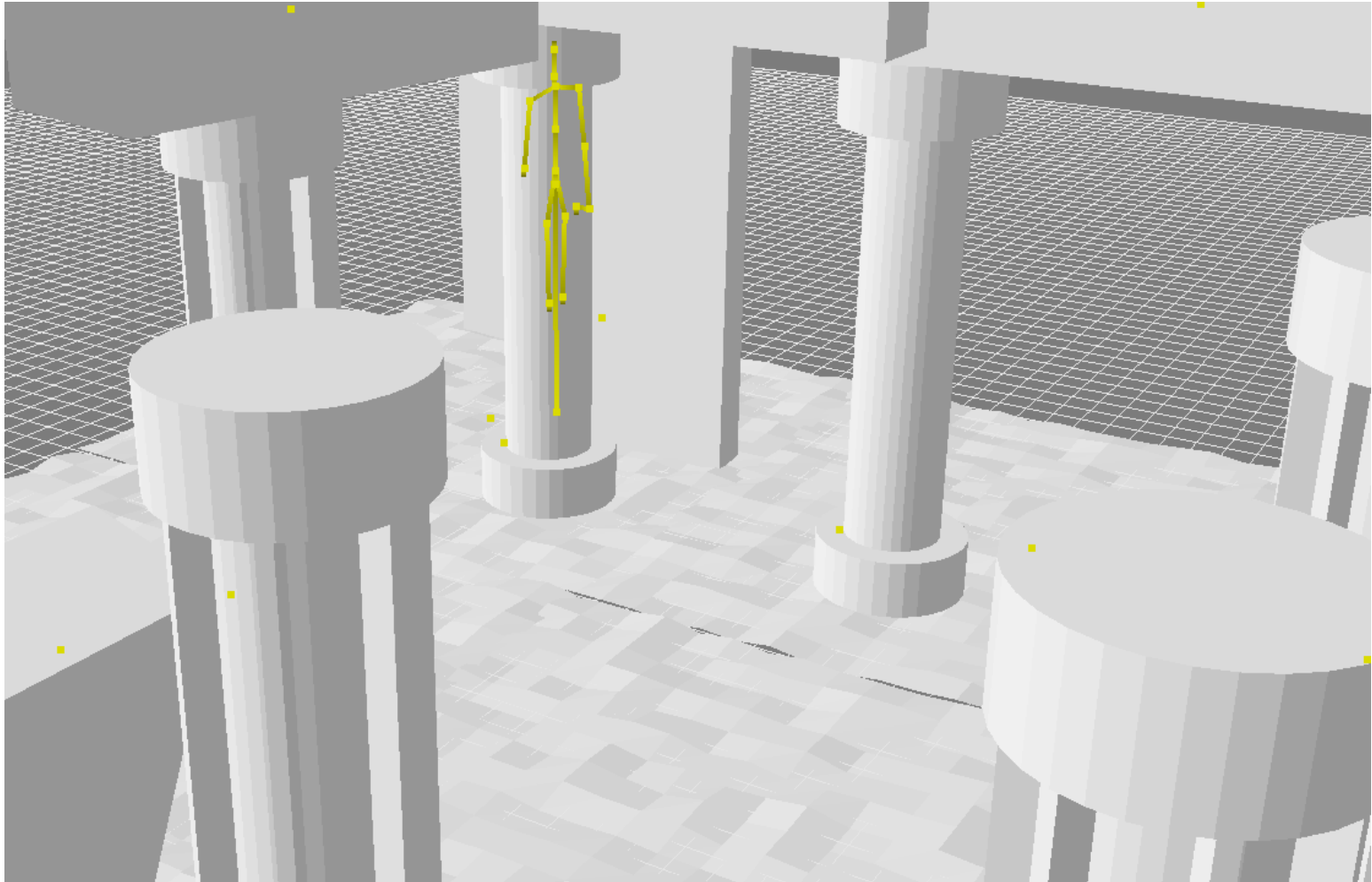
- Поддержка сцен и экземпляров моделей
- Поддержка скелетной анимации с весами
- Возможность импорта нескольких моделей в одну
- Возможность записи последовательности обработки модели и дальнейшего использования в качестве скрипта
- Можно указывать формат вершины, благодаря чему можно оптимизировать потребление памяти, вплоть до двумерной графики
- Обработка модели и её рендеринг, включая GUI, осуществляются параллельно.
- Использует Rust и Glum

