FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Računalna animacija

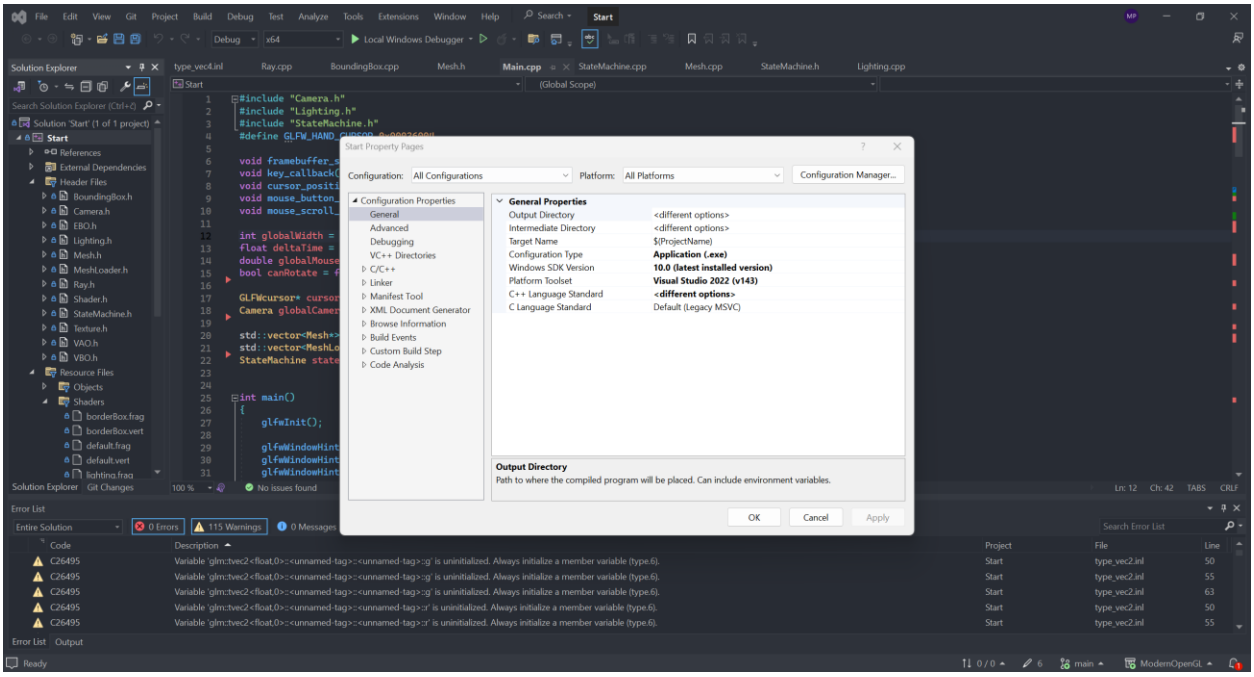
3. laboratorijska vježba

Marko Prosenjak

Zagreb, siječanj 2024.

1 Upute za pokretanje

Potrebno je klonirati repozitorij lokalno na računalo. Nakon toga je potrebno konfigurirati Visual Studio prema uputama sa sljedećeg linka: https://www.youtube.com/watch?v=XpBGwZNyUh0&list=PLPaoO-vpZnumdcb4tZc4x5Q-v7CkrQ6M-&index=1 (radi jednostavnosti, pošto ima dosta koraka, naveden je link umjesto opisa). U LAB3/Start/Libraries direktoriju postoje potrebni dokumenti i direktoriji za pokretanje vježbe, potrebno je samo u postavkama Visual Studio-a (desni klik na ime projekta u Solution Exploreru -> Properties) navesti te direktorije i dokumente u odgovarajućim poljima kao što je objašnjeno u videu. Nakon što je okolina konfigurirana, potrebno je pritisnuti na zeleni Play gumb kako bi se projekt buildao i program pokrenuo. Kamera se u prostoru kreće uz pomoć strelica na tipkovnici. Kamera se rotira uz pomoć pomicanja miša dok je pritisnut desni klik miša. Pritiskom tipke A, prelazi se u ADD stanje (stanje dodavanja objekata). Moguće je učitati različite objekte, te se njihov odabir vrši pritiskom na neku od tipki od 1 do 8 (nakon što je pritisnuto na tipku A). Ukoliko se korisnik nalazi u ADD stanju, objekt će se instancirati na mjestu koje je udaljeno 10 jedinica od pozicije klika miša (lijevi klik). Pritiskom tipke G, prelazi se u GRAB stanje (služi pomicanju objekta po sceni), pritiskom tipke S prelazi se u SCALE stanje, a pritiskom tipke DELETE prelazi se u stanje brisanja. Kako bi ova 3 stanja imala učinak, objekt treba biti označen prije pritiska na tipke G, S, DELETE. S tipkom ESC se program terminira.



