**Projektni zadatak iz predmeta LPRS2**

**Sega GPU – Galaga Wars**

**Zadatak**

Zadatak projekta je implementacija popularne Sega igrice, Galaga Wars, u kojoj pomerate svemirski brod i pokusavata da upucate sve muve. Potrebno je prilagoditi hardver i napisati softver.

**Realizacija grafičke periferije procesora**

U projektu Battle City realizovana je arhitektura koja se sastoji od *MicroBlaze* procesora, njegove *BRAM* memorije i grafičkog procesora. Zadatak periferija grafičkog procesora je da na osnovu ulaznih koordinata generiše rezultat u vidu boje tačke, definisane koordinatama (*RGB* vrednost boje, gde svaka komponenta ima 8 bita). Rad periferije koji se odvijao u četiri stejdža nije uspio da zadovolji vremenske kriterijume, te je zbog toga raspračan u više stejdževa koji dosta ubrzavaju sam process. Period čekanja koji se odonosi na čitanje novih vrijednosti signala sveden je na minimum. Proces je organizovan u 15 stejdževa. Postoje četiri faze u kojima se generiše adresa kojom se vrši pristup memoriji. Izlazi iz faza su povezani na multipleker koji iz trenutne faze čita vrijednost adrese i to upisuje u memoriju. Ulazni podatci se menjaju na svaka četiri takta.

Izlazi po fazama:

Faza 0 - Boja piksela koja treba da se iscrta, odredjena sortiranjem po Z osi i alfa vrednostima

Faza 1 - Adresa indeksiranog elementa mape

Faza 2 - Boja piksela pokretne slike (sprajta)

Faza 3 - Boja texela statičke slike

Modifikovana je veličina statičke slike, tj. omogućeno je pored iscrtavnja slika dimenzije 8x8 i iscrtavanja onih sa dimenzijom 16x16.

**Realizacija softvera**

Realizacija softvera je zapocela, skidanjem neophodnih slika sa internet, koje su dimenzije 16x16 piksela.Nakon toga je generisan mem.txt fajl koji je kopiran u ram, a nasa mapa je zamenjan sa crnim slikama.

Na pocetku battle\_city.c fajla, definisani su svi nephodni registri, globalne promenljive, te karakteri.Sam program pocinje od battle\_city funkcije, u kojoj je na pocetku napunjen niz muva, generisan brodic, a zatim se ulazi u while petlju, koja se izvrsava sve dok broj zivota ne padne na nula.

Unutar while petlje pozivaju se funkcije muva\_leti, koja crta muve po ekranu, te menja sprajtove(tako da imamo iluziju da muve sinhronizovano skupljaju I sire krila).

Zatim se proverava da li je korisnik stisnuo neki od tastera, i u zavisnosti da je stisnut taster za levo ili desno , poziva se funkcija koja menja poziciju naseg brodica, a u slucaju da je stisnut taster za gore, nas brodic puca.

Posle se nadju 3 random broja, za svaki niz po jedan, koji odredjuje koji red muva puca, te se pozivaju funkcije za pucanje muva. U toj funkciji proverava se da li nas je metak od neke od muva pogodio, te u slucaju da jeste, ulazimo u funkciju destroy u kojoj proveravamo da li je unisten brodic, te ako jeste smanjujemo broj zivota za jedan.

U slucaju da nas brod puca, ulazimo u funkciju battle\_shoot, u kojoj proveravamo da li je nas brod pogodio neku od muva, te ako jeste ulazimo u funkciju destroy u kojoj zamenimo tu muvu sa crnim sprajtom, a ako je pogodjena muva iz prvog reda, zamenimo je sa istom muvom, al plave boje(jer gornje muve imaju 2 zivota).