Сцены и экземпляры



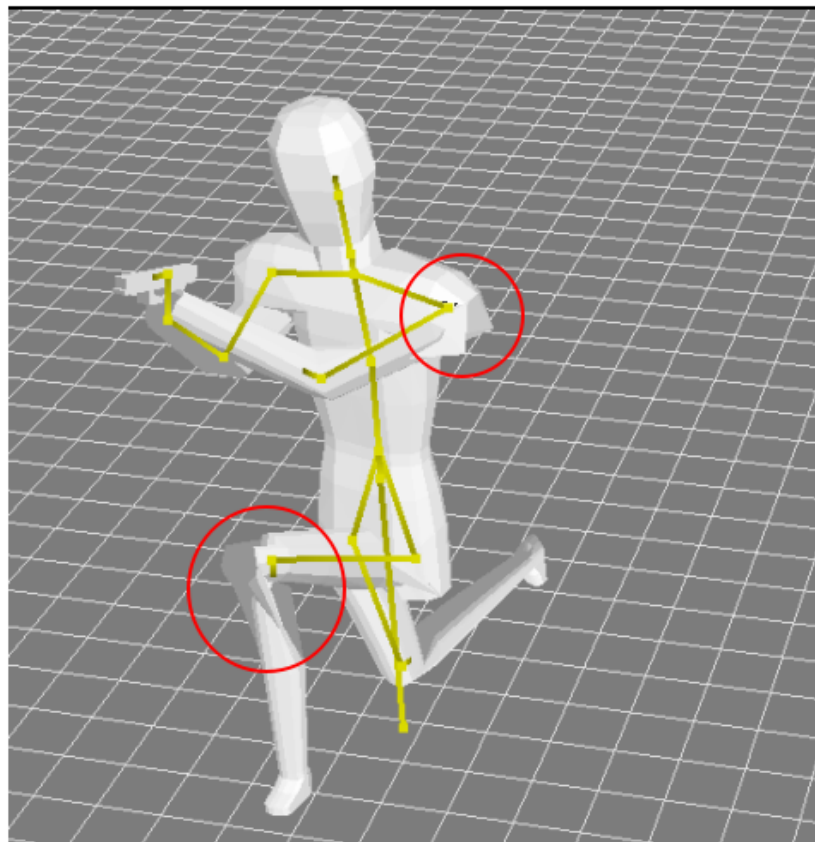
Поддержка LOD

Коллонны, что ближе, отличаются от тех, что дальше.

