

Overview

Max Ortner

10K Program

Language: Python

This program was created to easily retrieve 10K documents from the SEC's website. It works by inputting a company ticker and will retrieve and download the last five years of 10K documents from SEC.gov.

Dashboard

Language: Python

This program comes in two versions: the game version, and the no game version. The program displays a 2D graphical dashboard of a car. It links to most modern racing games using a localhost based UDP protocol and retrieves all the current statistics of the game such as speed, RPM, pedal position, car telemetry, and compiles it onto a visual dashboard. The no game version will boot up and render a dashboard without requiring any input, while the other version has a launcher for you to choose a dashboard and launches with the game.

Simulation Interface

Language: Python

Coupled with the last program, this program compiles data from modern racing games the same way, but this UI is oriented more towards applications in a physical simulation. Therefore, the data is positional data that could be translated for use in actuators that mimic the movement of the car.

First 3D Renderer in Python

Language: Python

Controls: WASD to move around and mouse to move the camera

This was my first attempt at rendering 3D points onto a 2D surface. I had several epiphanies during this time that led to a greater understanding of 3D graphics.

First Foray in 3D Graphics in C++

Language: C++

At last, I moved from Python into C++ to which I have remained to this day. After getting an understanding of basic 3D graphics concepts, I decided to move them over to C++. I learned to parse 3D object files and render them at the lowest level.

First 3D Engine

Language: C++
Controls: WASD to move around and mouse to move the camera

After my previous attempts in 3D graphics, I finally moved from the theory and into practice. This was the first time I stream-lined my OpenGL usage as well. This time I focused on memory management in terms of objects that needed to be rendered, as well as using polymorphism to expand the capabilities of the engine.

2D Poker Game

Language: C++
Controls: Space to move turn

After doing 3D graphics for a while, I took a small break and started focusing my efforts on understanding an efficient architecture for a graphics engine (which is arguably the most important part). Thus came a basic architecture I continue to use for smooth animations.

Differential Fields

Language: Python

My interests during this part of my life drifted from programming into mathematics, then back to programming when I found a mathematical application for some programming. I made a simple python script to render fields defined by a differential equation, and used the created vector field to simulate particles moving in a closed system.

3D Graphing

Language: C++
Controls: WASD to move around and mouse to move the camera

I had too much fun doing the 2D graphing that I extended it in the 3D realm with my then second iteration of my 3D rendering engine called Ansel. It allows you to make a 2D or 3D equation and graph it in a 3D environment. It allows you to make 3D shapes that are defined by equations.

Perceus

Language: C++

Perceus is my ongoing project that is now in its third iteration. It is aimed to be a 3D Game Engine, but at the moment serves its purpose as a 3D rendering engine. The main selling point is its completely api-abstract architecture. The engine doesn't have any popular rendering framework's code built in. Instead it has a layer of abstraction that allows you to implement your own rendering api.