

Projektni zadatak iz predmeta LPRS 2

Marko Đorđević, Radomir Zlatković i Aleksa Heler

Fakultet tehničkih nauka

I. Uvod

Ovaj projekat je realizovan u C jeziku pomoću emulatora. U emulatoru postoji pristup memoriji koja se ispisuje na ekran i implementiran je osnovni joypad. Ovo se koristi kao ulaz i izlaz za igru.

Kako bi igru bilo moguće kompajlovati za razne platforme izbegli smo korišćenje matematičkih biblioteka, već smo koristili naše funkcije sinusa i kosinusa kao i nekih drugih operacija. Takođe smo iz istog razloga koristili *fixed point* aritmetiku u celom programu.

II. IGRA

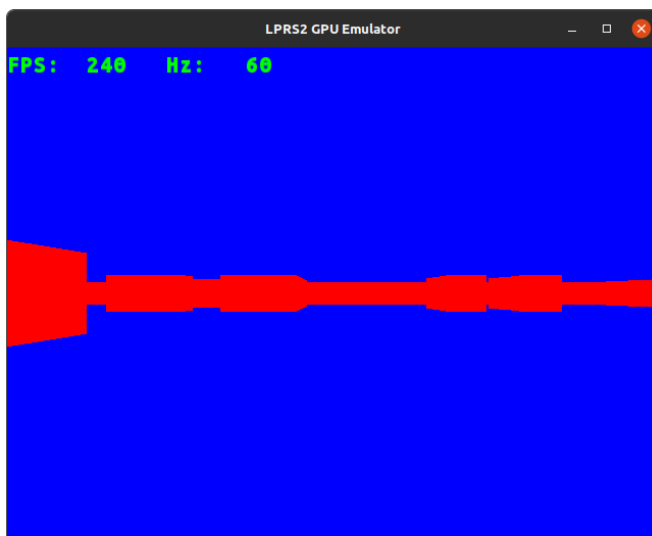
Kao svaki dobar projekat za novu platformu, odlučili smo da napravimo Doom igru za ovaj emulator. Igra ima samo jedan nivo, u kojem se može pobediti tako što se ubije svih 5 neprijatelja. Kako bismo otežali taj izazov, municija je ograničena na 20 metaka, dok je potrebno 3 metka po neprijatelju kako bi ih ubio. Ukoliko neprijatelji ubiju igrača, izgubio je. Rezultat se računa tako što svaki put kada igrač ubije neprijatelja, šteta koju taj neprijatelj nanosi se dodaje na rezultat. Takođe u slučaju pobeđe se preostali HP igrača dodaje na rezultat, što u našem slučaju znači maksimalno 180 poena.

III. NAČIN REALIZACIJE RENDERERA

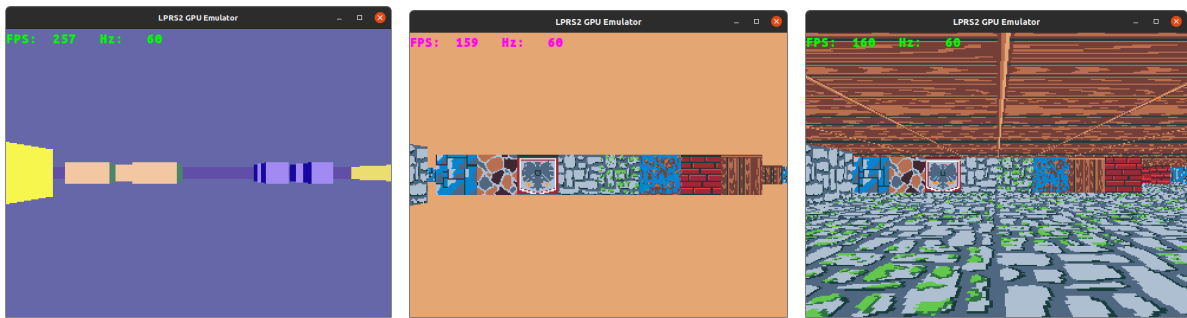
Ideja je bila napraviti igricu za emulator koja je slična *Doom* ili *Wolfenstein 3D* igri. Kako bi mogli napraviti 3D igru u 2D emulatoru, koristimo Raycasting algoritam (ne raytracing). Kako bi pojednostavili sebi problem, i kako nije neophodno imati vertikalnost u igri (spratovi, stepenice, skokovi), možemo mapu predstaviti 2D matricom celih brojeva, gde broj označava tip zida, i ako je 0 znači prazno polje po kojem se igrač može kretati.