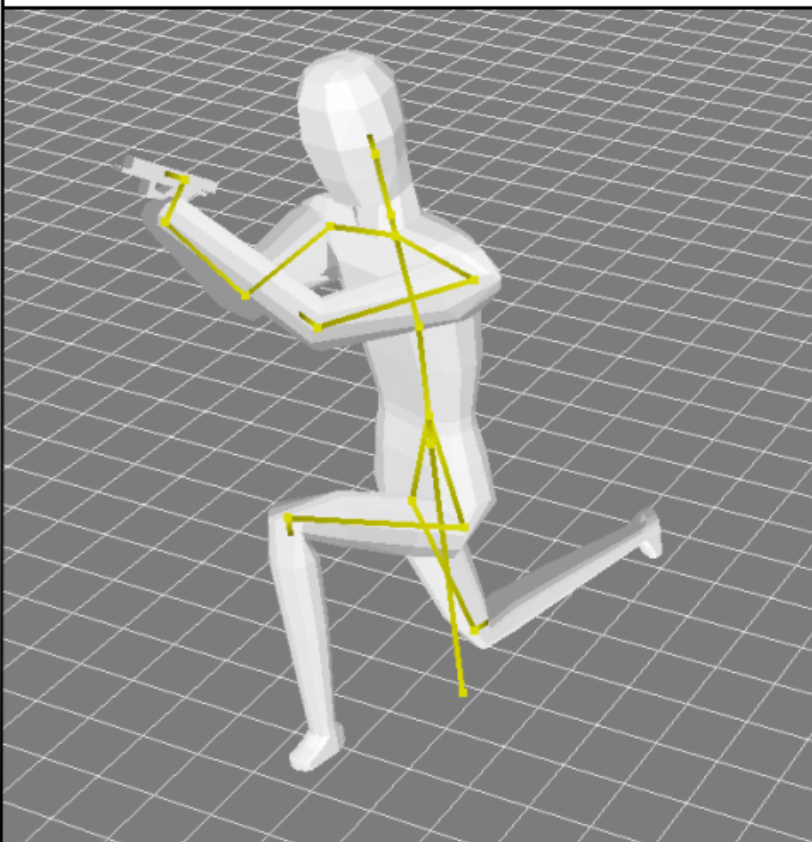


Поддержка скелетной анимации с весами

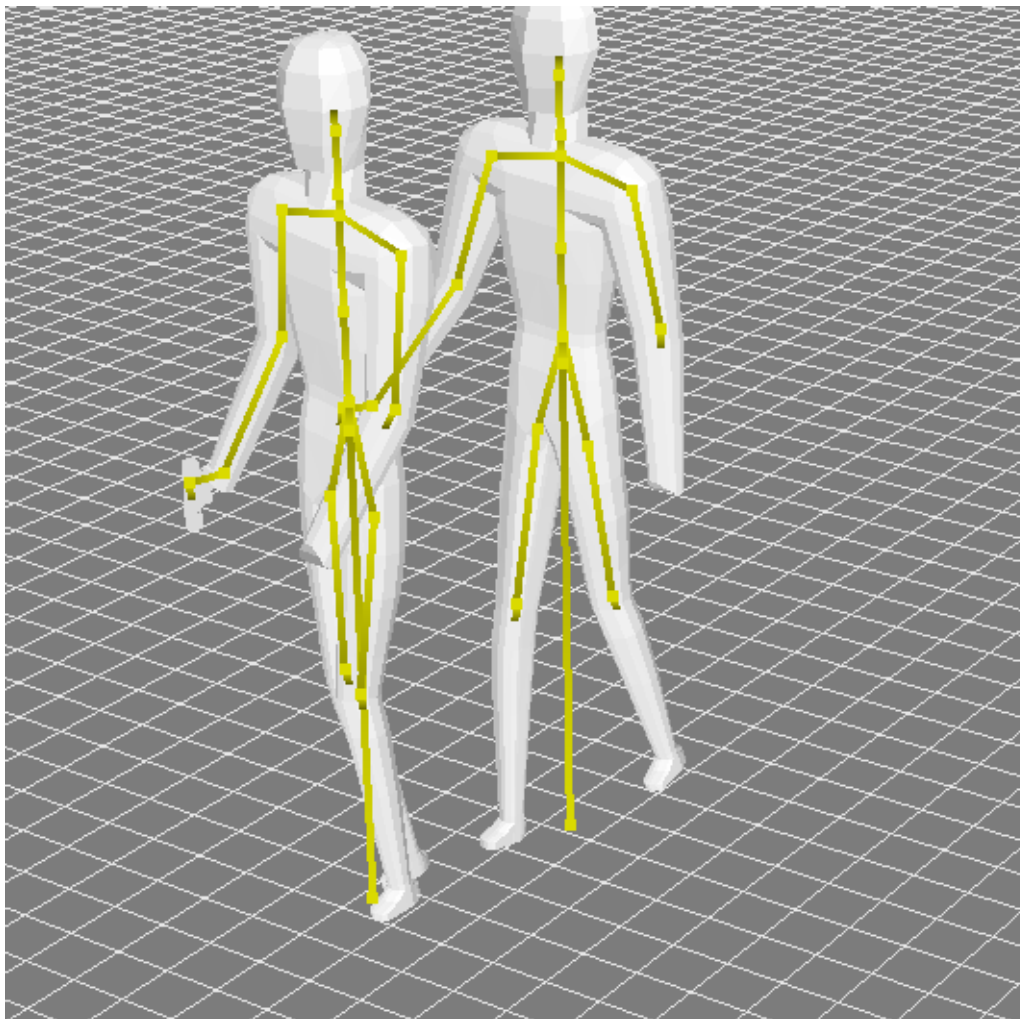
К одной кости без весов



К нескольким с весами



Возможность импортировать несколько моделей



Формат вершины

Формат вершины задается в следующей форме:

```
"VERTEX:(X:f32,Y:f32,Z:f32) NORMAL:(X:f32,Y:f32,Z:f32) BONE0:  
