**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

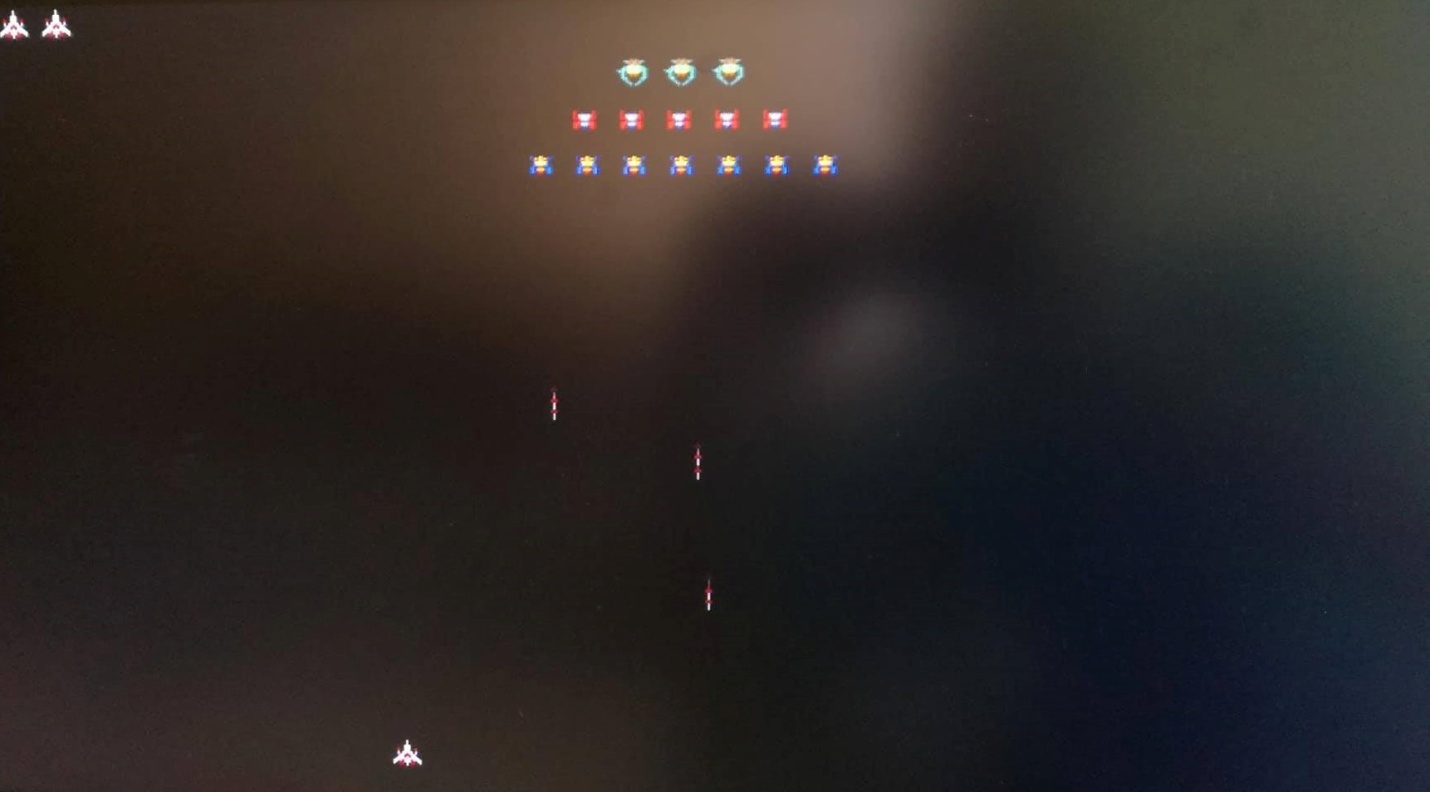
Modifikovana je veličina statičke slike, tj. omogućeno je pored iscrtavnja slika dimenzije 8x8 i iscrtavanja onih sa dimenzijom 16x16.

**Realizacija softvera**

U fajlu map.h nalazi se mapa generisana na osnovu adresa sličica (bitmapa) koje su smeštene u projekat. Dimenzija statičkih slika i sprajtova je 16x16 piksela. Sam softver je modifikovan i prilagodjen novim slikama i logistici same igre. Uradjena je potpuna funkcionalnost igrice, neprijatelji sinhronizovano lete, masu krilima i pucaju. Ostalo je da se urade animacije dolaska muva na svoje pozicije , da se naprave muve koje menjaju polozaj (padaju sa mape i bacaju bombe) i da se u hardveru omoguci ispis stringa, gde bi se ispisala poruka o pobedi, o porazu kao i o broju osvojenih bodova.

|  |  |
| --- | --- |
| 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**