



UNIVERZITET U NOVOM
SADU
FAKULTET TEHNIČKIH
NAUKA NOVI SAD



Alesandra Jagodić, PR161/2017
Marija Vukomanović, PR 67/2015
Irina Jovanović, PR 11/2015
Milica Maravić, E3 34/2014
Sara Pajčin, E3 33/2014

Projekat T3, Dyna Blaster

Distribuirani računarski sistemi u
elektroenergetici

- Primenjeno softversko inženjerstvo –

Uvod

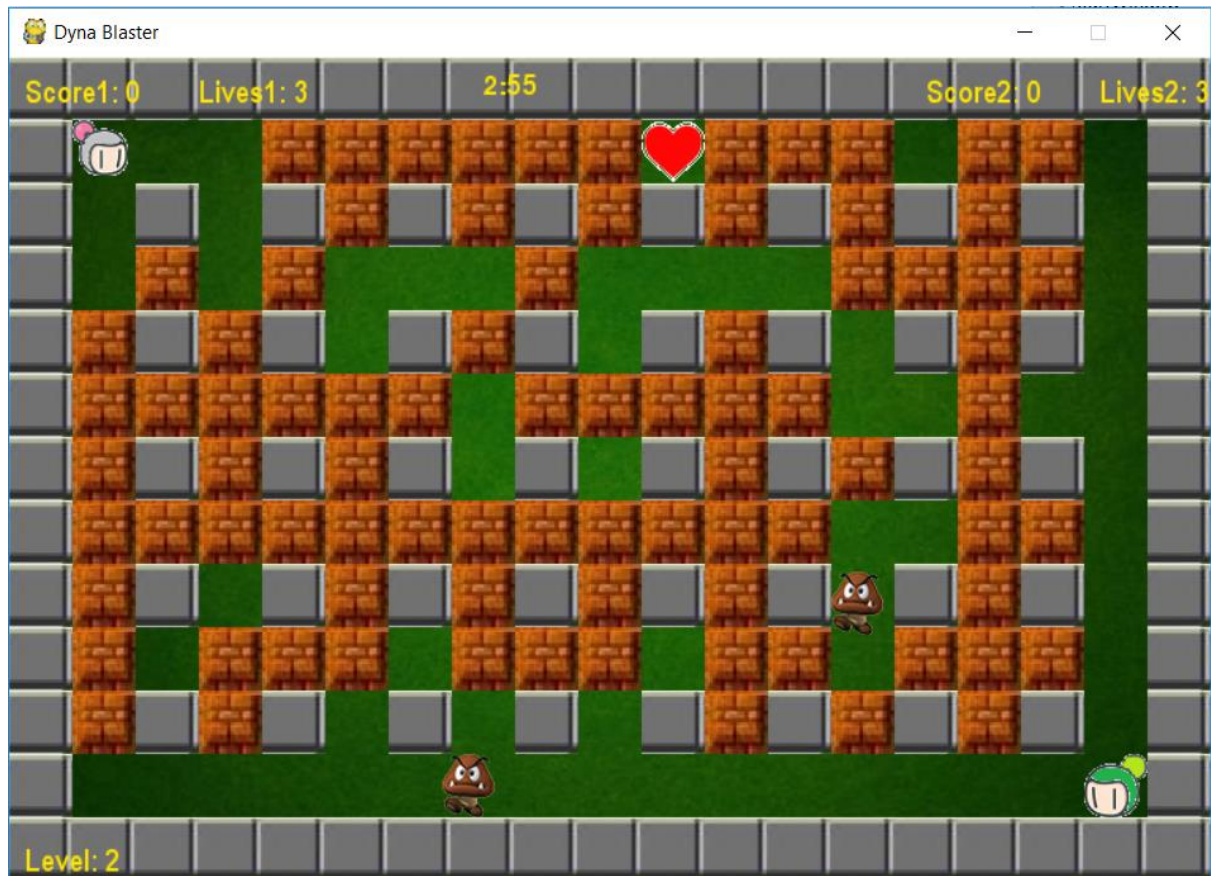
Projekat DynaBlaster predstavlja igricu napravljenu po uzoru na DynaBlaster igru u programskom jeziku Python. Igricu je moguće igrati pojedinačno, a postoji mogućnost igranja sa 2 igrača. Originalna igrica takođe podržava igru jednog ili više igrača. Igrači se kreću po predefinisanoj mapi, izbegavajući čudovišta i postavljajući bombe u cilju eliminisanja drugog igrača ili eliminisanja čudovišta. Nakon uništavanja svih čudovišta prelazi se na sledeći nivo, gde se sva čudovišta ponovo generišu i povećava se težina igrice pojavljivanjem većeg broja čudovišta. Igrica se završava kada jedan od igrača izgubi sva tri života ili mu istekne predefinisano vreme igranja (3 min). Pobednik je onaj koji ostane poslednji. Životi se gube ukoliko je igrač zahvaćen eksplozijom bombe ili prilikom dodira sa čudovištem. S vremena na vreme se u gornjem delu ekrana pojavljuje srce (dodatna sila) koja na random način povećava ili smanjuje 1 život igraču koji se pozicionirao na njega.

