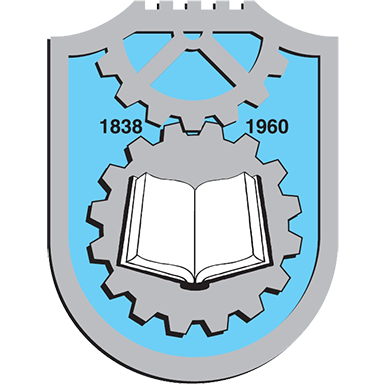
Univerzitet u Kragujevcu

Fakultet inženjerskih nauka



Računarska grafika

Projektni zadatak:

Izrada arkadne igrice ,,Defender"

korišćenjem OpenGL biblioteke

Student: Predmetni profesor:

Mihailo Nerandžić 643/2019 Nenad Filipović

Tijana Šušteršič

Kragujevac 2024.

**Sadržaj**

1. [**Uvod** 3](#_Toc177921633)

[**2.** **Korisničko uputsvo** 5](#_Toc177921634)

[**3. Objašnjenje koda i izrada aplikacije** 5](#_Toc177921635)

4. **Izgled igrice..................................................................................................................................................29**

5. **Zaključak**..........................................................................................................................................................................32

6. **Literatura**.......................................................................................................................................................................33

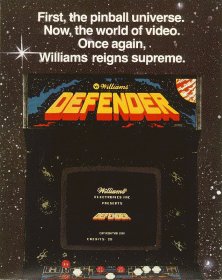
# **Uvod**

Zadatak projekta je napraviti aplikaciju igrice Defender koristeći OpenGL, omogućiti različite opcije (skupljanje poena, gubljenje života, bar dva nivoa itd.) uz preporuku da se koristi razvojno okruženje programskog jezika Visual C++ 6.0 ili Visual Studio.

**"Defender"** je arkadna video igra koju je 1981. godine razvila kompanija Williams Electronics. Kasnije je preneta na mnoge platforme, poput Atari 2600, Commodore 64 i MS-DOS-a, čime je stekla kultni status među zaljubljenicima u arkadne igre.

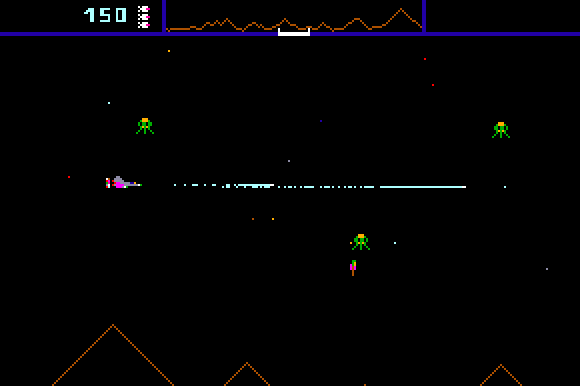
U igri "Defender," igrači preuzimaju kontrolu nad svemirskim brodom u futurističkom okruženju, gde je zadatak da se zaštite ljudi od napada neprijateljskih vanzemaljaca. Igrač patrolira dvodimenzionalnim svetom, spašava civile koje vanzemaljci pokušavaju oteti i uništava neprijateljske letelice pre nego što uspeju da sprovedu svoje planove.

Igra spaja brzu akciju i strateško razmišljanje, jer igrači moraju balansirati između uništavanja neprijatelja i zaštite civila, sve vreme izbegavajući projektile i sudare. Kako igra napreduje, nivo težine raste sa sve agresivnijim protivnicima i složenijim spasilačkim misijama. Ova kombinacija adrenalinske akcije i taktičkog izazova učinila je "Defender" jednom od najvažnijih arkadnih igara tog doba.



*Slika 1 Prva naslovna strana (1981).*

Na sledećoj slici je prikazan izgled prvog nivoa igrice Defender:



*Slika 2 Prvi nivo igrice Defender*

# **Korisničko uputsvo**

Projekat je izrađen po uzoru na "Defender 1981" gde glavni igrač upravlja ljubilčastim trouglom umesto ljubicastom svemirskom letelicom. Zadatak igrača je da ubije sve protivnike kako bi osvojio što više poena i prešao na sledeći nivo. Svaki naredni nivo ima protivnike koji se brže kreću i iz nivoa u nivo i potrebno je biti što veštiji kako bi prešli u naredni nivo.

Glavni igrač koji je predstavljen ljubilčastim trouglom može da se kreće levo, desno, gore, dole korišćenjem strelica na tastaturi respektivno, kao i mogućnost ispaljivanja metkova odnsono kružića svetlo plave boje klikom na taster SPACE. Metkovi pri pogotku protivnika ih automatski ubijaju, a igrač dobija poene.

Igrica se sastoji iz četiri nivoa, gde svaki nivo donosi nove izazove u vidu bržih neprijatelja i njihovim bržim kretanjem. Prvi nivo je lakši, sa dva neprijateljska koja su predstavljane krugovima svetlo plave i zelene boje, dok kasniji nivoi postaju sve teži sa dodatnim neprijateljima. Prva dva čudovišta se kreću najsporije u prvom nivou, na drugom su i dalje dva neprijatelja zelene i žute boje koja su brža u trecem nivou dva neprijatelja koja su još brža, a u četvrtom nivou se pojavljuje treći neprijatelj gde se oni kreću najbrze(žute, svetlo plave i zelene boje). Neprijatelji se u svakom nivou kreću levo desno nezavisno jedani od drugih.

Tranzicija iz jednog u drugi nivo se dešava automatski kada igrač eliminiše sve neprijatelje na trenutnom nivou. Na ekranu su prikazani broj poena, nivo i broj preostalih života. Igrač dobija 3 života startovanjem igrice kada izgubi jedan nivo se nastavlja ali sa jednim životom manje, a gubljenjem sva tri, igra se završava i izbacuje ekran sa osvojenim brojem poena i sa mogućnošću izlaska iz igrice ili ponovnog pokretanja iste.

# **Objašnjenje koda i izrada aplikacije**

Za izradu ovog projekta upotrebljen je softver Visual Studio 2022, dok je program napisan pomoću programskog jezika C++ uz korišćenje biblioteke OpenGL.

OpenGL (engl. Open Graphics Library) predstavlja višeplatformski programski interfejs za pisanje programa koji rade sa dvodimenzionalnom i trodimenzoinalnom računarskom grafikom. Ovaj interfejs čini veliki broj različitih funkcija koje se mogu koristiti za izradu kompleksnih trodimenzionalnih scena od jednostavnih elemenata.

OpenGL nudi mogućnost primene različitih transformacija nad objektima u sceni, nudi različite vrste projekcija, nudi odbacivanje delova objekata koji su posmatraču nevidljivi, primenu tekstura i još niz drugih mogućnosti. Razvijen je od strane Silicon Graphics-a 1992. godine i popularan je u industriji video igara gde je pandan Microsoft-ovom Direct3D-u.