Slika 1 Properties prozor za podešavanje okoline projekta

2 OPIS VJEŽBE

Cilj vježbe je bio implementiranje osnovnih funkcionalnosti koje posjeduju svi grafički alati (poput Blendera), te softwera za razvoj računalnih igara (eng. Game engine), kao što su: označanje (eng. Select) objekta u sceni uz pomoć klika, translacija te skaliranje objekta u sceni (prati se pozicija miša). Također su implementirane funkcionalnosti dodavanja i brisanja objekata iz scene. Dodatno je implementirano Phongovo sjenčanje, te transformacije kamere (rotacija kamere, pomicanje kamere u sceni, uvećavanje

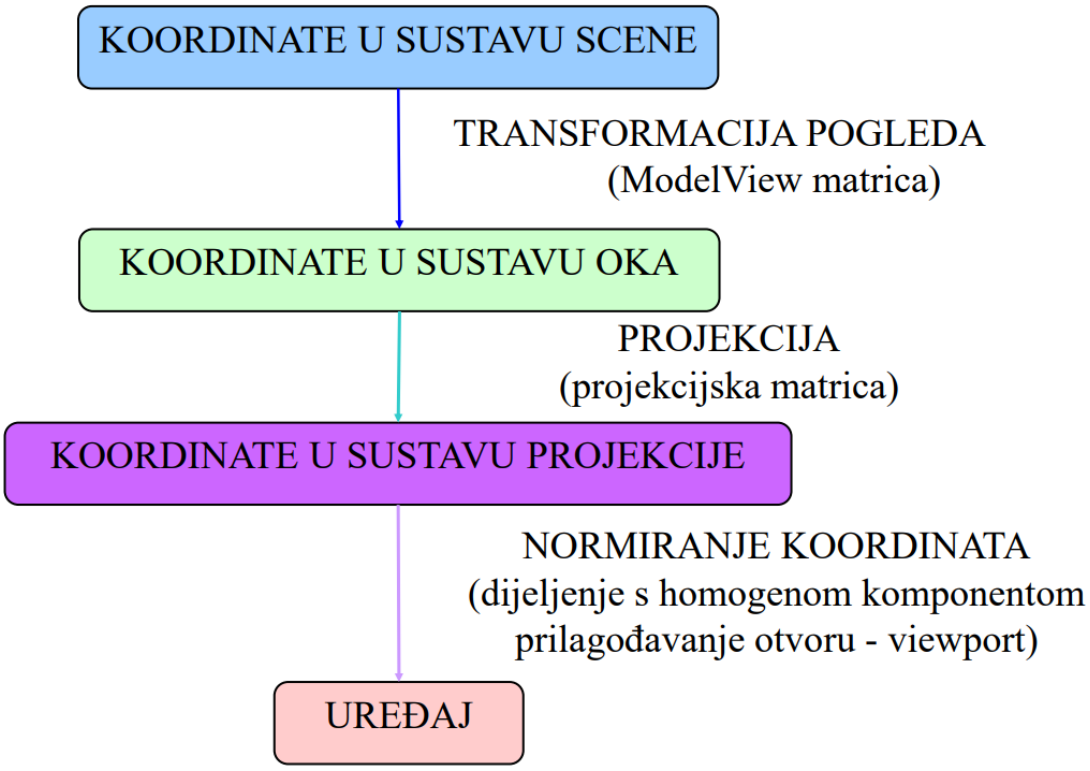
(eng. Zoom in) i udaljavanje (eng. Zoom out)). Program je pisan u jeziku C++, te su korištene vanjske biblioteke GLFW i GLAD. Korišten je moderni OpenGL verzije 3.3.

3 Implementacija

StateMachine.cpp služi za kontroliranje stanja ovisno o pritisku tipke. U trenutku kada je na tipkovnici pritisnuta neka tipka, poziva se funkcija ChangeState kako bi se odredilo stanje u kojem se korisnik treba nalaziti. StateMachine.cpp također ima funkciju za praćenje akcija na mišu (klik miša, otpuštanje miša) te poduzima definirane akcije. Svrha ove klase je izvršavanje različitih akcija, te prelazak u nova stanja ovisno o korisničkom unosu (eng. Input).

