

Zadatak je isprogramirati ray tracer u Pythonu.

U sklopu zadatka dodan je i funkcionalni pseudo kod ray trcera da se nebi gubilo vrijeme na proučavanje ray trcera jer to nije poanta zadatka nego naglasak je stavljen na Event-driven programiranje, parsiranje, pisanju Python ekstenzija, multithreading i ostalo. Program bi trebao imati jednostavni GUI u kojem će korisnik učitati scenu iz datoteke. Scena sadrži sve potrebne podatke za renderiranje a to su

parametri kamere (pozicija, smjer gledanja, udaljenost kamere), listu svjetala, listu geometrijskih objekata, rezoluciju slike, broj samplova po pikselu (antialiasing), broj threadova (dretvi). Format datoteke je proizvoljan (json, toml, txt...) Kad se učitava scena prije samog renderiranja u GUI bi se također trebale moći mijenjati neke stvari poput rezolucije slike, broj threadova, pozicije kamere... Nadalje GUI bi trebao sadržavati opcije Start/Pause/Restart. Za vrijeme renderiranja GUI bi trebao ostati responzivan tako da se renderiranje može u bilo kojem trenutku pauzirati ili prekinuti. Program bi također trebalo ubrzati koristeći paralelizaciju u Pythonu (multiprocessing) ili pisanjem Python ekstenzija u C++, Rust ili C.

GUI

- Event based arhitektura sucelja koristeći Tkinter
- Učitavanje scene iz datoteke.
- Modifikacija nekih osnovnih parametara (rezolucija slike, broj threadova, parametri kamere)
- Opcije Start/Pause/Restart
- Za vrijeme renderiranja GUI mora ostati responzivan (non-blocking)
- Vizualizacija progressa za vrijeme renderiranja – prikaz odrađenog dijela finalne slike
- Export renderirane slike

- Automatizirani testovi za GUI

Generalno

- Paralelizacija (multiprocessing ili low-level threads), svaki worker renderira dio slike npr. podijeljene na blokove
- Ubrzanje pisanjem ekstenzija u C++, Rust ili C

Izlaz demo koda :

