

Projektna dokumentacija iz predmeta Distribuiranih  
računarskih sistemi u elektroenergetici

Zadatak: Galaga (T2)

Programski jezik: Python

Naziv tima: NoName

Članovi tima:

Mladen Milošević PR 77/2015

Saša Dragomirović PR 75 /2015

Kristina Stjepanović PR 128 /2015

Milan Falčić PR 122 /2015

## Sadržaj:

Opis zadatka koji se rešava.....	3
Opis rešenja zadatka.....	4
Zaključak.....	6

# 1. Opis zadatka koji se rešava

Napisati igricu nalik Galaga igrici. Napraviti šemu, sličnu kao u originalnoj igrici, kroz koje avatari mogu da se kreću.

Uloge:

- Dva ili više igrača, pokreću svog avatara pomoću strelica, odnosno ASDW taster. Kreću se levo-desno po dnu ekrana. Mogu da ispaljuju projekte, koji lete vertikalno gore. Svaki igrač ima 3 života po nivou. Na početku nivoa, nalaze se u ćoškovima.

- Vanzemaljci su organizovani u formaciji 3x10. Svi zajedno, kreću se levo-desno, od jedne do druge ivice. Nasumično, nezavisno jedni od drugih, pucaju projekte vertikalno dole. Povremeno jedan ili više vanzemaljaca se obrušava na nekog od igrača. Na početku nivoa, nalaze se na sredini, u gornjem delu ekrana.

Pravila igre:

- Igra se beskonačno nivoa. Ukoliko projektil pipne igrača, on gubi život. Ukoliko projektil igrača pipne vanzemaljca, on nestaje. Nakon svakog nivoa projektili se brže kreću. Za prelaz na sledeći nivo potrebno pogoditi sve vanzemaljce. Pobednik je igrač koji najduže ostane u igri. Igrica se završava kada svi igrači izgube sve živote.

## 2. Opis rešenja zadatka

Rad aplikacije zasniva se na korišćenju labela koje čine da aplikacija tj. svaka promena na ekranu bude realizovana u realnom vremenu. Prirodno je kretanje igrača, neprijatelja i lasera što izaziva zadovoljstvo kod korisnika ovog programa i pruža mu maksimalno užvanje.

Prilikom pokretanja aplikacije, otvara se prozor (meni) koji korisniku omogućava da odabere jednu od tri opcije, a to su:

- One Player: omogućava početak igrice za jednog igrača.
- Two Player: omogućava početak igrice za dva igrača.
- Quit: omogućava izlazak iz programa.

Pokretanjem aplikacije pokreće se file Galaga.py čiji je zadatak da poziva funkcije iz drugog file-a ko će pokrenuti glavne delove programa.

Klasa GameWindow (koja se nalazi u file-u gui.py) otvara glavni i jedini prozor u kome se sve dešava. U ovoj klasi se iscrtava glavni meni, određuje se veličina prozora kao i pozicioniranje prozora u odnosu na ekran. Kada korisnik odabere One Player ili Two Player opciju kreira se nova klasa Game (file game.py) u kojoj se nalazi sva logika ovog programa.

Klasa Game predstavlja glavnu nit u ovom programu, ona izvršava iscrtavanje igrača, neprijatelja, pozicioniranje, brisanje i dodavanje lasera, kao i ispis života i trenutnog nivoa. U glavnom tredu se pokreću sve ostale niti i drugi procesi. Ostale niti vrše proveru da li je došlo do kolizije, da li je potrebno pomeriti igrača, da li je potrebno pomeriti neprijatelja kao i sve ostale stvari koje mogu izazvati promenu na ekranu. U slučaju da je potrebno promeniti nešto na ekranu, ostale niti, kao i procesi koji rade , javljaju glavnoj niti da je potrebno da promeni određenu stvar na ekranu. Niti se uz pomoć signala (pyqtSignal) obraćaju glavnoj niti , a procesi se uz pomoć podataka koje stavljaju na Queue obraćaju drugim procesima.

Klasa Player (player.py) sadrži sve podatke vezane za igrača.

Klasa PlayerShoot (player\_action.py) je klasa koju izvršava posebna nit. Ova klasa vodi računa o tome da li je potrebno iscrtavanje lasera, da li je potrebno pomeranje lasera na gore i da li je doslo do kolizije izmedju lasera i neprijatelja. U slučaju da se nešto od toga desilo ova klasa obaveštava glavu klasu Game tj sporedna nit obaveštava glavnu nit da je potrebno izmeniti sadržaj na ekranu.

U enemy\_action.py nalaze se tri klase:

1. EnemyMove
2. EnemyShoot
3. EnemyAttack

Sve tri klase se izvršavaju u posebnim nitima i uz pomoć signala komuniciraju sa glavnom niti. EnemyMove obezbeđuje kretanje neprijatelja levo-desno. EnemyShoot obezbeđuje neprijateljima mogućnost pucanja lasera ka igraču što podrazumeva signale koji javljaju glavnoj klasi da je potrebno da nacрта laser i pomera ga na niže. EnemyAttack omogućava neprijateljima da napadnu igrača, tj da se jedan neprijatelj iz poslednjeg reda kreće na niže ka igraču.

Dodatni proces koji se kreira u klasi Game zadužen je za nasumično određivanje „sile“ koja se nakon nekog vremena pojavljuje na ekranu i omogućuje igraču da dobije određeni bonus ukoliko je igrač pokupi.

Klasa DeusExMachina se izvršava u posebnoj niti i omogućava iscrtavanje „sile“ koja se dobija iz procesa kreiranog u Game klasi. Proverava se kolizija između „sile“ i igrača, ukoliko je doslo je kolozije (poklapanje koordinata ) signal se salje ka glavnoj niti koja ce odraditi modifikaciju u obliku ubrzavanja igrača, usporavanja igrača, uvećavanjem života ili nemogućnost pucanja od strane igrača.

Brzina pucanja neprijatelja i brzina kretanja lasera neprijatelja se iz nivoa u nivo povećavaju i one su određene uz pomoć ugrađene klase Timer čiji se interval podešava u glavnoj klasi sa unapred definisanim brojkama iz file-a config.py.

### 3. Zaključak

Iako je Python vrlo popularan jezik i dosta jednostavan za savladanje, ne preporučuje se izrada bilo kakvih igara u ovom programskom jeziku. Performanse su dosta lošije u odnosu na druge jezike, distributivnost na druge računare kao i mogućnost povezivanja više računara radi multiplayer-a spadaju u vrlo teške i komplikovane akcije. Biblioteke koje ovaj programski jezik nudi se veoma slabe u poređenju sa ostalim programskim jezicima gde je izrada mnogo lakša i razumljivija za programere. Python je previše ograničen i ne preporučuje se za izradu modernih igrica.