|  |  |  |
| --- | --- | --- |
|  | UNIVERZITET U NOVOM SADU  **FAKULTET TEHNIČKIH NAUKA U**  **NOVOM SADU**  **ODSEK ZA RAČUNARSKU TEHNIKU I RAČUNARSKE KOMUNIKACIJE** |  |

Milan Pašić RA 89/2016,

Mihajlo Solda RА 152/2013,

Nemanja Havlović RА 207/2013,

**Šah**

Predmetni Projekat

- LPRS 2 -

**Zadatak**

Zadatak je bio realizovati brutalnu simulaciju penzionerskog šaha.

**Rešenje**

Kao osnovu koja je omogućila hardversku podršku, koristili smo jedan od prošlogodišnjih projekata, Minesweeper.

Implementaciju smo podelili na dva dela:

1. Grafika

2. Logika igre

**Grafika**

Mega-mapa je realizovana u GIMP-u, dimenzije 180x120 piksela (24 kvadrata 30x30). Korišćen je RGB565 format radi uštede memorije (30% uštede u odnosu na RGB888). Figurice skinute sa interneta, dodatno obrađene sa GIMP-om radi pobošljanja vizuelnog ugođaja.

Po završetku mega-mape ona se eksportuje u C fajl, odakle joj je moguće programski pristupiti.

**Logika igre**

U daljem objašnjenju ćemo se referencirati na izvorni kod.

POINT – struktura koja opisuje kordinate X i Y

PIECE – struktura figura

SQUARE – pojedinačna polja na tabli

PIECE white[16] – figure belog igrača (pozicija i stanje)

PIECE black[16] – figure crnog igrača (pozicija i stanje)

SQUARE board[8][8] - matrica table sa pokazivačem na figuru na svom polju

POINT playable[27] - niz koji se računa nakon selekcije figure i predstavlja sve pozicije na koje ta figura može da se pomeri

Zadatak se može podeliti u dve celine kako mu se i pristupilo pri rešavanju.

Najpre je rešen hardverski deo zadatka koji obuhvata smanjenje rezolucije adresiranjem signala koji određuju rezoluciju tako da su sada 9-obitni (množenje signala horizontalne komponente adresirane da obuhvata bite od prvog do desetog sa signalom koji predstavlja horizontalnu rezoluciju takođe adresiranim da obuhvata bite od prvog do desetog, a potom sabiranje sa signalom vertikalne komponente adresiranim na isti način). Ovi signali definisani u okviru modula vga top, gde su određeni pomoćni signali takođe smanjeni na odgovarajuće vrednosti i ovim je postignuto smanjenje rezolucije. Sledeći korak pri rešavanju zadataka je prilagođenje boja traženoj rezoluciji, što je odrađeno u okviru modula vga sync. Boje su predstavljene sa tri osmobitna signala (R-red, G-green, B-blue) koja su indexirana na odgovarajuće načine u zavisnosti od vrednosti signala (da li su postavljeni na visoku ili nisku vrednost ) aktivnog pixela, okvira, kao i režima rada. U okviru ovog modula svi signali povezani su na izlaz. Pri rešavanju ovog dela zadatka korišćen je VHDL i Xilinx-ovo okruženje ISE.

Radi demonstracije uspešno realizovanog prethodnog dela zadatka bilo je potrebno implementirati algoritam neke igrice po izboru. Odabrana igrica je popularni “Minolovac” ( engl. “Minesweeper”). Ovo je logička igra za jednog igrača gde se na početku pred korisnika postavlja tabela neotvorenih polja koja on otvara i na osnovu brojeva na otvorenim poljima zaključuje gde se nalaze “mine” i na ta polja je potrebno da stavi “zastavice”. Igra je gotova ukoliko igrač na sva polja gde su skrivene mine postavi zastavice ili ukoliko otvori polje na kom se nalazi mina.

Pred igrača se postavlja tabela neotvorenih pola dimenzija 9x9 koja skriva matricu istih dimenzija koja se generiše po slučajnom principu svaki put po pokretanju programa ili pritiskom na dugme “Reset”. Otvaranje polja se ostvaruje pozicioniranjem na željeno polje uz pomoć tastera na platformi (gore, dole, levo i desno) i pritiskom na centralni taster. Kretanje je ostvareno tako što su registrovane binarne vrednosti svih prekidača u aktivnom stanju i vrednost kada ništa nije pritisnuto, zatim upoređivanjem svake od ovih vrednosti sa vrednošću kada ništa nije pritisnuto i primenjivanjem binarne operacije “I” nad njima pri čemu izdvajamo željeni bit koji upoređujemo sa nulom i u zavisnosti od njegove vrednosti definišemo smer i pravac kretanja. Prvo otvoreno polje nikada ne može biti mina, već broj koji označava koliko mina to otvoreno polje dodiruje, prazno polje ili više polja što se dešava u slučaju da polje koje želimo da otvorimo dodiruje više praznih polja pri čemu se sva ova polja otvaraju - što je ostvareno u posebnoj funkciji koja najpre registruje sva prazna polja u čitavoj mapi, izdvaja ona koja dodiruju to označeno polje i otvara ih istovremeno sa njim . Postavljanje zastavica na polja se vrši pozicioniranjem na to polje na prethodno opisan način i okidanjem prvog prekidača (SW0), a skidanje zastavica se vrši na isti način, ali upotrebom drugog prekidača (SW1). Ove promene – postavljanje i skidanje zastavice mogu se ispratiti na brojaču zastavica koji se nalazi iznad same tabele i koji na početku pokazuje raspoloživi broj zastavica koji je jednak broju skrivenih mina. Takođe, svakim postavljanjem zastavice na ispravno mesto, tj. na mesto gde se zapravo nalazi mina uvećava se brojač koji je naravno skriven od korisnika i koji kada se izjednači sa brojem mina označava kraj igre (pobeda) što korisnik vidi kroz ikonicu smiley-ja (koji sada nosi naočare! ). Do kraja igre takođe dolazi i ukoliko igrač otvori minu koja eksplodira i na mapi mu se prikažu ispavno i neispravno postavljene zastavice, kao i preostale mine i (tužan) indikatorski smiley. Iscrtavanje svih grafičkih elemenata na ekrenu ostvareno je izdvajanjem odgovarajućih pixela, tj. adresiranjem slike (bitmape od koje su ovim postupkom izdvojene nove) koja sadrži sve potrebne komponente (smiley-je, brojeve, prazna polja, neotvorena polja, brojač, mine i zastavice). Ovo je ostvareno upisivanjem u grafičku memoriju, pri čemu pozivana funkcija za upis kao parametre prima baznu adresu, offset za grafičku memoriju, kao i željenu boju ili u slučaju iscrtavanja bitmapa odgovarajuću kombinaciju pixela. Textualna memorija se ne koristi, te nije moguće ispisavanje karaktera na ekran na bilo koji drugi način.

Pri rešavanju ovog dela zadatka korišćen je programski jezik C i Xilinx-ovi alati i okruženje, kao i E2LP platforma na koju se spušta sam programski kod i preko koje korisnik zadaje komande što je ostvareno odgovarajućim mapiranjem prekidača i tastera u okviru ucf file-a.