Početni(prvi) ekran koji se prikazuje pri pokretanju igrice:

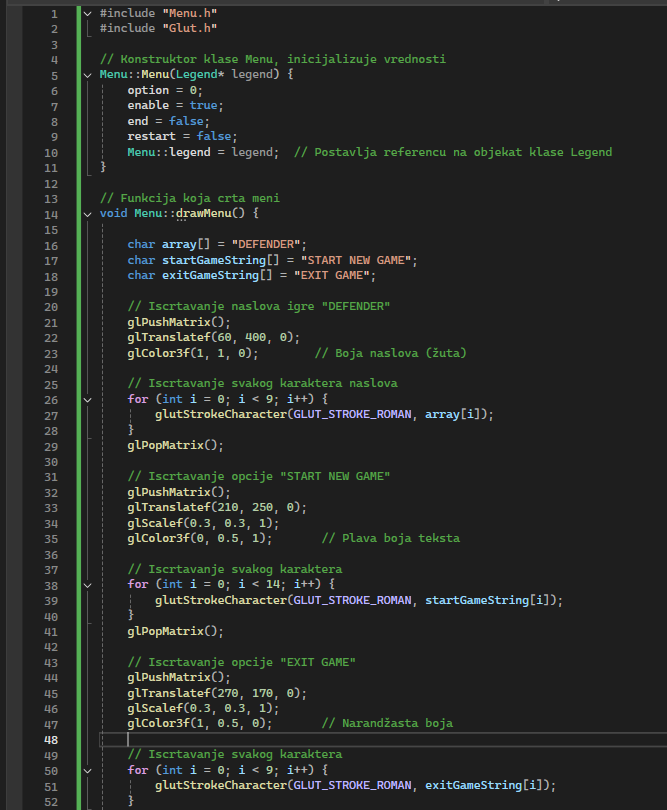


*Slika 3 Početni ekran*

Kod koji se nalazi u fajlovima Menu.cpp i Menu.h odgovoran je za prikazivanje i upravljanje početnim ekranom igre "Defender". Početni ekran je deo korisničkog interfejsa gde igrač može započeti igru ili izaći iz aplikacije. Ovaj kod obuhvata nekoliko osnovnih funkcionalnosti, uključujući crtanje teksta, prikazivanje dugmića na ekranu, i detektovanje klikova miša kao i inicijalizaciju prvog nivoa igre.

Na početku koda su definisane biblioteke koriščćene za izradu ovog projekta, a zatim funkcija za crtanje menija prikazuje ključne elemente korisničkog interfejsa igre. Prvo, iscrtava se naslov igre "DEFENDER" na vrhu ekrana, gde se koristi žuta boja za tekst. Ovaj naslov se pozicionira pomoću funkcije za translaciju kako bi se postavio na odgovarajuću koordinatu u prozoru igre.

Nakon naslova, prikazuju se dve opcije menija: "START NEW GAME" i "EXIT GAME". Opcija za pokretanje nove igre iscrtava se plavom bojom, dok je opcija za izlazak iz igre prikazana narandžastom bojom. Tekst ovih opcija se pozicionira ispod naslova, svaka na različitim koordinatama, i skalira kako bi se vizuelno uklopila u meni.



*Slika 4 Fajl Menu.cpp 1*

Strelica u meniju prikazuje trenutnu izabranu opciju, počevši od "START NEW GAME." Kada korisnik izabere drugu opciju, strelica se pomera pored "EXIT GAME." Strelica je crveni trougao, pozicioniran i skaliran kako bi bio vidljiv pored aktivne opcije.

Nakon završetka igre, menja se prikaz menija. Prikazuje se poruka "GAME OVER" narandžastom bojom na vrhu ekrana, a zatim se iscrtava rezultat uz tekst "SCORE:" žutom bojom. Opcije "PLAY AGAIN" i "EXIT" su prikazane zelenom i crvenom bojom. Strelica opet označava aktivnu opciju, bilo da je reč o ponovnom igranju ili izlasku iz igre.



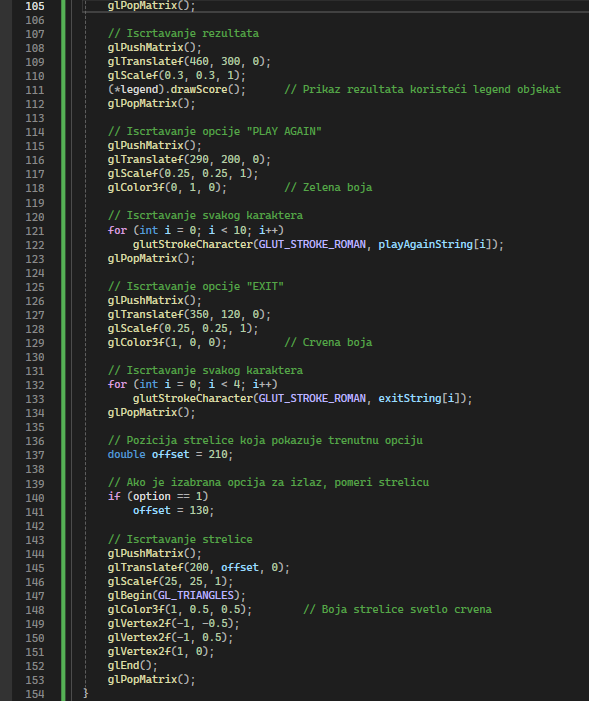
*Slika 5 Menu.cpp 2*

Ovaj deo koda prikazuje završni meni igre, uključujući rezultat, opcije za ponovno igranje i izlazak, kao i strelicu koja označava izabranu opciju.

Prvo se prikazuje rezultat igrača na ekranu pomoću objekta klase Legend, koji poziva funkciju drawScore() i pozicionira rezultat na odgovarajućoj koordinati (460, 300), uz odgovarajuće skaliranje za vizualnu korekciju.

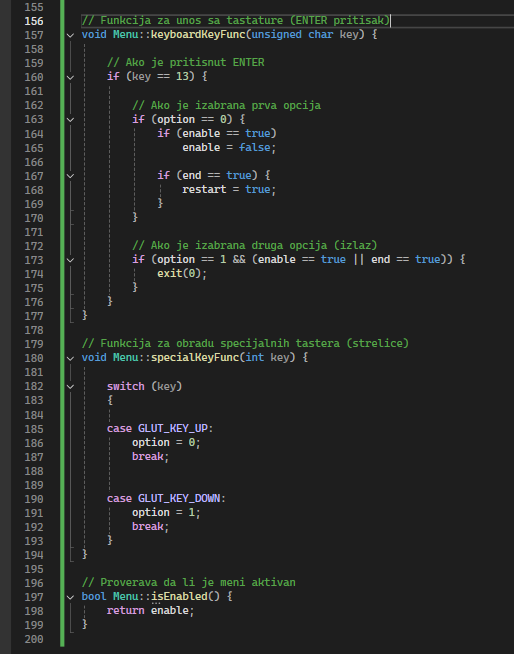
Zatim se iscrtava opcija "PLAY AGAIN", zelene boje, koja se nalazi na koordinatama (290, 200). Svaki karakter iz niza se postepeno prikazuje na ekranu, nakon čega sledi iscrtavanje opcije "EXIT", crvene boje, koja je pozicionirana ispod prve opcije na koordinatama (350, 120).