Pokretanjem same igrice otvara se startni prozor i pruža se mogućnost izbora između play i multiplayer igranja. Startni prozor je urađen u skripti StartWindow.

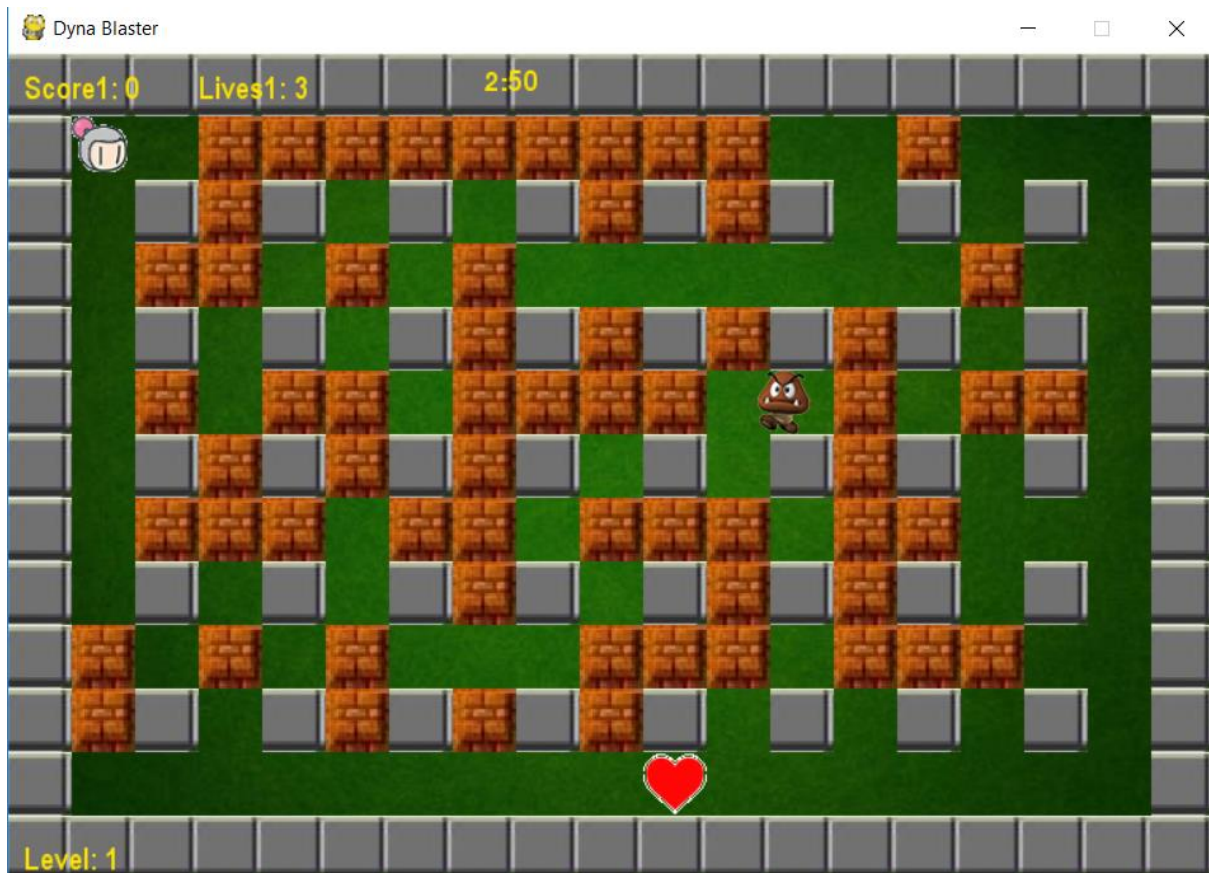


Startni Prozor

Na sledećoj slici se vidi režim sa 2 igrača koji su trenutno na nivou 2 i dodatna sila (srce). Sila je implementirana u klasi **Gifts**, koja pored inicijalizacije sadrži metode `update`, `draw`, `poziv`, `force` (na random način povećava ili smanjuje živote). Igrači i čudovista su implementirani u klasama **Bomberman** i **Enemy**. Klasa Bomberman sadrži inicijalizaciju (slike i koordinate gde će se pojaviti igrač), metode : `move` (pomeranje igrača u svim smerovima), `score_up`, `score_show`, `hide` (privremeno sklanja igrača sa ekrana), `update`, `lives_down`, `lives_up` i `show_lives`. Klasi Enemy prosledjujemo listu slobodnih pozicija u matrici iz koje uzimamo random coordinate na koje se čudovista postavljaju svaki put pri pokretanju igrice ili prelaskom na sledeći nivo. Sadrži metode `set_random_direction`, `update`, i `move` (proverava po kojim koordinatama čudoviste može da se kreće).

