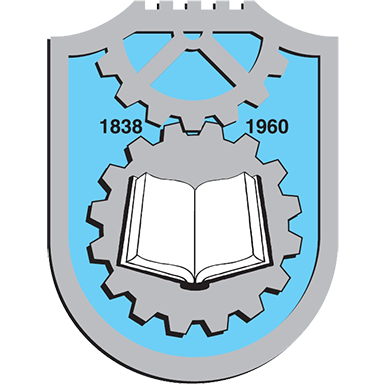
Univerzitet u Kragujevcu

Fakultet inženjerskih nauka



Računarska grafika

Projektni zadatak:

Izrada arkadne igrice ,,Bubble Bobble"

korišćenjem OpenGL biblioteke

Student: Predmetni profesor:

Mihajlo Janković 616/2019 Nenad Filipović

Tijana Šušteršič

Kragujevac 2024.

**Sadržaj**

[**1.** **Uvod** 3](#_Toc175253695)

[**2.** **Korisničko uputsvo** 4](#_Toc175253696)

[**3.** **Objašnjenje koda i izrada aplikacije** 5](#_Toc175253697)

[**4.** **Izgled igrice** 20](#_Toc175253698)

[**5.** **Zaključak** 23](#_Toc175253699)

[**6.** **Literatura** 24](#_Toc175253700)

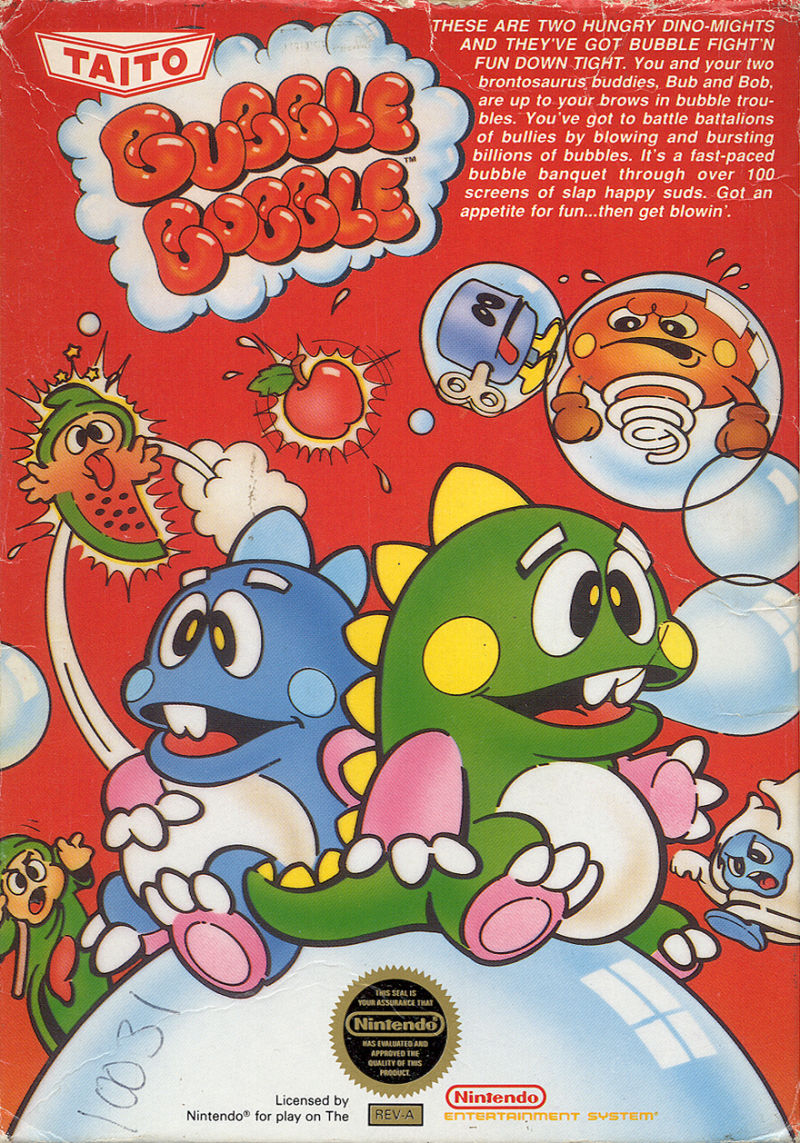
# **Uvod**

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

"Bubble Bobble" je arkadna video igra koja je prvobitno objavljena 1986. godine od strane Taito Corporation. Igra je kasnije doživela brojne portove na različite platforme, uključujući Nintendo Entertainment System (NES), Sega Master System, i Commodore 64, postajući klasik među ljubiteljima arkadnih igara.