Na kraju, strelica koja pokazuje na trenutno izabranu opciju se pozicionira. Ako je korisnik izabrao prvu opciju, strelica ostaje na početnoj poziciji (210), dok se u slučaju izbora opcije "EXIT", strelica pomera na nižu poziciju (130). Strelica je svetlo crvene boje i iscrtana je kao trougao koji se pozicionira pored aktivne opcije.



*Slika 6 Menu.cpp 3*

Ovaj deo koda sadrži funkcije za unos sa tastature i obradu specijalnih tastera.Funkcija za unos reaguje na pritisak tastera "ENTER". Ako je prva opcija izabrana, meni se onemogućava, a ako je igra završena, postavlja se varijabla restart na true. U slučaju izbora druge opcije, poziva se exit(0) za izlazak iz igre.Funkcija za specijalne tastere omogućava navigaciju kroz meni pomoću strelica. Pritiskom na strelicu na gore ili na dole, izabrana opcija se menja između "START NEW GAME" i "EXIT GAME".Funkcija isEnabled vraća status aktivacije menija putem varijable enable, omogućavajući drugim delovima koda da znaju da li je meni aktivan.

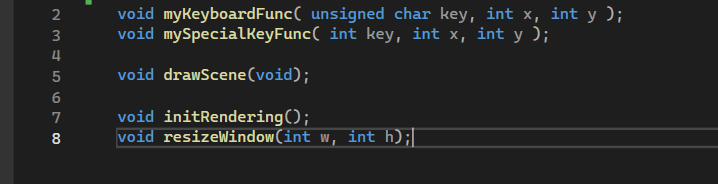
  
*Slika 7 Menu.cpp 4*

U header fajlu se definišu prototipovi funkcija koje se koriste u programu.

myKeyboardFunc obrađuje pritisak tastera sa tastature, uzimajući u obzir karaktere i poziciju kursora (x, y). mySpecialKeyFunc se fokusira na specijalne tastere, kao što su strelice, i takođe uzima njihove kodove i poziciju kursora.

drawScene je zadužena za iscrtavanje grafike u aplikaciji, dok initRendering inicijalizuje renderovanje, podešavajući potrebne parametre za prikaz. Na kraju, resizeWindow omogućava promenu veličine prozora, podešavajući širinu i visinu prilikom promene dimenzija.

Sve ove funkcije su ključne za interakciju korisnika sa programom i za upravljanje grafikom.



*Slika 8 Menu.h*

Ovaj kod definiše klasu Legend, koja je odgovorna za upravljanje i iscrtavanje informacija o igraču, uključujući rezultat, živote i nivo.

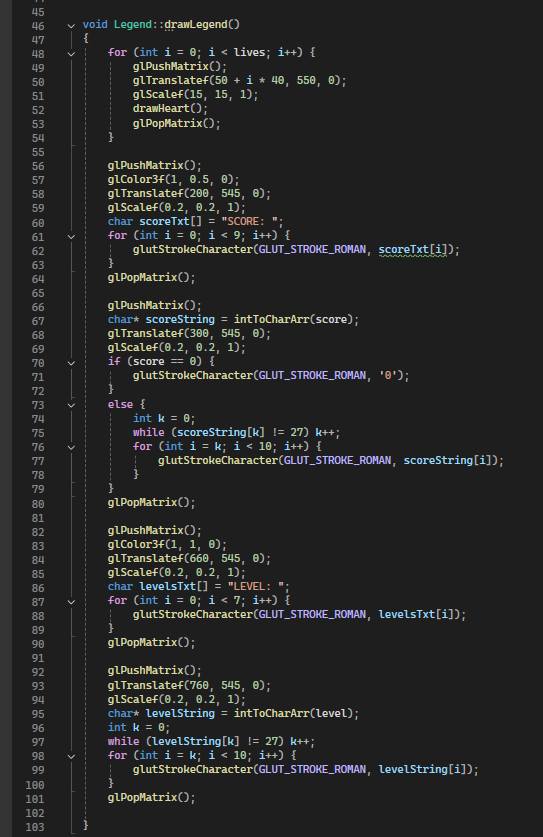
U konstruktoru se inicijalizuju score (rezultat), lives (broj života) i level (nivo).

Funkcija dotDistance koristi se za izračunavanje određene vrednosti potrebne za crtanje srca. Funkcija drawHeart iscrtava srce koristeći GL\_POLYGON i određuje njegove tačke.

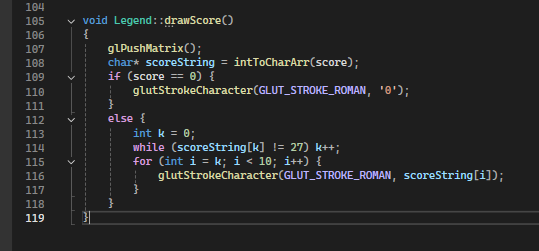
Funkcija intToCharArr konvertuje celobrojnu vrednost u niz karaktera, koji se kasnije koristi za iscrtavanje rezultata i nivoa.

Funkcija drawLegend prikazuje broj života (kao srca) na vrhu ekrana, iscrtava tekst "SCORE:" uz trenutni rezultat, kao i "LEVEL:" uz trenutni nivo. Rezultat i nivo se konvertuju u string pre nego što se iscrtaju.

Funkcija drawScore posebno iscrtava trenutni rezultat igrača, koristeći istu logiku konverzije kao u prethodnoj funkciji, ali se fokusira samo na iscrtavanje rezultata.

***Slika 9 Legend.cpp 1*

*Slika 10 Legend.cpp 2*

**

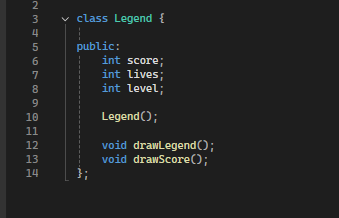
*Slika 11 Legend.cpp 3*

Ova klasa Legend definiše tri javne članice: score (rezultat), lives (broj života) i level (nivo), koje čuvaju informacije o stanju igrača u igri.

Konstruktor Legend inicijalizuje ove članice sa podrazumevanim vrednostima.

Metoda drawLegend koristi se za iscrtavanje vizuelnih informacija o igraču, uključujući život i rezultat, dok metoda drawScore iscrtava samo trenutni rezultat igrača.