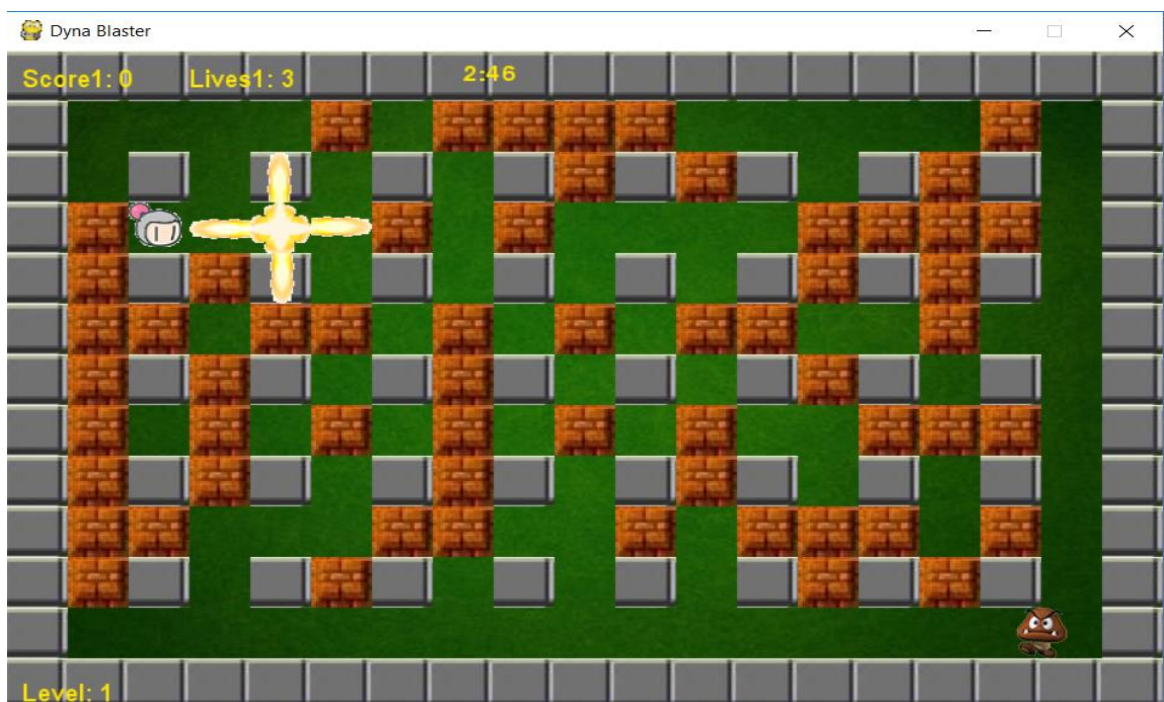


Multiplay režim, nivo 2, prikaz dodatne sile

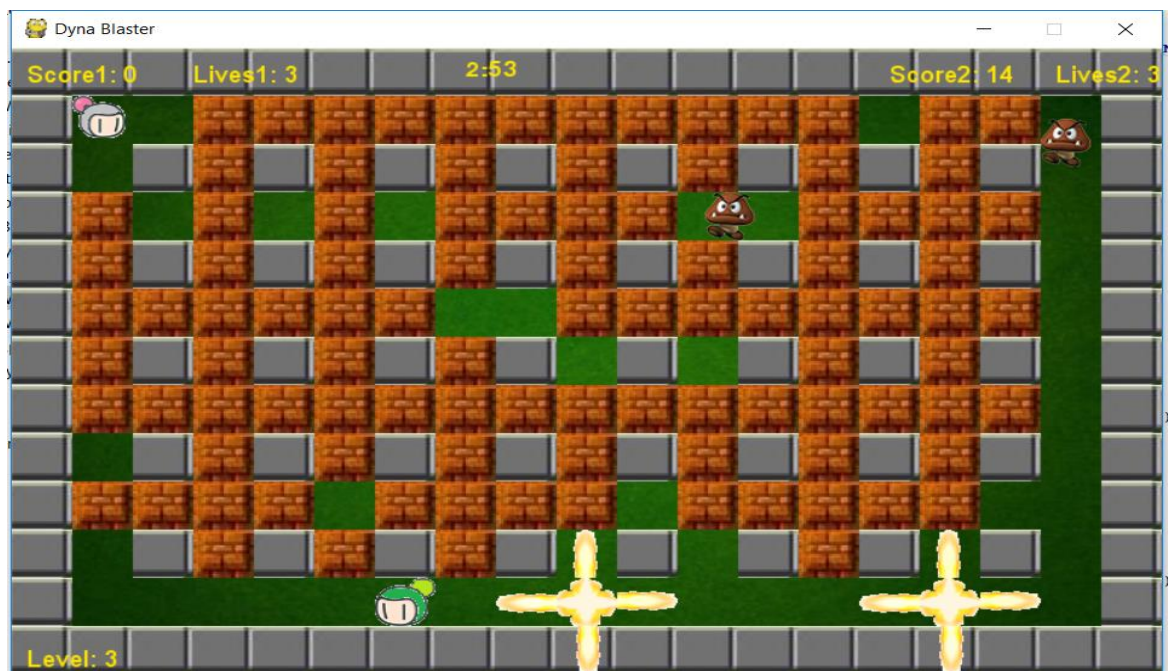


Režim sa jednim igračem,nivo 1

Na sledećim fotografijama se vidi aktivirana bomba ,koja je implementirana u klasi **Bomb**. Sadrži metode update,explode (proveravamo da li se igrač, uništivi zid ili neprijatelj nalazi u njenom opsegu eksplozije) i draw (iscrtavanje eksplozije).