(INDEX:i32,WEIGHT:f32) BONE1:(INDEX:i32,WEIGHT:f32) BONE2:  
(INDEX:i32,WEIGHT:f32)"
```

Где:

VERTEX:(X:f32,Y:f32,Z:f32) Позиция вершины

NORMAL:(X:f32,Y:f32,Z:f32) Нормаль из вершины

BONE0:(INDEX:i32,WEIGHT:f32) Нулевая и последующие кости

BONE1:(INDEX:i32,WEIGHT:f32)

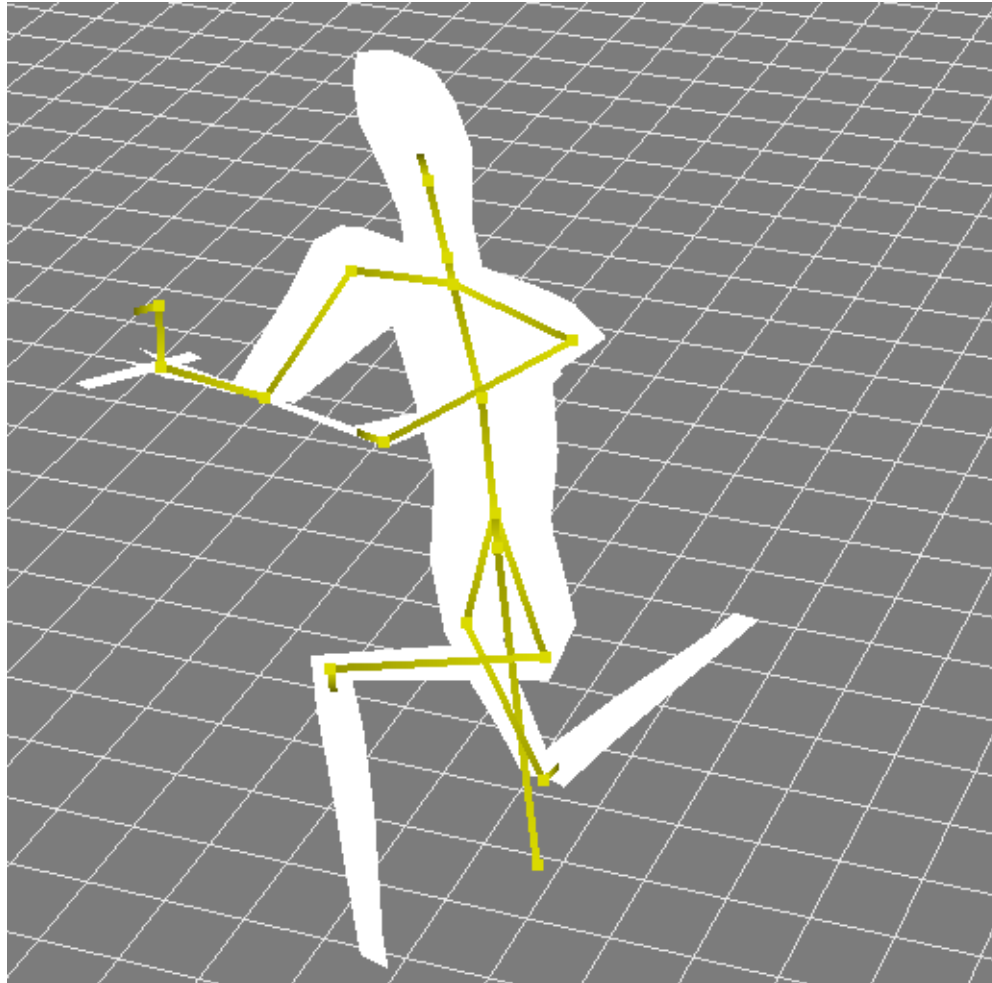
BONE2:(INDEX:i32,WEIGHT:f32)