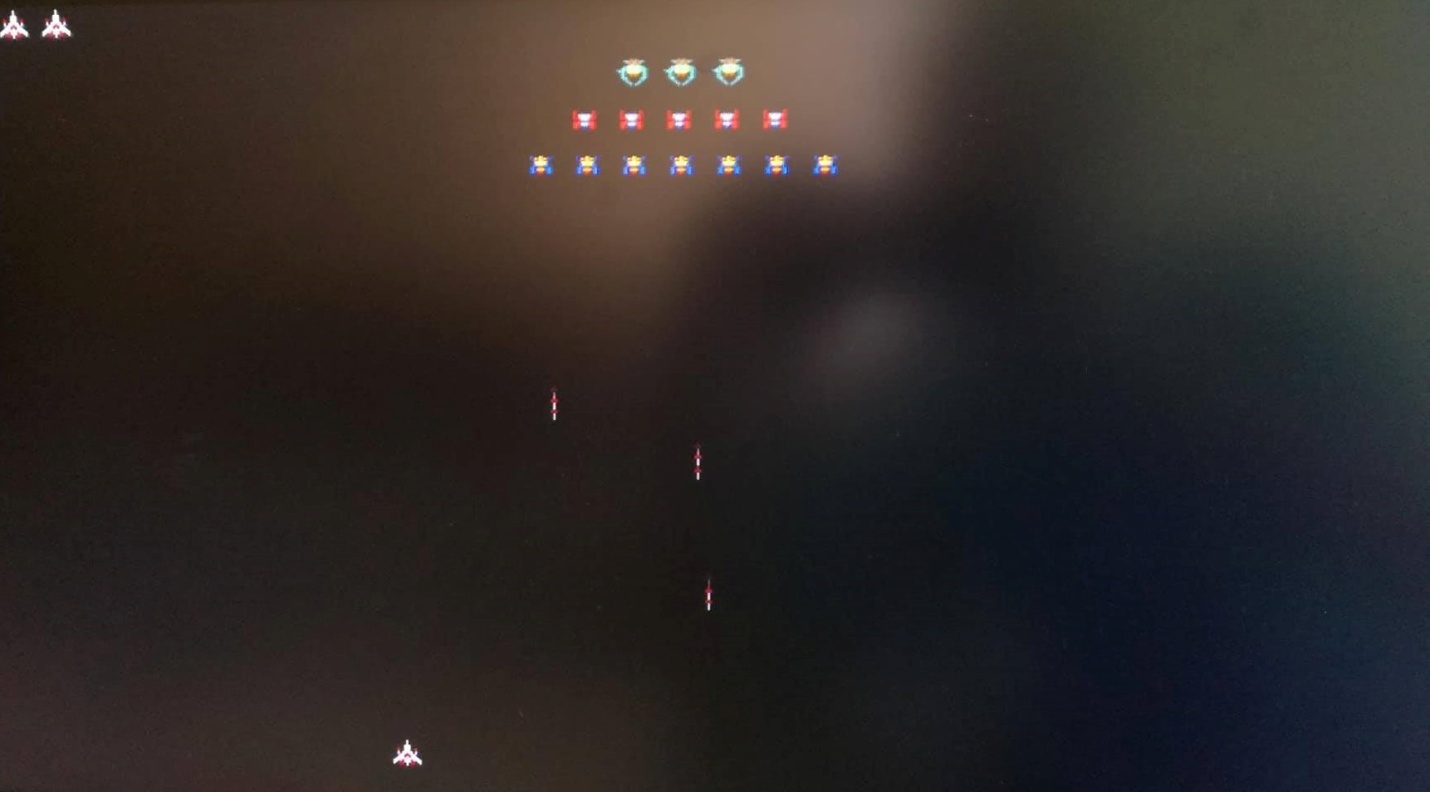
Posle svega u while petlji se poziva funkcija promeni\_polozaj koji svaki red muva pomera levo do granice ekrana pa onda desno.

Na kraju se proverava koliko imamo zivota, te se crta toliko brodica u gornjem levom uglu, koji vizuelno podsecaju korisnika koliko mu je zivota ostalo.

Nakon izlaska iz while petlje potrebno je bilo obojiti ceo ekran u crno, te ispisati poruku da je igra zavrsena, ali zbog specificne strukture hardvera to nije ucinjeno.

|  |  |
| --- | --- |
| Definicija funkcija | opis funkcije |
| rand\_lfsr113( void ) | Generator slučajnih brojeva |
| chhar\_spawn( characters \* chhar ) | Stvaranje sprajta na mapi |
| map\_update( map\_entry\_t \* map ) | Ponovno iscrtavanje izmenjenih delova mape |
| map\_reset( map\_entry\_t \* map ) | Vraćanje mape na prvobitan izgled |
| void muva\_leti(characters muva[], int broj, int vrsta) | Muve sinhronizovani sire i skupljaju krila, u ovoj funkciji se muve crtaju po ekranu |
| mario\_move( map\_entry\_t \* map, characters \* tank, direction\_t dir ) | Pokretanje Marija u datom pravcu |
| static void destroy( characters \* coin,unsigned int x, unsigned int ya) | Unistavanje sprajta sa ekrana, bilo da je muva,brodic ili metak |
| void promeni\_polozaj(characters muva[], int broj) | Pomeranje muva po ekranu |
| void muva\_shoot(characters \*muva, characters \* metak\_muva, characters \* brod, int \*y) | Funkcija u kojoj se slucajno biraju 3 muve, od preostalih zivih koje ce ispucati metak.Takodje se proverava da li je neka od njih pogodila brod |
| battle\_city( void ) | Glavna funkcija u kojoj se radi sva obrada potrebna za igru |
| void battle\_shoot(characters \* brod , characters \* metak) | Funkcija u kojoj brod puca, takodje se proverava da li smo pogodili neku od muva |
| void fill\_muve(void) | Funkcija u kojoj se popunjava prvi red muva |
| void fill\_muve2(void) | Funkcija u kojoj se popunjava drugi red muva |
| void fill\_muve3(void) | Funkcija u kojoj se popunjava treci red muva |

**Table 2. Definicija funkcija realizovanih u fajlu battle\_city.c**

****

**Slika 1. Iscrtavanje igre na grafičkom uredjaju**