Klasa služi kao način za organizovanje i prikaz informacija o igraču tokom igre.



*Slika 12 Legend.h*

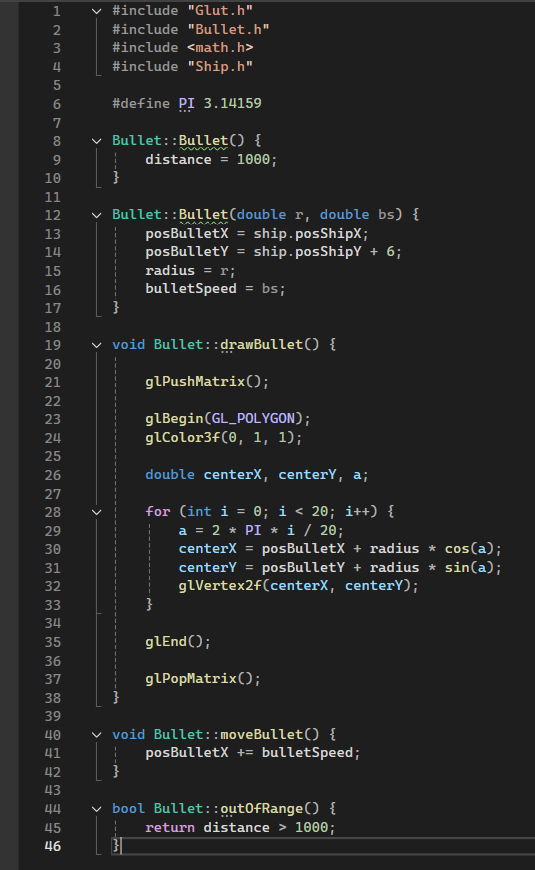
Ova klasa Bullet definiše metke ispaljene iz svemirskog broda u igri.

Konstruktor bez parametara inicijalizuje metak na podrazumevanu udaljenost od 1000 jedinica, dok konstruktor sa parametrima postavlja poziciju metka na osnovu trenutne pozicije broda i određuje radijus metka i brzinu metka.

Metoda drawBullet crta metak kao poligon, koristeći kružne koordinate za iscrtavanje oblika metka. Metak je cijan boje i pozicioniran prema izračunatoj vrednosti.

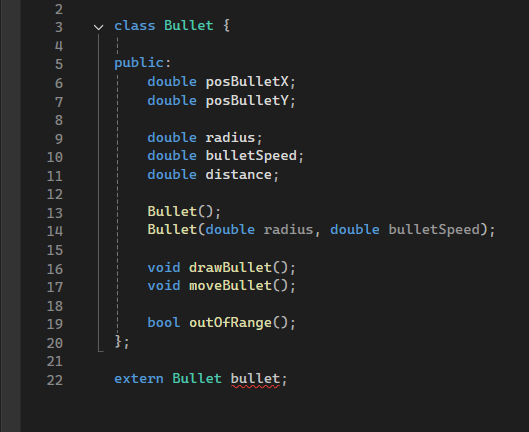
Metoda moveBullet pomera metak unapred u zavisnosti od njegove brzine.

Metoda outOfRange proverava da li je metak prešao maksimalnu dozvoljenu udaljenost, vraćajući true ako je metak izvan opsega.



*Slika 13 Bullet.cpp*

Klasa Bullet definiše metak sa atributima za poziciju, poluprečnik, brzinu i pređenu udaljenost. Konstruktor postavlja ove vrednosti, a metode služe za crtanje i pomeranje metka na ekranu. Takođe, postoji metoda koja proverava da li je metak izašao iz dozvoljenog opsega. Extern označava da je metak globalno dostupan u drugom fajlu.



*Slika 14 Bullet.h*

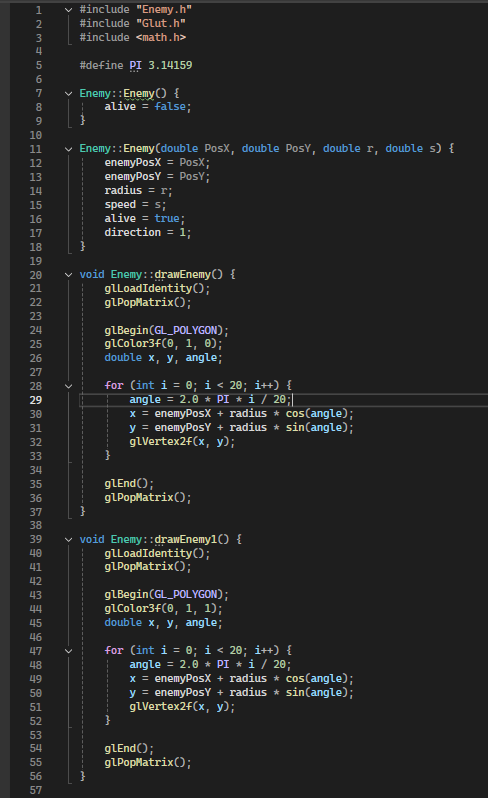
Klasa Enemy predstavlja neprijatelja u igri i sadrži atribute kao što su pozicija neprijatelja (enemyPosX i enemyPosY), poluprečnik (radius), brzina kretanja (speed), smer kretanja (direction), i status života (alive). Konstruktor bez parametara kreira neaktivnog neprijatelja, dok parametrizovani konstruktor inicijalizuje poziciju neprijatelja, poluprečnik, brzinu i postavlja status neprijatelja kao aktivan.

Postoje tri metode za iscrtavanje neprijatelja: drawEnemy, drawEnemy1 i drawEnemy2, koje crtaju neprijatelje različitih boja (zelena, svetloplava i žuta). Ove metode koriste OpenGL funkcije za crtanje kruga koji predstavlja neprijatelja, gde se krug definiše kao poligon sa tačkama postavljenim na osnovu pozicije i poluprečnika.

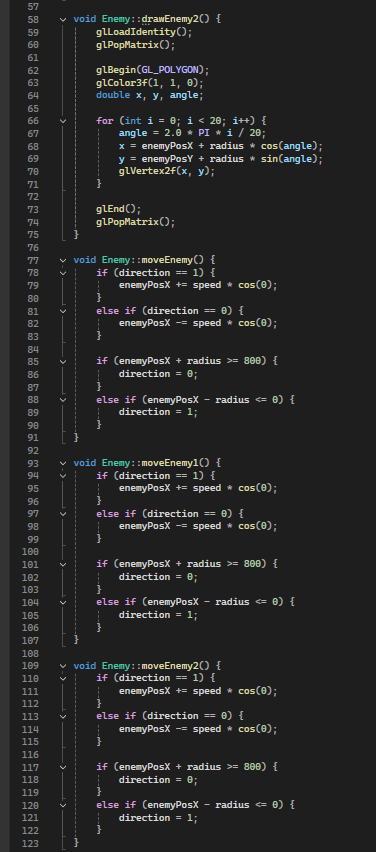
Kretanje neprijatelja je implementirano u metodama moveEnemy, moveEnemy1 i moveEnemy2. Neprijatelj se kreće horizontalno po ekranu, a smer kretanja se menja kada neprijatelj dođe do ivice ekrana. Ako neprijatelj stigne do desne ivice, on se kreće u levo, i obrnuto.