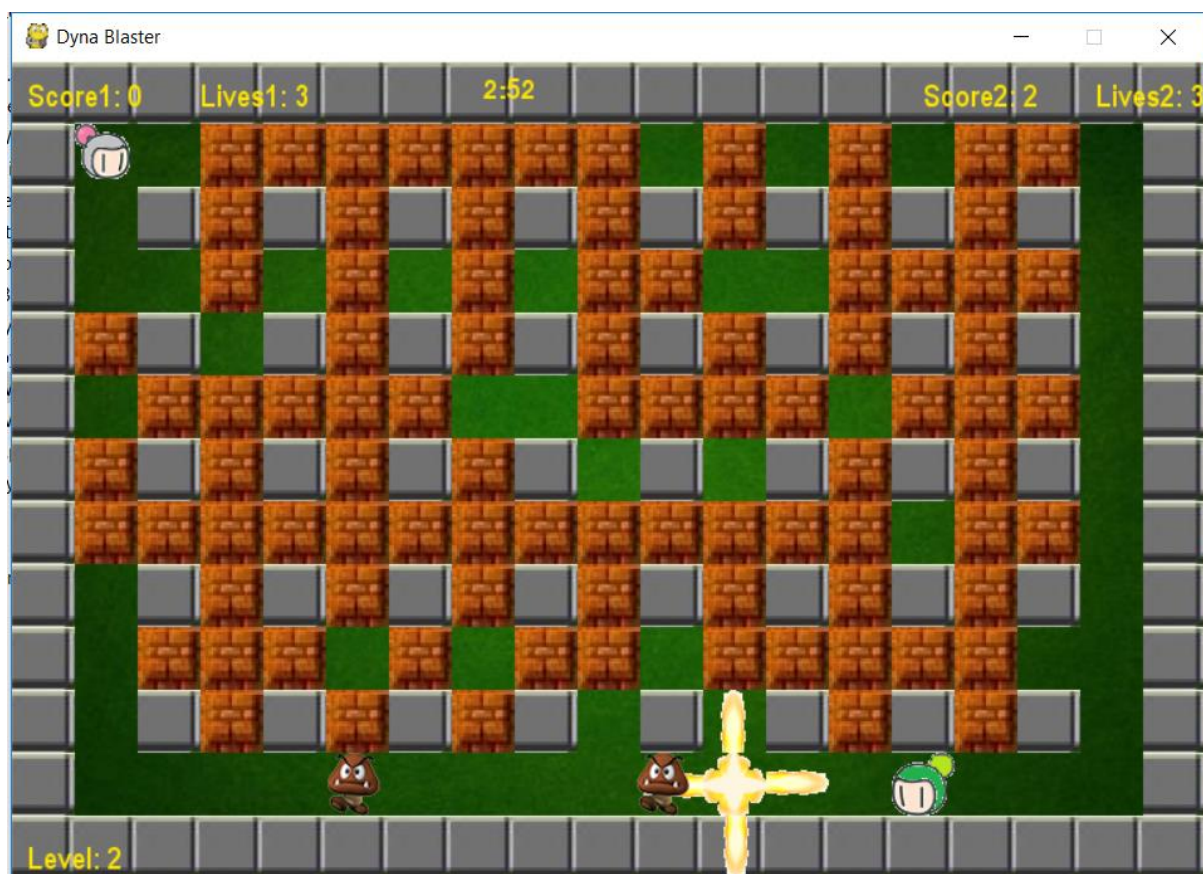


Eksplzija bombe

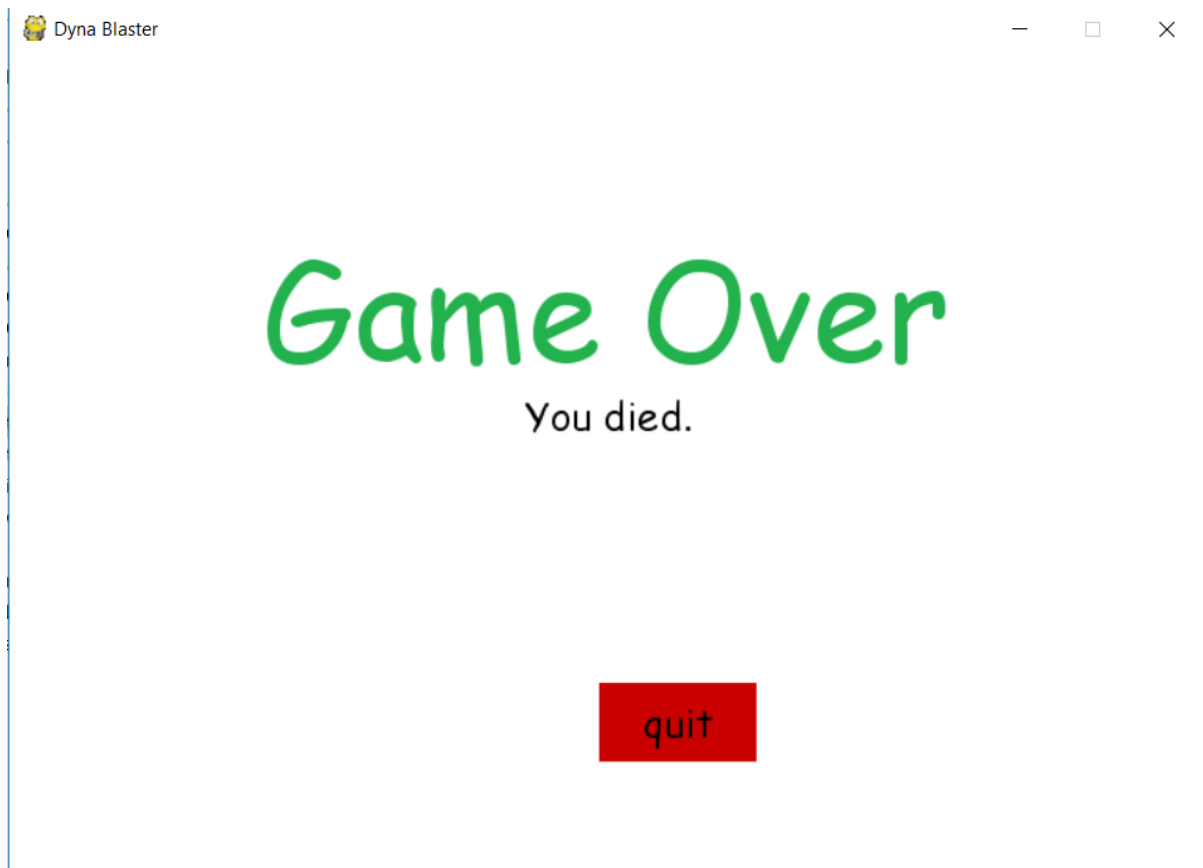


Eksplodije 2 bombe,nivo 3,multiplay

Nakon eksplozije kao sto je prikazana na narednoj fotografiji,čudoviste je eliminisano i zeleni igrač dobija poene.



U slučaju da se igrač nađe u domenu eksplozije i ostane bez života, otvara se sledeći prozor :



2. Razvoj aplikacije

Za razvoj ove igrice korišćen je Python programski jezik, Pygame framework i time, os, random i multiprocessing biblioteke za paralelizaciju rada.

Python programski jezik je interpretativni, „visoki“ programski jezik koji podržava više tipova paradigmi, kao što su: objektno-orijentisana, imperativna, funkcionalna, proceduralna. Korišćenje *Pythona* nam je mnogo olakšava sam razvoj igrice jer automatski vodi računa o stvarima kao što su *memory management*, o tipovima podataka, lako skaliranje aplikacija, podržava *cross-platform* programiranje, itd.

Loše strane *Pythona* su pre svega što je *Python* spor, pored toga nije baš najidealnije rešenje kada do programiranja *multi-processor/multi-core* aplikacija, ima svoje limitacije kada je u pitanju pristup bazama podataka, gotovo je nemoguće napraviti kvalitetnu 3D grafiku, i nije najsajnije rešenje prilikom rešavanja zadataka koji zahtevaju optimizaciju korišćenja memorije.