Kako bismo dobili 3D efekat, potrebno je zidove nacrtati tako da im je visina obrnuto proporcionalna udaljenosti od kamere. To radimo tako što crtamo jednu po jednu vertikalnu liniju. Za svaku vertikalnu liniju bacimo zrak u tom smeru i gledamo kad ce udariti u prvi zid i time dobijemo udaljenost, i za to koristimo relativno jednostavan "**Digital Differential Analysis**" (DDA) algoritam.

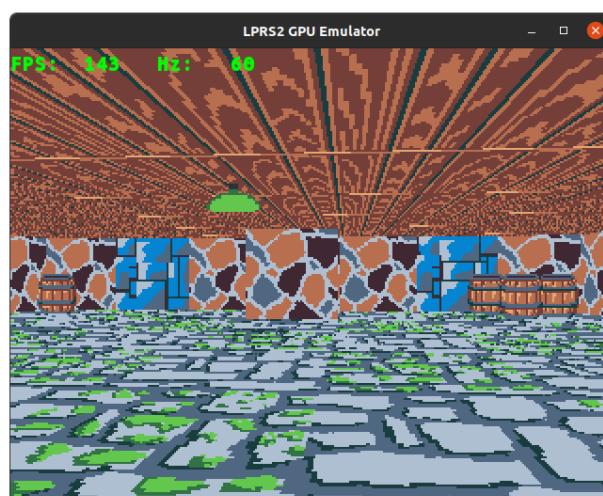
DDA je relativno brz i koristi se za pretragu koje kvadrate zrak (ray) pogađa. Mi ga koristimo da nađemo koji kvadrat naše mape je zrak pogodio i zaustavimo algoritam u tom trenutku i nacrtamo vertikalnu liniju čija visina odgovara udaljenosti zida tj. dužini zraka.



Na slici iznad se može videti kako ovaj efekat izgleda koristeći samo dve boje. Mapa se čuva u dvodimenzionalnom nizu tako da svaki element predstavlja jedan blok na mapi, i vrednost elements predstavlja koji tip zida se tu nalazi, ili je samo prazan prostor po kojem se može šetati.



Dodatnim iteracijama, dodavanjem više boja, zatim tekstura na zidove, i na kraju plafon/pod i billboard sprajtova, dobili smo *game renderer* koji izgleda sasvim lepo:



To je predstavljalo kraj naše portage za idealnim rendererom, što znači prelazak na igrača, neprijatelje i ostalu mehaniku igre.

IV. NAČIN REALIZACIJE MEHANIKE IGRE

Igrač sadrži informacije o poziciji i rotaciji, koji se koriste prilikom renderovanja da iscrtaju odgovarajuću perspektivu. Tokom renderovanja kada se crta sprajt koji se nalazi na X koordinati koja je sredina ekrana, zabeleži se koji je to sprajt. Zatim se prilikom pucanja igrača proverava da li je taj sprajt neprijatelj, i ako jeste nanese se šteta tom neprijatelju.

Neprijatelji imaju veoma jednostavnu logiku održavanja razdaljine. Ako postoji neprekidna linija od njih do igrača i ako je igrač unutar određenog dometa, krenuće ka njemu, i ići sve dok ne stignu do određene idealne razdaljine, takođe ako igrač priđe previše blizu neprijatelju oni će bežati unazad dok ne dodju do zida. Ako je igrač u dometu pucaće na njega određenom frekvencijom.

V. ZAKLJUČAK

Tokom izrade ovog projekta smo naučili i usavršili mnoge veštine, između ostalog rukovanje bitima i bitske operacije, kao i osnove programiranja igrica i *game development*-a. Bilo je izazova oko renderovanja i rada sa slikama, ali na kraju smo savladali sve prepreke i napravili jedan nivo koji je bilo lepo igrati, sa svim osnovnim mehanikama koje su potrebne za igru.

Na slici ispod se može videti kako izgleda konačna igra:

