|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Saša Serafimovski RА162/2016

Nemanja Lekić RА48/2015

Aleksa Janjatović RА126/2016

**Bomberman - Unapređen**

Projektni Zadatak

- Logičko projektovanje računarskih sistema 2 -

Mentor: Miloš Subotić

Novi Sad, jun 2019

**Sadržaj**

[1. Opis igre i Gameplay 3](#_Toc472220115)

[2. Opis rada (uvod) 3](#_Toc472220116)

[3. Datoteka - Battle city 3](#_Toc472220117)

**1.Opis igre i Gameplay**

Igrica Bomberman realizovana je kao projektni zadatak u okviru predmeta LPRS2. Dizajnirana je za FPGA platformu – Microblaze procesor.

**Bomberman** je strategijska igra koja je originalno razvijena u *Hudson Soft* komplaniji, a trenutno je poseduje *Konami*. Prvi put igra je objavljena 1983. godine, a poslednja verzija *Super Bomberman R,* objavljena je trećeg marta 2017. godine. Danas je Bomberman razvijen na više od 70 različitih platformi.

Mapa igrice je kvadratnog oblika, sačinjena od blokova i ciglica. Blokovi se ne mogu uništiti bombom, a Bomberman i neprijatelji ne mogu da prođu kroz blok. Cigla se uništava bombom i stvara se novi prolaz u mapi.

Kada igra započne, na mapi se nalazi Bomberman i četiri neprijatelja koja se kreću po mapi. Njegov zadatak je da postavlja bombe i ubija neprijatelje. Takođe, Bomberman mora na vreme da se pomeri od bombe koja eksplodira posle određenog vremena, jer gubi život ukoliko ne pobegne. Mora se paziti na neprijatelje koji se kreću i ako dođu u kontakt sa Bombermanom takođe mu oduzimaju život. Ukoliko Bomberman izgubi sva tri života, na ekranu se iscrtava mapa na kojoj piše „Game over“ i tu se igra završava. Ukoliko igrač uspe da eliminiše sve neprijatelje, na mapi se pojavljuju vrata i kada igrač stane na njih prelazi na sledeći nivo ove igrice ili pobeđuje.

**2.Opis rada**

Na početku, bilo je potrebno generisati **sprajtove** (sličice komponenata na mapi).

Priprema sličica:

\*sve sličice potrebno je izmeniti da budu dimenzija 16x16 sa ekstenzijom .*jpg.*

Generisanje sprajtova:

\* Visual Studio program *battle-mem* generiše memorijske fajlove , koje je potrebno kopirati u fajl *ram.vhd.* Fajl *ram.vhd* predstavlja sadržaj memorije. Na početku ovog fajla se nalaze palete boja koje se koriste za različite sprajtove. Nakon paleta sledi niz sprajtova. Oni obuhvataju sličice Bombermana, blokova, neprijatelja, bombe i svih ostalih komponenata mape. Prilikom iscrtavanja sprajtova, hardver prolazi kroz ove adrese i iščitava indekse (adrese) boja odgovarajućih piksela i iscrtava ih. Program *battle-mem* sadrži funkcije koje generišu odgovarajući oblik parova adresa i polja, kako se duge sekvence ovih parova ne bi kucale ručno. U okviru funkcija vrši se računanje adresa i mapiranje sprajtova na odgovarajuće boje, odnosno mapiranje same mape na sprajtove.

**3. Datoteka - Battle city**

U datoteci *battle\_city.c* nalazi se glavni deo programa. Čitav program nalazi se u beskonačnoj petlji kako bi se omogućio Gameplay.

**Mapa:**

Pri pokretanju igrice prvo se iscrta mapa pomoću funkcije *map\_update,* koja se poziva svaki put kada je potrebno da se mapa iscrta odnosno promeni. Ukupno u projektnom zadatku postoje tri različite mape koje se menjaju u zavisnosti od trenutnog statusa igrice.