Metode bulletCollision, bulletCollision1 i bulletCollision2 proveravaju da li je došlo do sudara između neprijatelja i metka. One koriste funkciju collision koja izračunava udaljenost između dve tačke (pozicija neprijatelja i metka), i ako je ta udaljenost manja ili jednaka zbiru poluprečnika, smatra se da je došlo do sudara.

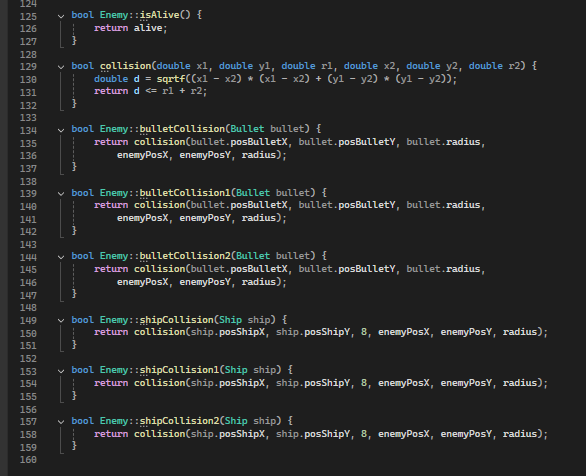
Na sličan način, metode shipCollision, shipCollision1 i shipCollision2 proveravaju sudar između neprijatelja i broda igrača. Ove metode takođe koriste funkciju collision, ali umesto metka, u obzir uzimaju poziciju i veličinu broda. Ako dođe do sudara, neprijatelj može oštetiti ili uništiti brod igrača.



*Slika 15 Enemy.cpp*



*Slika 16 Enemy.cpp 2*



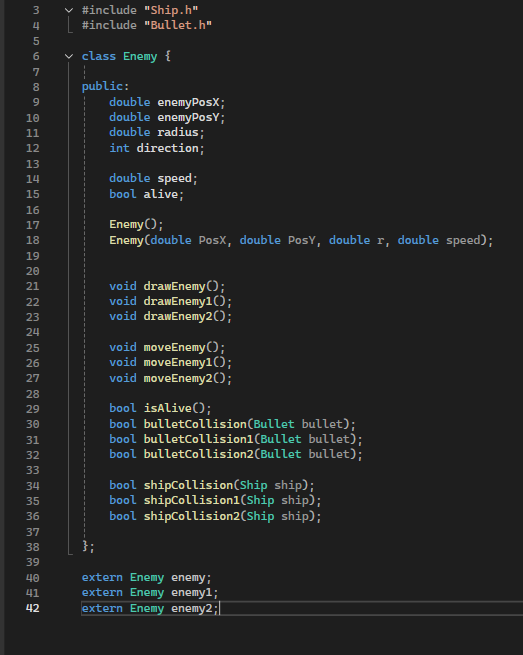
*Slika 17 Enemy.cpp 3*

Klasa Enemy definiše neprijatelje u igri sa atributima kao što su pozicija (enemyPosX, enemyPosY), poluprečnik (radius), brzina (speed), smer kretanja (direction) i status života (alive).

Dva konstruktora služe za inicijalizaciju neprijatelja – jedan postavlja podrazumevane vrednosti, dok drugi prima parametre za kreiranje aktivnog neprijatelja.

Postoje tri funkcije za crtanje neprijatelja različitih boja: drawEnemy, drawEnemy1 i drawEnemy2. Metode moveEnemy, moveEnemy1 i moveEnemy2 pomeraju neprijatelje horizontalno, menjajući smer kada neprijatelj dosegne ivicu ekrana.

Funkcije bulletCollision i shipCollision proveravaju sudar metka i broda sa neprijateljem koristeći njihove pozicije i poluprečnike. Globalne deklaracije extern Enemy enemy, enemy1 i enemy2 omogućavaju korišćenje više neprijatelja u programu.



*Slika 18 Enemy.h*

Klasa Pool služi kao centralno mesto za upravljanje logikom igre, uključujući neprijatelje, nivoe i stanje igre. U konstruktoru se inicijalizuje osnovna konfiguracija, poput nivoa igre i povezanosti sa objektom Legend, koji prati rezultat i živote igrača. Takođe se pokreće generisanje prvog nivoa.

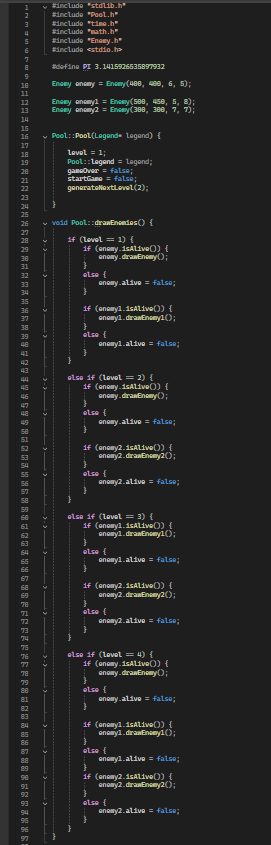
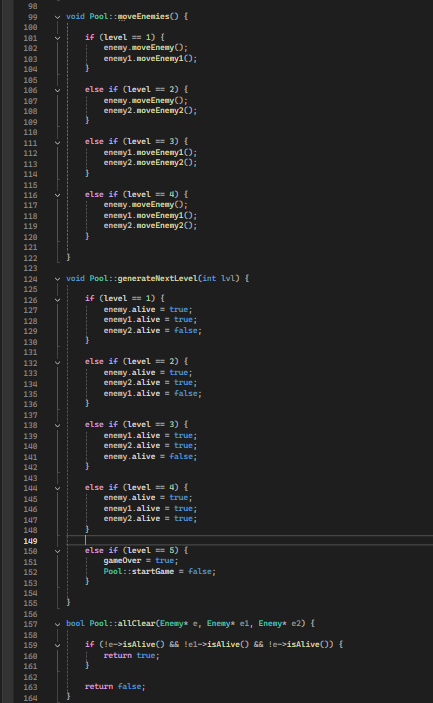
Metoda drawEnemies je odgovorna za crtanje neprijatelja na ekranu. Na osnovu trenutnog nivoa, koristi različite metode za crtanje svakog tipa neprijatelja. Ako su neprijatelji uništeni, njihovo stanje se ažurira tako da nisu više aktivni.

