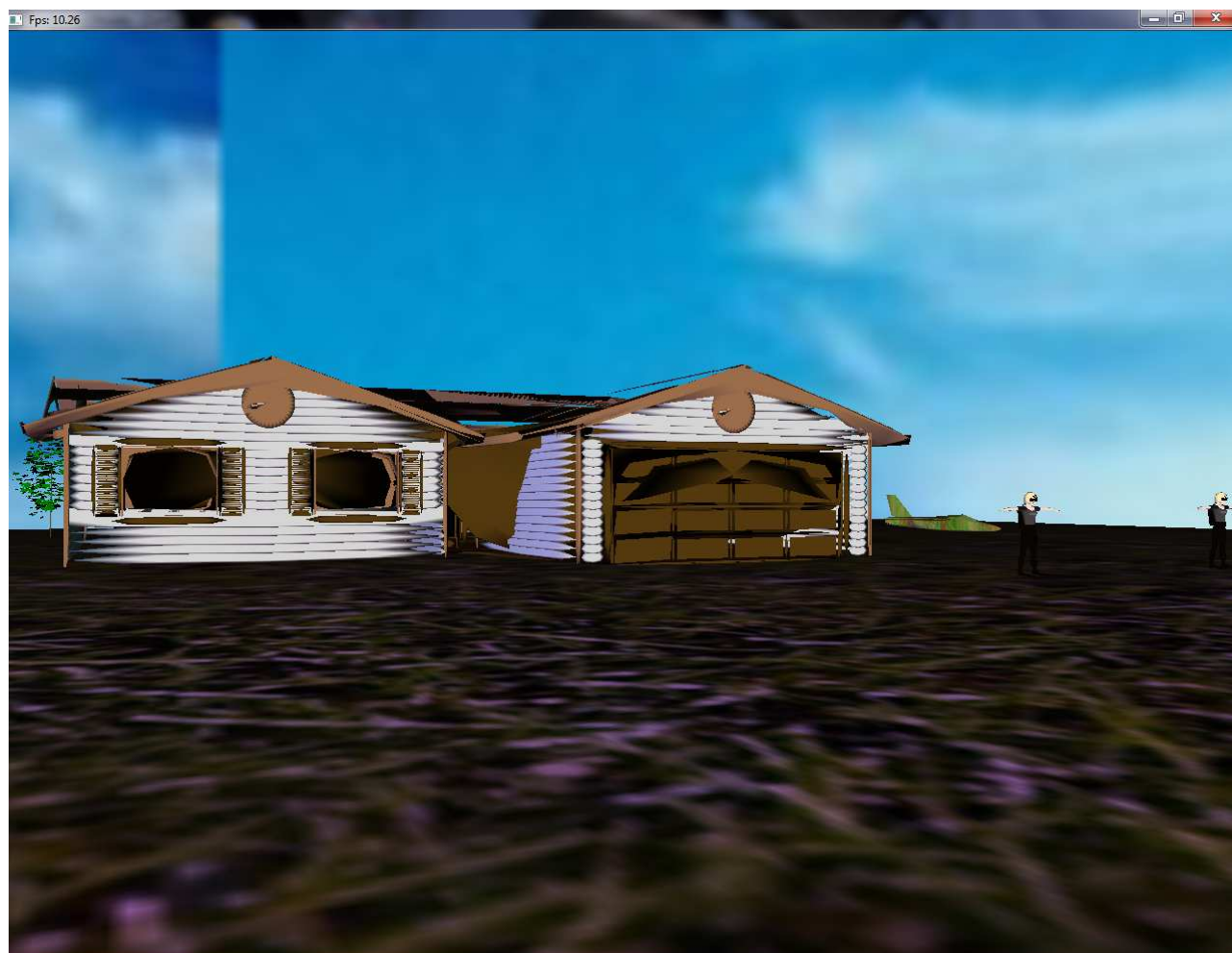


Virtualna Okruženja

Vježba 2

Ubrzavanje iscrtavanja

Scena u kojoj sam vršio mjerenja nalazi se na slici 1. Odabrao sam zahtjevnu scenu sa više različitih modela (stablo, kuća, Lara Croft i avion) kako bih bolje uočio razlike kod različitih metoda.



Slika 1. Testna scena

U nastavku su tablice sa izmjerenim brojem sličica u sekundi za pojedine opcije.