Transformacija klika u koordinate svijeta:

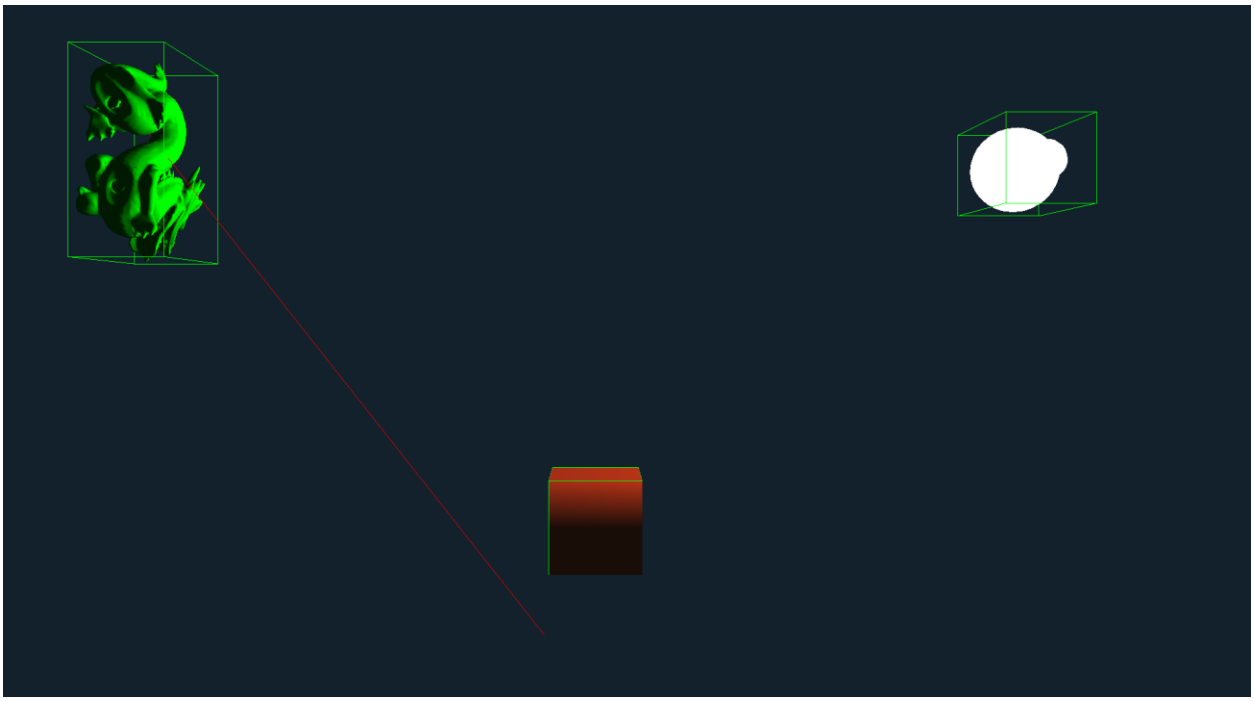
S obzirom na to da su koordinate klika zapravo koordinate piksela, koji su 2D, potrebno ih je inverznim transformacijama pretvoriti u koordinate u svijetu (u kojem se scena i svi objekti u njoj nalaze). To se obavlja u ScreenToWorldCoordinates funkciji unutar Camera.cpp.



Slika 2 Prikaz procesa pretvorbe točaka iz 3D (svijet) u 2D (monitor) (preuzeto s: https://www.zemris.fer.hr/predmeti/irg/predavanja/3_primitive.pdf)

Kako bi se iz koordinata uređaja došlo do koordinata svijeta, potrebno je najprije normalizirati koordinatu klika, te zatim primijeniti inverznu matricu projekcije, pa inverznu matricu pogleda te zatim podijeliti dobiveni rezultat s homogenom koordinatom (krećemo se unazad po pipelienu kako bi iz 2D došli u 3D).

Za određivanje presjeka između objekta i klika miša koriste se granični okviri (eng. Bounding box) objekata, te zraka, koja je dobivena iz koordinata klika u svijetu (eng. World-space coordinate). Provjerava se je li trenutna točka pravca na kojem leži zraka manja od maksimalnih x,y,z komponenti graničnog okvira i veća od minimalnih x,y,z koordinati graničnog okvira (duljina zrake se iterativno povećava: rayStart + rayDirection * i). Ukoliko je, znači da se nalazi unutar graničnog okvira objekta, tj. da je korisnik kliknuo na objekt, te se taj objekt postavi kao aktivan te se oboja u zeleno. Ta se provjera vrši nad svim objektima u sceni u trenutku kada korisnik klikne lijevom tipkom miša bilo gdje u sceni (osim u slučaju kada je korisnik u stanju ADD (tada lijevi klik označava dodavanje objekta), ili se nalazi u nekom od stanja GRAB ili SCALE jer tada klik označava kraj primjene transformacije).