U moveEnemies, neprijatelji se pomeraju horizontalno. Metoda prati pravac kretanja, a kada neprijatelji dostignu ivicu prozora, menjaju pravac. Ovo se ponavlja za sve neprijatelje u zavisnosti od nivoa.

Metoda generateNextLevel upravlja prelaskom na sledeći nivo igre. Aktivira neprijatelje koji će se pojaviti u tom nivou i postavlja njihovo stanje. Ako su svi neprijatelji uništeni, igra prelazi na sledeći nivo i ažurira nivo u Legend klasi.

Metoda BulletCollisionHandler se koristi za upravljanje sudarima između metaka i neprijatelja. Kada dođe do sudara, neprijatelj se označava kao mrtav, udaljuje se iz igre, a rezultat se ažurira povećanjem poena igrača. Ako su svi neprijatelji uništeni, aktivira se prelazak na sledeći nivo, a brod se vraća na početnu poziciju.

Slično, metoda shipCollisionHandler upravlja sudarima između broda i neprijatelja. Kada neprijatelj udari brod, igrač gubi život, a pozicija broda se takođe vraća na početak. Ako igrač ostane bez života, igra se završava. Ako su svi neprijatelji uništeni, prelazi se na sledeći nivo, a nivo u Legend se takođe ažurira. Ova logika omogućava dinamično i interaktivno iskustvo u igri.

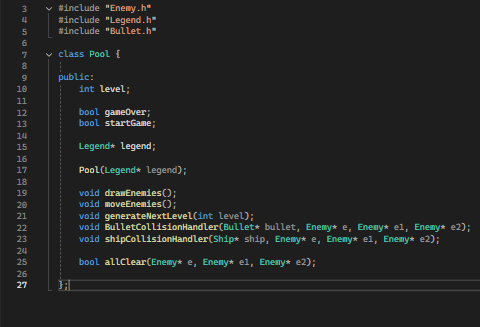
 🡨 *Slika 19 Pool.cpp Slika 20 Pool.cpp 2*

*Slika 21 Pool.cpp* *3*

*Slika 22 Pool.cpp 4*

Klasa Pool upravlja logikom igre, uključujući nivoe, stanje igre i interakcije između objekata. Sadrži članove kao što su trenutni nivo igre, indikatori za kraj igre i početak, kao i pokazivač na objekat Legend koji prati rezultat i živote igrača.

Metode uključuju drawEnemies za crtanje neprijatelja na ekranu, moveEnemies za pomeranje neprijatelja, i generateNextLevel za prelazak na sledeći nivo igre. BulletCollisionHandler i shipCollisionHandler se koriste za upravljanje sudarima između metaka i neprijatelja, odnosno broda i neprijatelja. Metoda allClear proverava da li su svi neprijatelji uništeni, omogućavajući dinamično napredovanje u igri.



*Slika 23 Pool.h*

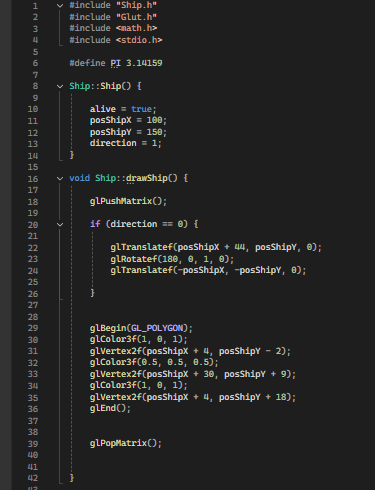
Klasa Ship predstavlja brod u igri, sa više atributa koji definišu njegovo ponašanje i stanje. U konstruktoru se postavljaju početni uslovi: brod je živ, a njegova početna pozicija je na koordinatama (100, 150). Takođe, pravac kretanja se inicijalizuje na desno.

Metoda drawShip koristi OpenGL za crtanje broda. Ova metoda se brine o vizuelnom predstavljanju broda, uključujući njegov oblik i boje. Ako je brod okrenut ulevo, rotira se za 180 stepeni pre nego što se nacrta. Brod se sastoji od poligonalne forme sa različitim bojama za različite delove, što doprinosi njegovoj vizualnoj diferencijaciji.

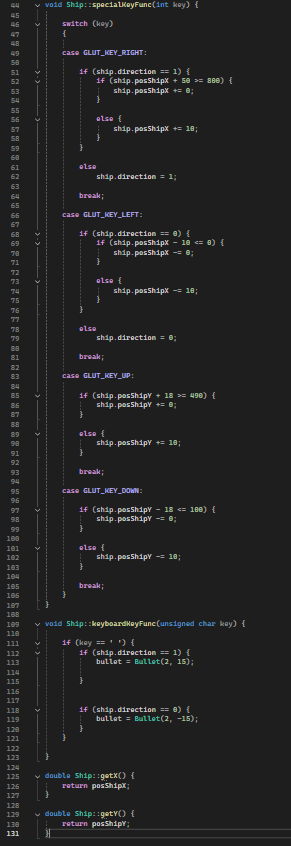
Metoda specialKeyFunc omogućava upravljanje brodom putem strelica na tastaturi. Ova metoda proverava pritisnute tastere i pomera brod u odgovarajućem pravcu. Kada korisnik pritisne desnu strelicu, brod se pomera desno, osim ako ne dostigne desnu ivicu ekrana. Slično važi i za levo, gde se pomera levo, ali ne može preći van leve ivice. Takođe, brod može da se pomera nagore i nadole, pri čemu se proverava da li ostaje unutar verticalnih granica ekrana.

Metoda keyboardKeyFunc omogućava ispaljivanje metaka kada korisnik pritisne razmaknicu. U zavisnosti od trenutnog pravca broda, metak se ispaljuje ili nagore ili nadole. Ova funkcionalnost je ključna za interakciju s neprijateljima i ostvarivanje ciljeva igre.

Na kraju, metode getX i getY vraćaju trenutne koordinate broda, što omogućava drugim delovima programa da pristupaju njegovoj poziciji. Ove informacije su važne za procene sudara, praćenje pozicije u odnosu na neprijatelje ili ciljeve, i druge logike igre.



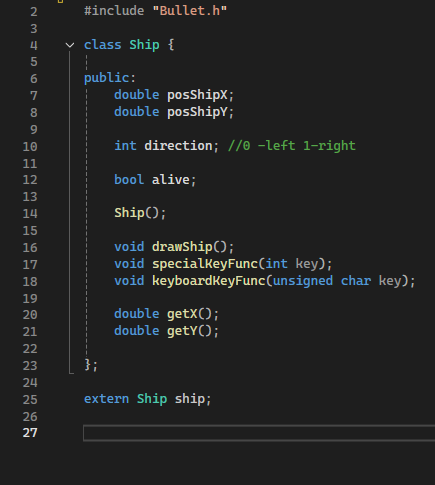
*Slika 24 Ship.cpp*



*Slika 25 Ship.cpp 2*

Klasa Ship predstavlja brod u igri s atributima za poziciju (posShipX, posShipY), pravac (direction), i status (alive). Konstruktor postavlja početne vrednosti.