Так же можно использовать i16, или сгруппировать например так:

```
VERTEX.X:i16 NORMAL.X:f32 BONE0.INDEX:i16 BONE1.INDEX:i16 WEIGHT0:f32,  
WEIGHT1:i32
```

Форматы вершины

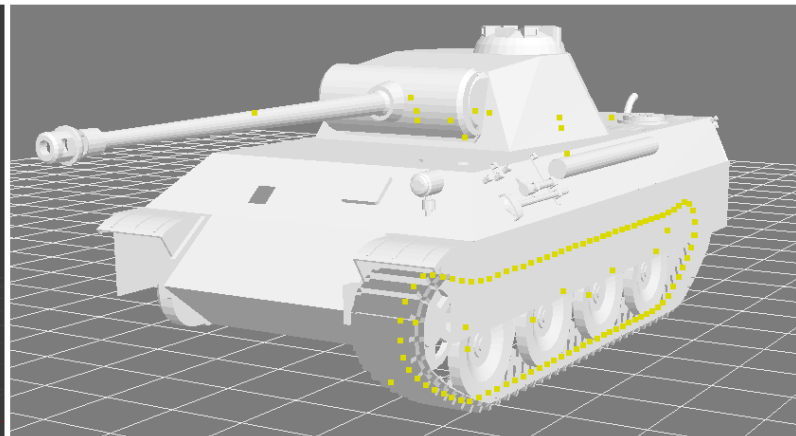
Поскольку 2D графика устроена так же, как и 3D, то есть возможность превратить трёхмерную модель в двухмерную



Многопоточность

Модель PzV Panther обладает высокой детализацией, поэтому она загружается примерно 2-3 секунды. Чтобы избежать зависания, у меня используются 2 потока:

- Render, который рендерит модель, обрабатывает GUI, а так же загружает готовые данные в видеопамять.
- Process, который слушает и исполняет приказы, строит геометрию и через канал отправляет её Render-у для последующей загрузки в видеопамять.



Rust и Glum

Чтобы обеспечить максимальную надежность, я использую язык программирования Rust и библиотеку Glum — высокоуровневую обертку над OpenGL. Я всячески стараюсь избежать потенциально опасного и скучного API для C. Благодаря особенностям Rust, гонки данных, переполнение стека, утечки памяти, `segfault` не возможны в принципе или легко поддаются обнаружению. При этом Rust обладает быстродействием C++ и элементами функционального программирования, которые выведены в ежедневную практику. Поэтому для моих целей он является идеальным языком программирования, что не удивительно – по статистике он является самым любим ЯП.



Спасибо за внимание

Выполнил
Шляков Антон
гр. Р3317