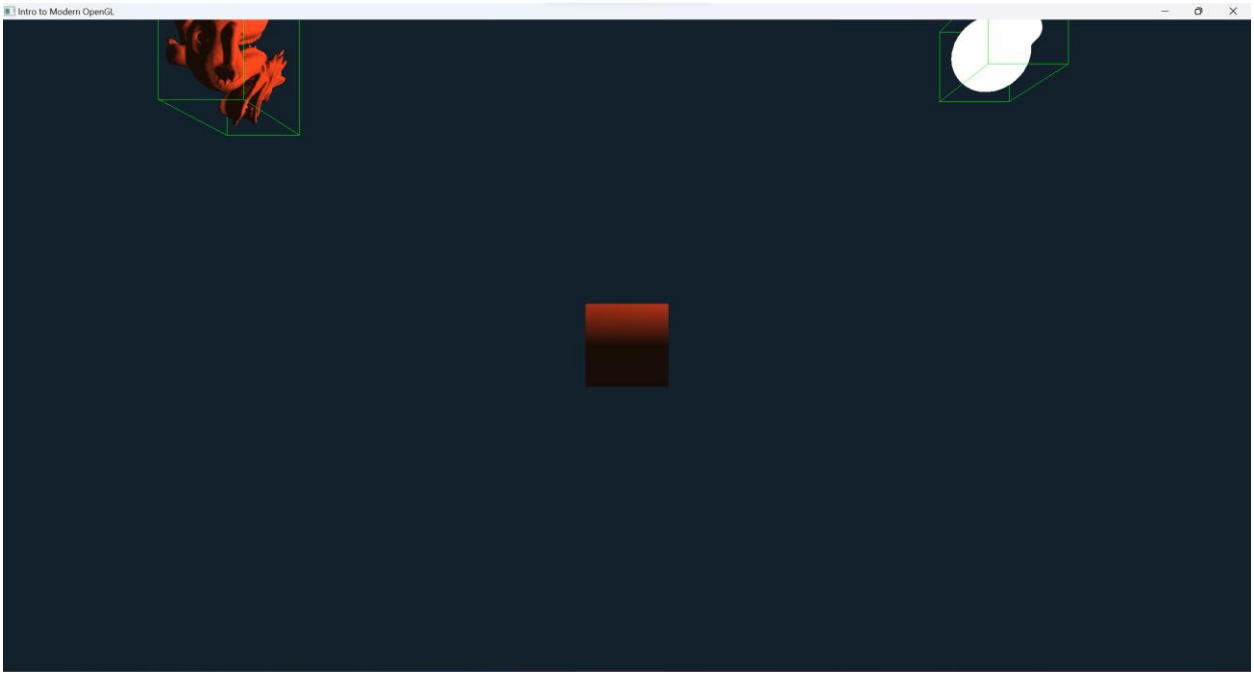


Slika 3 Prikaz presjeka zrake (klik miša u svijetu) s graničnim okvirom objekta

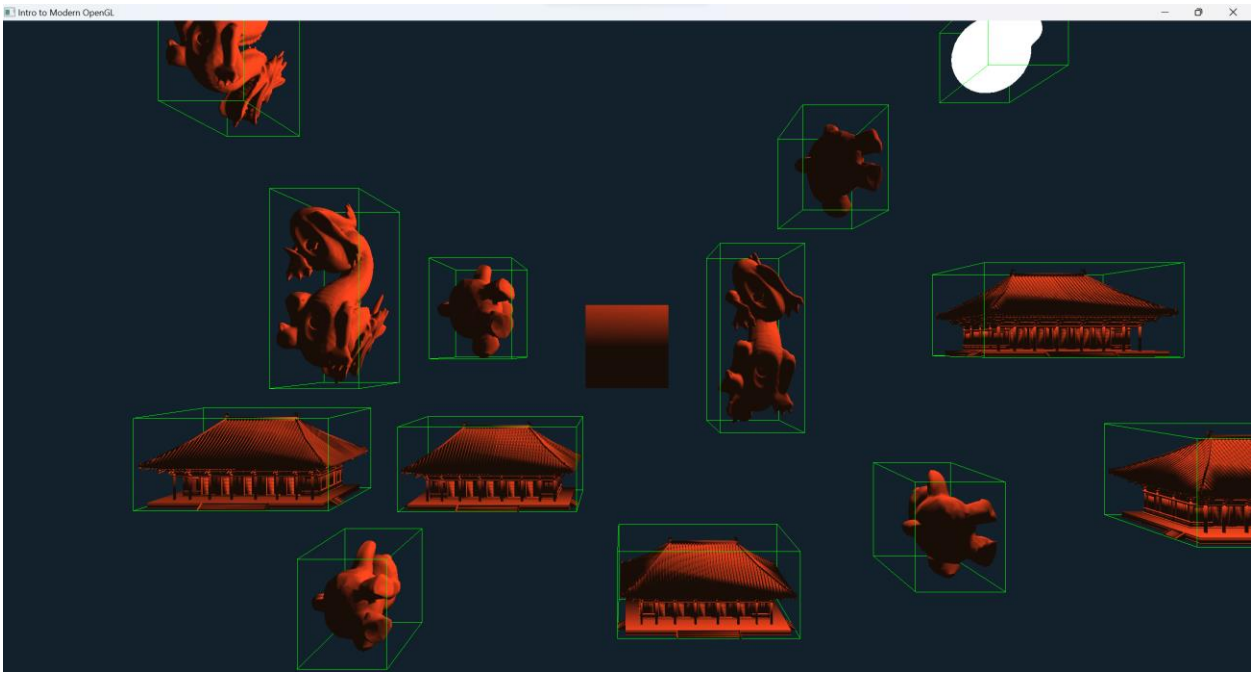
Translacijski vektor (vektor za koji se objekti pomiču u sceni kada se korisnik nalazi u GRAB stanju) se nalazi u ravnini koja je uvijek orijentirana prema korisniku, odnosno prema kameri. To je postignuto tako da je normala te zamišljene ravnine jednaka vektoru suprotnom od smjera u kojem kamera gleda. Ta ravnina je definirana kako objekt prilikom transformacija ne bi nestajao (tj. pobjegao) iz vidljivog dijela ekrana u slučaju da je kamera rotirana (ovako se konstanto kreće po površini koja je orijentirana prema kameri). Kada se korisnik nalazi u GRAB ili SCALE stanju, prati se pozicija miša, te su uz pomoć ScreenToWorldCoordinates funkcije izračuna pozicija miša u svijetu, te njegov smjer (ide od pozicije

kamere u svijetu prema poziciji miša u svijetu). Te vrijednosti se koriste dalje za dobivanje zrake koja sjeće ravninu, što će biti objašnjeno u nastavku.

Jednadžba ravnine koja gleda prema kameri je Ax_p + By_p + Cz_p + D = 0, pri čemu su A, B i C zapravo x, y i z komponente normale ravnine, tj. komponente od vektora -cameraDirection, dok su x_p, y_p i z_p koordinate neke točke koja se nalazi u toj ravnini. Ta točka je pozicija objekta na koji je korisnik kliknuo. Ovime je moguće izračunati vrijednost jedine nepoznanice, tj. vrijednost