Metoda drawShip crta brod koristeći OpenGL. specialKeyFunc omogućava upravljanje brodom pomoću strelica, dok keyboardKeyFunc ispaljuje metke pritiskom na razmaknicu. Metode getX i getY vraćaju trenutne koordinate broda, što omogućava drugim delovima igre pristup informacijama o njegovoj poziciji. Klasa je ključna za interakciju igrača i dinamiku igre.



*Slika 26 Ship.h*

Ovaj kod predstavlja glavni deo igre, koristeći OpenGL i GLUT biblioteke za grafiku i korisnički interfejs. Prvo, definiše se nekoliko osnovnih globalnih promenljivih, uključujući igrača (brod), metak, legendu (score i lives), meni i pool neprijatelja.

Funkcija myKeyboardFunc obradi obične tastature, dok mySpecialKeyFunc obrađuje specijalne tastere (kao što su strelice). Ove funkcije se koriste za kontrolu broda i menija.

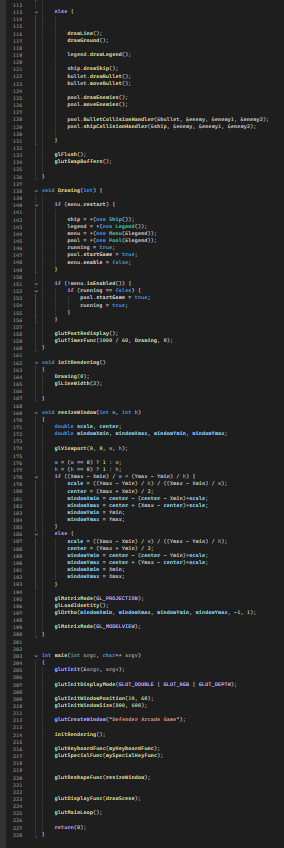
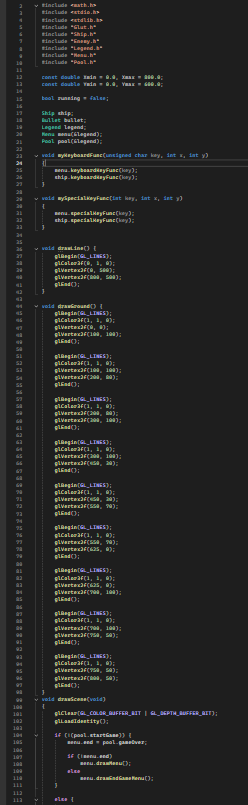
Funkcije drawLine i drawGround crtaju osnovne elemente okruženja igre. drawScene je ključna funkcija koja crta sve vizuelne elemente igre, uključujući meni, brod, metak i neprijatelje, kao i njihove interakcije.

Funkcija Drawing se koristi za ažuriranje stanja igre, uključujući ponovno postavljanje broda i ostalih objekata prilikom ponovnog pokretanja. initRendering postavlja osnovne parametre za renderovanje.

Funkcija resizeWindow prilagođava viewport kada se prozor promeni, čime se osigurava da je sadržaj igre pravilno prikazan.

U main funkciji, inicijalizuje se GLUT, podešavaju se parametri prozora, registruju callback funkcije, i pokreće se glavna petlja igre. Ova struktura omogućava interaktivno igranje i dinamiku u grafici.

4o mini



*Slika 28 Main.cpp 2*

*Slika 27 Main.cpp*

# **Izgled igrice**

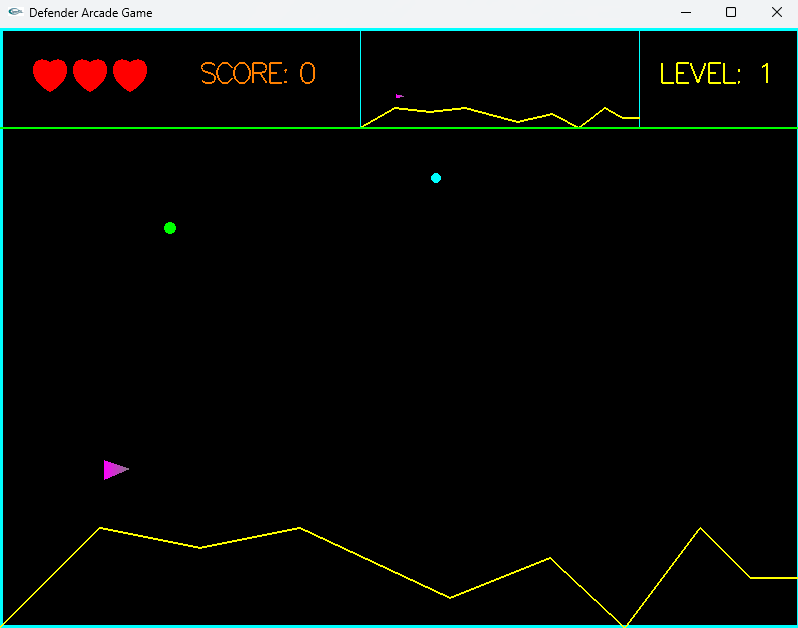
Prilikom pokretanja igrice pojavljuje se početni ekran sa crnom pozadinom na kojoj se nalazi naziv igrice ,,Defender“ i ispod koje se nalaze dva dugmeta plave i narandžaste boje koji predstavljaju dve opcije Start Game i Exit i on izgleda ovako:



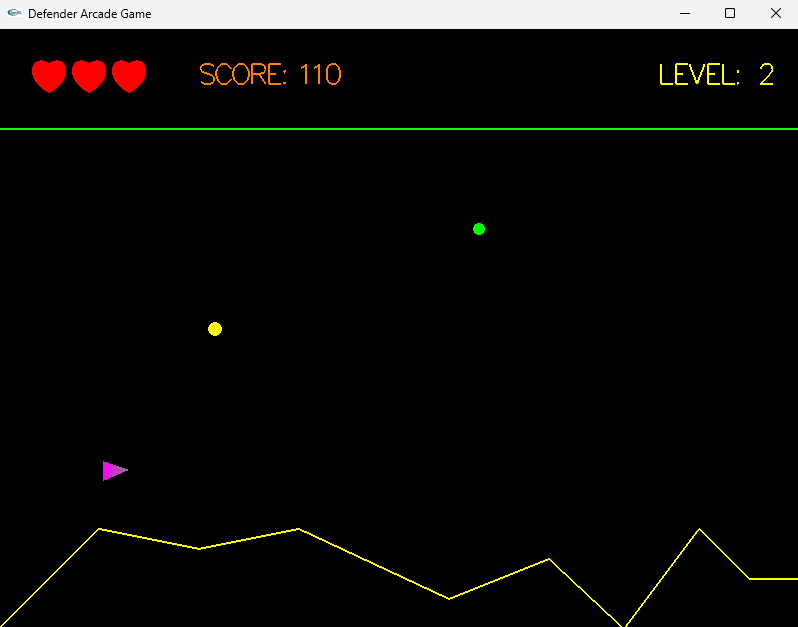
*Slika 29 Početni ekran*

Kada se klikne dugme Exit izlazi se iz aplikacije i ona je ugašena i onda je potrebno ponovo je pokrenuti.