Framework koji je korišćen za izradu ove igrice je Pygame. Zamenio je PySDL i danas se koristi za izradu multimedijalnih aplikacija kao što su igre. Koristi SDL biblioteku koja omogućava razvoj igara u realnom vremenskom periodu bez korišćenja programskog jezika C. Aplikacije koje koriste pygame su vrlo prenosive i mogu raditi i na android telefonima i tabletima uz upotrebu pgsp4a (pygame subset for android).

Jedan od najvećih problema SDL-a je što ne podržava vektorsku matematiku, detekciju kolizije i upravljanje 2D i 3D grafičkim prikazom, MIDI suport ili manipulaciju nizom piksela.

Multiprocessing biblioteka, koja nam pruža kako lokalnu tako i distribuiranu konkurenciju. Zamenujući Global Interpreter Lock (mehanizam u Python-u koji garantuje da samo jedan thread može istovremeno da izvršava bytecode) koristeći podprocese umesto thread-ova. Zahvaljujući ovom modulu (modul je fajl u Python-u sa ekstenzijom .py koji sadrži definicije i izraze) koji dozvoljava programeru da u potpunosti iskoristi procesore na datoj mašini i samim tim paralelizuje izvršavanje zadataka. Biblioteka se koristi i na Windows i na Unix platformama.

Problem koji se javlja prilikom korišćenja multiprocessing biblioteke umesto thread-ova je što se koriste procesi koji svaki ima zasebni memorijski prostor što otežava deljenje objekata i resursa između procesa, dok korišćenjem thread-ova koristimo deljeni memorijski prostor i ovaj problem ne postoji. Kod thread-ova moramo zato koristiti neke od mehanizama zaključavanja kako ne bi više niti pisalo na istu memorijsku lokaciju, što bi dovelo do nekonzistentnog stanja prilikom izvršavanja aplikacije.

Zaključak

Razvijajući ovu igricu imali samo priliku da se susretnemo sa nekim novim konceptima koje *Python* omogućava. Korišćenje programskog jezika koji do sada nije viđen, kao i potpuno novi *framework* i biblioteke doprinele su da ovaj zadatak bude izazovan. Korišćeno je paralelno programiranje i projekat je razdvojen na više komponenti. Paralelizacija je demonstrirana preko thread-ova i procesa koji izvršavaju određene module, a pored toga Pygame već ima ugrađene mehanizme za paralelnu obradu modula, te je kroz ovaj projekat programski upotreba niti i procesa demonstrirana pri izvršavanju samo određenih delova aplikacije.