od D. Sada je moguće uz pomoć presjeka ravnine i zrake koja je ispaljena iz miša, otkriti gdje ta zraka sjeće ravninu. To se odredi tako da se napravi presjek ravnine Ax + By + Cz + D i pravca rayStart + rayDirection * t, te se dobije vrijednost parametra t, iz kojeg je vidljivo gdje se u svijetu nalazi točka presjeka, nakon što se parametar t uvrsti u jednadžbu zrake. Sada je moguće izračunati translacijski vektor, koji je jednak točka_presjeka – pozicija_aktivnog_objekta. To je vrijednost za koju je potrebno pomaknuti objekt u slučaju da se korisnik nalazi u GRAB stanju (vektor se nalazi u istoj ravnini u kojoj se nalazi i aktivni objekt). S obzirom na to da se prilikom svakog pomicanja objekta u sceni izrađuje nova matrica translacije, kao argument se predaje točka sjecišta zrake s ravninom. Kod skaliranja se koristi duljina translacijskog vektora kao faktor kojim se objekt skalira. Svi objekti koji se iscrtavaju, nalaze se u vektoru (tip strukture std::vector) koji se proširuje s novim objektima kada ih korisnik doda (pri čemu mora biti u stanju ADD), te se smanjuje kada korisnik briše objekte iz scene (mora se nalaziti u stanju DELETE). Ta struktura predstavlja sve objekte koji se nalaze u sceni, te se nad svakim njenim članom provodi provjera presjeka sa zrakom ispaljenom iz miša prilikom lijevog klika miša.



Slika 4 Početna scena



Slika 5 Scena nakon dodavanja objekata