**Main Menu:**

Funkcija *main\_menu* omogućava da se izabere broj neprijatelja i njihova brzina. Zatim pritiskom na srednji taster **JOY2** ulazimo započinjemo igricu na *map1*.

**Kretanje Bombermana:**

Pritiskom tastera na E2LP ploči podešava se kretanje Bombermana. On može da se kreće svuda po mapi, osim na mestima na kojima se nalaze cigle i blokovi. To je omogućeno funkcijom *bomberman****\_****move.* Smer kretanja Bombermana se podešava u zavisnosti od toga da li su u okolini Bombermana određene prepreke koje se detektuju funkcijom *obstacles****\_****detection.*

**Kretanje neprijatelja:**

U *main****\_****menu* birmao broj neprijatelja koji će biti u igri. Startom igre izabrani broj neprijatelja se kreću po mapi u četiri smera pomoću funkcije *random\_move\_enemy.* Svaki se kreće na slučajan način (smer).

**Detekcija i uništavanje prepreka:**

Prepreke koje treba da uništi bomba (neprijatelji i cigle) detektuju se pomoću funkcije *obstacles\_detection.*  Bomba se postavlja pritiskom srednjeg taster na E2LP ploči i nakon određenog vremena eksplodira. Ona takođe oduzima život i samom Bombermanu ako ne pobegne na vreme.

**Detekcija i uništavanje neprijatelja:**

Detekcija za uništenje neprijatelja za se vrši u okviru funkcije *explosion\_detection* koja se poziva unutar same funkcije *detonate*. Ako se primeti da je neprijatelj na putu ekplozije, poziva se *destroy\_enemy* funkcija koja prolazi kroz niz struktura neprijatelja za zadatu mapu i proverava koji je tačno neprijatelj u pitanju.

**Super moći**

Tokom igre postoji 2 tipa pojačanja za bombermana. Generišu se random nakon eksplozije u njihovoj oklini (funkcija *place\_random\_power\_up*). Skupljanjem njih bomberman dobija odredjene moći

* **PowerUP Jačanje bombe -**  Skupljanjem ove moći dobijamo veći opseg eksplozije (max 6)
* **PowerUP Broj Bombi –** Skupoljanjem ove moći bomberman dobija +1 dodatnu bombu koju može da postavi (max 3)

**Kraj igre**

Postoje 2 ishoda igre

1. **WIN –** Ukoliko bomberman uništi sve neprijatelje koji se kreću po mapi pojavljuje se na mapi vrata kroz koje bomberman mora proći da bi uspešno prešao nivo i prikazujemo »GAME WON« screen
2. **LOSE –** Bomberman sadrži 2 života, ukoliko izgubi oba igra se završava i prikazuje se »GAME OVER« screen.

Bomberman život može izgubiti nailaženjem na neprijatelja koji ga ubije i vraća ga na početak mape.

Bomberman život može izgubiti ako se nalazu u dometu eksplozije prethodno postavljenje bombe.

U slučaju svakog gubitka života poziva se funkcija *kill\_bomberman* koja smanjuje broj života bombermana i koja ga vraća na početnu poziciju. Funkcija *find\_bomberman* se koristi kao veza između bombermana i eksplozije (ali i neprijatelja) pošto se sama lokacija bombermana ne upisuje u mapu nego se iscrtava pojedinačno preko *char\_spawn*.

**Postavljanje i širenje eksplozije**

Glavna funkcija za proveravanje eksplozija se zove *place\_explosion* koja dirketno u zavisnosti od smera i polja postavlja eksploziju. Unutar ove funkcije se takođe proverava da li se eksplozija sme prostirati u zatadom smeru. Ova funkcija se poziva unutar *detonate,* dok funkcija *check\_and\_remove\_explosions* za svaku postavljenu eksploziju proverava da li je istekla. Eksplozija se pamti kao struktura *explosion\_t*.