Kada se pokrene igra klikom na Start Game startuje se prvi nivo na kom se na vrhu ekrana prikazuje broj preostalih života, skor odnosno skupljeni poeni i trenutni nivo. Na slici ispod se prikazuje prvi nivo aplikacije. Nivo ima crnu pozadinu sa iscrtanim reljefom na dnu kao mapom i dva protivnika koja se krecu levo desno nezavisno jedan od drugog.

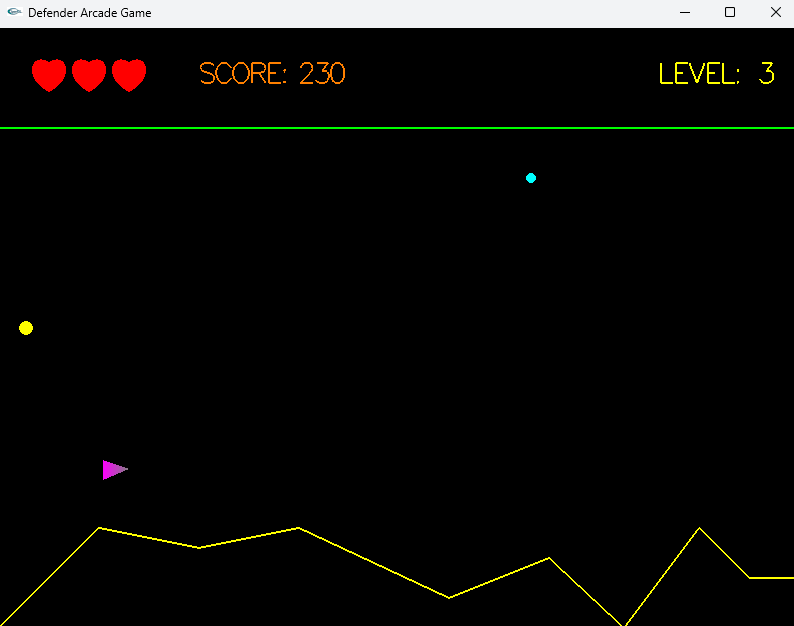


*Slika 30 Prvi nivo*

Nakon što se reši oba čudovišta automatski se prelazi na sledeći nivo broj 2 koji izgleda ovako:  
  


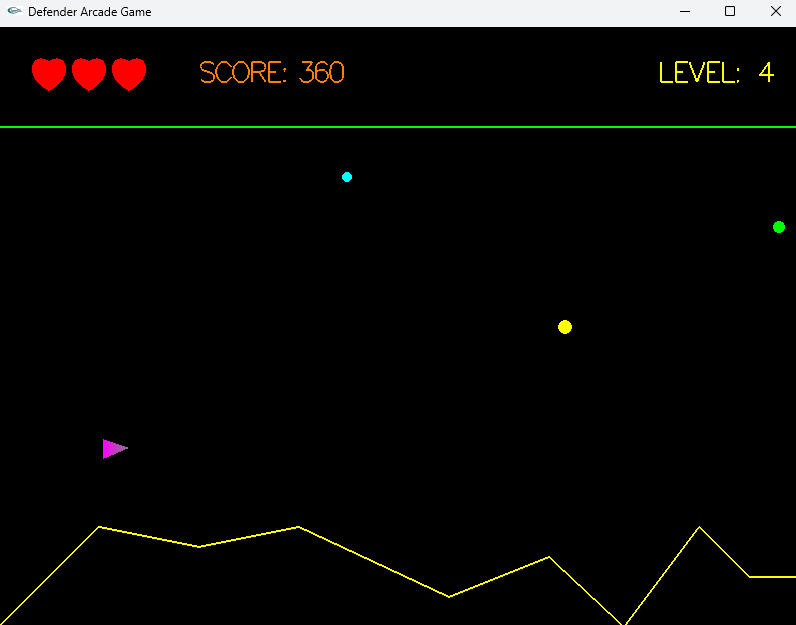
*Slika 31 Drugi nivo*

Drugi nivo ima takođe dva neprijatelja koja se krecu nezavisno jedan od drgog različitim brzinama, potrebno je pogoditi oba kako presao na sledeci nivo. Cim se poslednji neprijatelj ubije automatski se prelazi na sledeci nivo.



*Slika 32 Treći nivo*

Treći nivo sadrži dva čudovišta koja se krecu za nijansu brže nego u prethodnim nivou. I treba se naglasiti da ubijanjem svakog različitog čudivišta dobija se različiti broj poena za zelene – 50, plave – 60, žute 70.



*Slika 33 Četvrti nivo*

U četvrtom nivou se nalaze tri čudovišta koja se kreću različitim brzinama. Cilj nivoa je kao u prethodnim da se ubiju sva tri. Ukoliko čudovište pogodi svemirski brod automatski se gubi jedan život, a nivo se nastavlja dalje.

Posle pređenog četvrtog nivoa automatski izlazi ekran game over na kojem piše stečeni broj poena sa dve opcije plaz again i exit. Ukoliko igrač pritisne na play again igrica se ponovo startuje na prvom nivou. Pritiskom na dugme exit igrica se gasi i potrebno je ponovno pokretanje.



# **Zaključak**

Razvoj igrice "Defender" iz 1981. godine korišćenjem OpenGL-a pruža odličnu priliku za učenje osnova grafičkog programiranja i rad sa 2D okruženjem. OpenGL omogućava programerima da implementiraju sve karakteristične elemente igre, kao što su horizontalno skrolovanje, jednostavne spriteove i kontrolu nad neprijateljima u realnom vremenu. Iako danas postoje mnogo napredniji alati za razvoj igara, kao što su Unity i Unreal Engine, korišćenje OpenGL-a je i dalje odličan način da se razumeju osnovni principi grafičke obrade i animacije. Budući da je "Defender" jedna od pionirskih arkadnih igara sa jednostavnom grafikom i mehanikama, njena rekonstrukcija pomoću OpenGL-a nije previše zahtevna, ali pruža dragoceno iskustvo u radu sa osnovama game development-a.

# **Literatura**

* Moodle: Računarska grafika.

<http://moodle.fin.kg.ac.rs/course/view.php?id=988>

* Wikipedia: Defender 1981.

<https://en.wikipedia.org/wiki/Defender_(1981_video_game)>

* OpenGL.

<https://www.opengl.org/Documentation/Specs.html>