Tablica 1. Mjerenje bez LOD opcije

Frame rate	Bez upotrebe metoda za ubrzavanje	Teksture OFF	Odbacivanje stražnjih poligona ON	Odbacivanje po projekcionom volumenu ON	Broj svjetala		
					0	4	6
Scena male složenosti	44	44.3	44	62	44.2	44	44.1
Scena srednje složenosti	35	34.8	35	51.2	35.2	35	35
Scena velike složenosti	10.2	10.1	10.3	15.9	10.2	10.4	10.4

Tablica 2. Mjerenja sa LOD opcijom

Frame rate	Bez upotrebe metoda za ubrzavanje	Teksture OFF	Odbacivanje stražnjih poligona ON	Odbacivanje po projekcionom volumenu ON	Broj svjetala		
					0	4	6
Uključen LOD	29.3	28.9	29.4	36.5	28.7	29	29.1

Lako je primijetiti kako su jedine metode koje utječu na brzinu iscrtavanja odbacivanje po projekcionom volumenu i kontrola složenosti objekata. Ostale metode praktički ne utječu na brzinu crtanja jer ne mijenjaju broj vrhova koji ulaze u grafički cjevovod. **Zeleno** sam označio metode koje ubrzavaju, a **crveno** metode koje ne utječu na brzinu.

Promjena složenosti povećava broj sličica koje se iscrtavaju u sekundi jer naprosto šalje modele sačinjene od manje vrhova što znači manje posla u cjevovodu. LOD je interesantan jer omogućava balansiranje performansi i detaljnosti. Mali problem sa LOD-om je pri prelasku iz scene sa malo detalja u scenu sa više detalja.

Odbacivanje po projekcionom volumenu je odlična metoda ubrzavanja jer omogućava da se objekt koji nije u vidnom polju u potpunosti zanemari. Ubrzanje je 25% do 50% za ovu scenu. Ubrzanje ovisi o složenosti i broju odbačenih objekata.

Teksturiranje je postupak gdje se umjesto interpolacije boje vrhova, interpoliraju koordinate teksture i fragment se oboji čitanjem boje iz teksture. To za današnje sustave nije problem i u ovom slučaju ne predstavlja udar na performanse. U slučaju da se koristi postupak filtriranja tekstura ili da se koriste teksture veće razlučivosti moglo bi doći do pada performansi.

Odbacivanje stražnjih poligona je korisno kod modela sa iznimno puno poligona gdje se svaki poligon sjenča složenim programom za sjenčanje fragmenata. U ovom slučaju broj poligona je relativno malen i nema sjenčanja, tako da odbacivanje stražnjih poligona ne popravljia performanse. Usprkos tome, dobra je praksa uvijek odbacivati stražnje poligone.

Broj svjetala isto ne popravljia performanse, a to je vjerojatno jer sustav osvjetljenja u *openGL*-u nije toliko složen da desetak svjetla predstavlja udar na brzinu iscrtavanja. U projektu se koristi samo difuzno osvjetljenje, što predstavlja efektivno jedan skalarni produkt po dodanom svjetlu, a to nije tako zahtjevna operacija.

Primijetio sam kako se u projektu za svaki poligon objekta vrti petlja koja poziva svaki vrh i postavlja njegovu normalu, koordinate teksture/boju i poziciju. Za performanse bi puno bolje bilo *openGL*-u predavati pokazivače na polja vrhova, normala, koordinata i poligona. To je moguće korištenjem naredbi *gl*Pointer*, gdje * može biti *Vertex*, *Normal* i *TexCoord* te crtanje sa *glDrawRangeElements*.

Dodatni zadatak

Tablica 3. Utjecaj veličine prozora

Veličina prozora	<i>Frame rate</i>
Cijeli ekran	10.2
2/3 cijelog ekrana	10.4
1/3 cijelog ekrana	10.4

Promjena veličine prozora ne utječe na brzinu iscrtavanja jer ne smanjujemo složenost iscrtane scene. I dalje će program proći kroz sve (ili sve vidljive) vrhove, samo će rasterizator imati manje posla jer će sliku crtati na manje područje, što ne predstavlja promjenu na performanse u grafičkom cjevovodu. Smanjenje veličine prozora utjecalo bi na performanse kada bi program koristio složene izračune u programima za sjenčanje ili kada bi radio *antialiasing*. Način na koji bi mogli popravljati performanse za smanjenje veličine prozora je da proporcionalno mijenjamo i LOD jer na manjoj slici detalji modela ne znaće previše.