



POLITECHNIKA
OPOLSKA

Opis pracy inżynierskiej

Rozproszony serwer renderujący dla oprogramowania blender

Autor:

Schyboll Robin

0. Krótki wstęp

W wstępie chciałem tylko zaznaczyć, że ten dokument to tylko wstępny koncept projektu, wiele jego elementów może się zmienić w późniejszym etapie tworzenia projektu.

W skrócie to po prostu zapisałem pomysł który siedział mi w głowie już od jakiegoś czasu.

1. Opis projektu

Projekt ten jest oprogramowaniem, pozwalającym na zdalne renderowanie projektów graficznych z oprogramowania blender.

Jego podstawową funkcjonalnością jest pobranie od użytkownika pliku z projektem do wyrenderowania, wybranie do której z maszyn renderujących ma on zostać wysłany, wysłanie go, oraz odebranie wyniku renderowania i wysłanie go z powrotem do użytkownika.

Projekt ten będzie się składał głównie z dwóch części, oprogramowania serwerowego, którego celem jest właśnie zarządzanie zadaniami. Oraz z tzw. robotnika, którego celem jest odbieranie projektów z serwera, renderowanie ich, oraz wysyłanie z powrotem wyniku.

Tego typu struktura pozwala na bardzo łatwe rozszerzanie aplikacji o nowe maszyny renderujące, w celu zwiększenia jej wydajności, oraz pozwala ona rozwinięcie później aplikacji w całkiem nowe rozwiązania jak np. czegoś w stylu sklepu w którym można wykupywać pracę maszyn do wyrenderowania swojego projektu, lub w drugą stronę, zaoferować maszynę do renderowania dla innych użytkowników za określoną kwotę.

2. Planowane technologie do wykorzystania

Backend – **Python** – **FastAPI** – **Turtle ORM** / Blender Python API (BPY)

*Opcjonalnie **Rust** do skomplikowanych obliczeniowo zadań

Frontend – **React** / **Redux**

Baza danych – **PostgreSQL**

3. Wymagania funkcjonalne

1. Możliwość konfiguracji parametrów renderowania, oraz konfiguracji samego projektu blendera poprzez stronę internetową.
2. Możliwość prostej instalacji oprogramowania robotniczego.
3. Możliwość zarządzania robotnikami z panelu administratora, dodawanie / usuwanie robotników, podglądanie ich historii zadań itp.
4. Możliwość włączenia kolejki zadań, czyli automatyczne rozdzielanie zadań pomiędzy robotnikami.
5. Możliwość podglądu zadań w kolejce, jeśli takowa jest włączona.
6. Zadania zostaną uznane za wykonane, dopiero po zaakceptowaniu wyniku przez użytkownika, oraz wpłaceniu odpowiedniej kwoty za usługę.
7. Przed akceptacją oraz wpłatą wynik zostanie pokazany użytkownikowi ze znakami wodnymi.

4. Wymagania niefunkcjonalne

1. Wszystkie pliki projektowe oraz wyniki renderowania zostaną przechowywane tylko w postaci szyfrowanej zarówno na serwerze jak i robotnikach.
2. Przesyłając pliki pomiędzy robotnikami a serwerem zostanie użyte szyfrowanie asynchroniczne, tak że wynik renderowania będzie mógł zostać zdeszyfrowany tylko na serwerze, a pliki projektu tylko na robotniku.
3. Każdy robotnik będzie posiadał unikalny klucz do szyfrowania plików, którego odpowiednik prywatny będzie przechowywany na serwerze.
4. Pliki projektów będą przechowywane na robotniku tylko w pamięci RAM, aby utrudnić ich przechwycenie.
5. Na serwerze pliki projektów będą przechowywane tylko do momentu konfiguracji danego zadania, po jego konfiguracji, pliki będą przechowywane tylko w postaci zaszyfrowanej.

5. Diagram relacyjny bazy danych