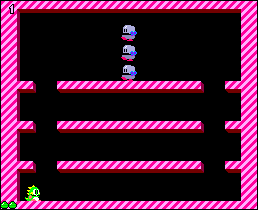
Radnja igre "Bubble Bobble" prati dva mala zmaja, Bobblea i Bobbla, koji moraju da spasavaju svoje devojke zarobljene u misterioznom pećinskom sistemu. Igrači preuzimaju kontrolu nad ovim zmajevima, koji imaju jedinstvenu sposobnost da stvaraju mehuriće. Cilj igre je jednostavan – zarobiti neprijatelje u mehuriće, a zatim ih uništiti. Nakon što su svi neprijatelji na ekranu eliminisani, igra prelazi na sledeći nivo.

Igra kombinuje brzu akciju i strategiju, jer igrači moraju pažljivo planirati kako da zarobe neprijatelje i izbegnu opasnosti. Svaki nivo je jedinstveno dizajniran sa različitim rasporedom platformi i prepreka, što igračima pruža nove izazove na svakom koraku.



*Slika 1 – Jedna od prvih naslovnih strana iz1986.*

Na sledećoj slici je prikazan izgled prvog nivoa igrice Bubble Bobble:



*Slika 2 – Prvi nivo igrice Bubble Bobble*

# **Korisničko uputsvo**

Projekat je izrađen po uzoru na "Bubble Bobble" gde glavni igrač upravlja zelenim kvadratom umesto zelenim zmajem. Osnovni zadatak igrača je da zarobi neprijatelje u mehuriće tj. u plave krugove i da ih uništi kako bi osvojio što više poena i prešao na sledeći nivo. Svaki nivo ima različit raspored prepreka i neprijatelja, a igrač mora pažljivo da planira svoje pokrete i strategiju kako bi napredovao kroz igru.

Glavni igrač koji je predstavljen zelenim kvadratom može da se kreće levo, desno i gore korišćenjem tastera na tastaturi A, D i W respektivno, kao i mogućnost ispaljivanja mehurića odnsono kružića plave boje klikom na taster K. Mehurići mogu da zarobe neprijatelje, a igrač zatim mora da skoči na njih kako bi ih eliminisao i osvojio dodatne poene.

Igrica se sastoji iz tri nivoa, gde svaki nivo donosi nove izazove u vidu različitih neprijatelja i složenijih prepreka. Prvi nivo je lakši, sa dva neprijateljska čudovišta koja su predstavljane kvadratima crvene boje i jednostavnijim rasporedom platformi, dok kasniji nivoi postaju sve teži sa dodatnim neprijateljskim čudovištem i komplikovanijim rasporedom prepreka. Prva dva čudovišta se kreću najsporije, na drugom nivou se pojavljuje novo koje je brže, a na trećem nivou i četvrto koje je najbrže. Njihov cilj je stići do glavnog heroja i kada ga uhvati automatski se gubi život.

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 restartuje se nivo a gubljenjem sva tri, igra se završava i vraća se na početni ekran.

# **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.

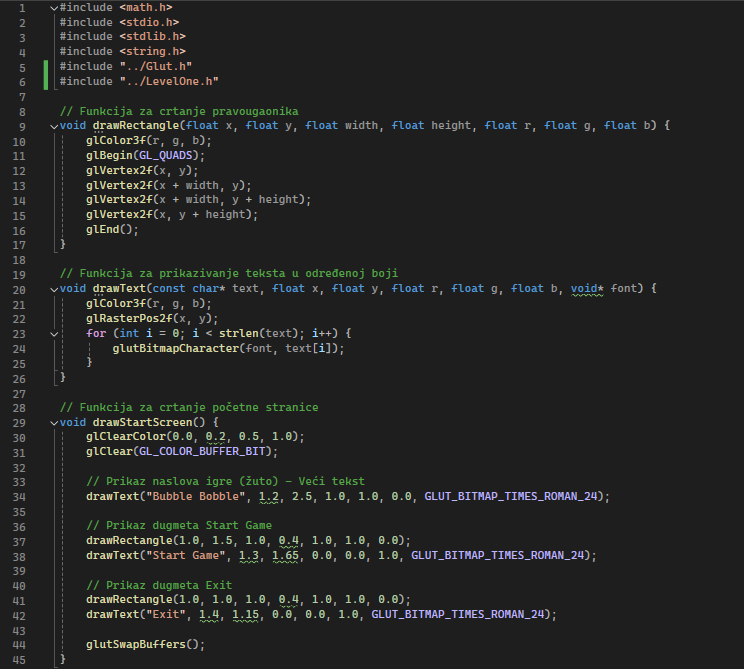
Početni(prvi) ekran koji se prikazuje pri pokretanju igrice:



*Slika 3 – Početni ekran*

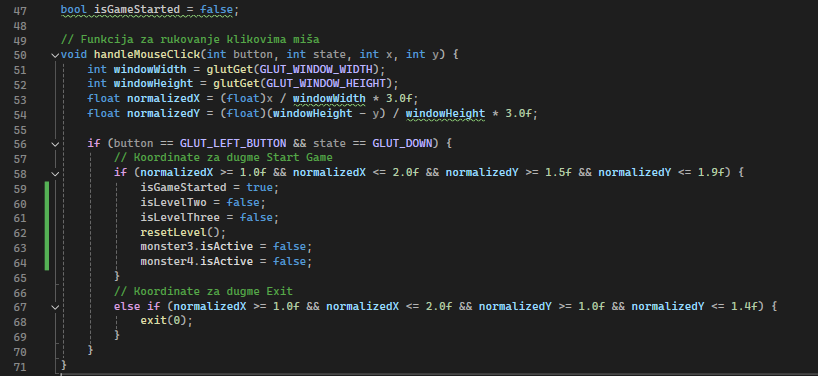
Kod koji se nalazi u fajlovima StartScreen.cpp i StartScreen.h odgovoran je za prikazivanje i upravljanje početnim ekranom igre "Bubble Bobble". Početni ekran je deo korisničkog interfejsa gde igrač može započeti igru ili izaći iz aplikacije. Ovaj kod obuhvata nekoliko osnovnih funkcionalnosti, uključujući 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 i funkcije za crtanje pravougaonika koji predstavljaju dugmiće, zatim funkcija za ispis teksta na ekranu(naslov i tekst ispisan na dugmićima) i na kraju funkcija koja je odgovorna za crtanje celokupnog početnog ekrana igre. Prvo, poziva glClearColor kako bi postavila boju pozadine ekrana, a zatim koristi glClear za brisanje prethodnog sadržaja. Nakon toga, funkcija crta naslov igre, dugme za početak igre ("Start Game") i dugme za izlazak ("Exit"). Tekst i pravougaonici su pozicionirani na ekranu korišćenjem prethodno opisanih drawText i drawRectangle funkcija.



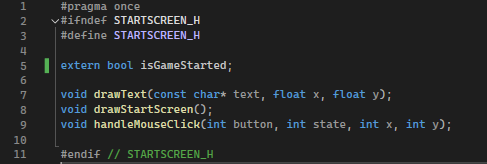
*Slika 4 – Fajl StartScreen.cpp prvi deo*

Nakon toga se definiše promenljiva isGameStarted i postavlja na false, i ona označava da li je igrica pokrenuto odnosno da li je pokrenut prvi nivo a zatim je definisana i funkcija handleMouseClick. Ova funkcija detektuje klikove miša i reaguje na njih. Koristi se da prepozna kada igrač klikne na dugme "Start Game" ili "Exit". Funkcija prvo dobija širinu i visinu prozora kako bi normalizovala koordinate miša u odnosu na ekran. Zatim proverava da li je levi taster miša kliknut (GLUT\_LEFT\_BUTTON) i da li je taster pritisnut (GLUT\_DOWN). Ako su koordinate miša unutar granica dugmeta "Start Game", igra se pokreće (isGameStarted = true), dok ako su unutar granica dugmeta "Exit", aplikacija se zatvara (exit(0)).



*Slika 5 – Fajl StartScreen.cpp drugi deo*

Fajl startscreen.h je zaglavlje koje sadrži deklaracije funkcija koje se koriste za crtanje početnog ekrana i rukovanje klikovima miša. Takođe sadrži deklaraciju globalne promenljive isGameStarted koja se koristi za praćenje stanja igre (da li je igra započeta). Fajl koristi #pragma once kako bi se osigurala jedinstvena inkluzija tokom kompajliranja.



*Slika 6 – Fajl StartScreen.h*

Kod koji se nalazi u fajlovima LevelOne.cpp i LevelOne.h odgovoran je za svu logiku igrice i prikaz sva tri nivoa.

Na početku koda su definisane biblioteke koriščćene za izradu ovog projekta, a nakon njih i struktura Character koja inicijalizuje heroja i 4 čudovišta sa parametrima pozicije dimenzija i stanjima da li su pogođeni, zaustavljeni i aktivni. Nakon toga konstante fizike kretanja karaktera u igrici kao sto su:

 **HERO\_STEP** određuje koliko se heroj pomera pri svakoj akciji (leva/desna strelica).

 **JUMP\_HEIGHT** i **JUMP\_VELOCITY** regulišu visinu i brzinu skoka heroja.

 **GRAVITY** je konstanta koja utiče na kretanje heroja prema dole kada nije u skoku.

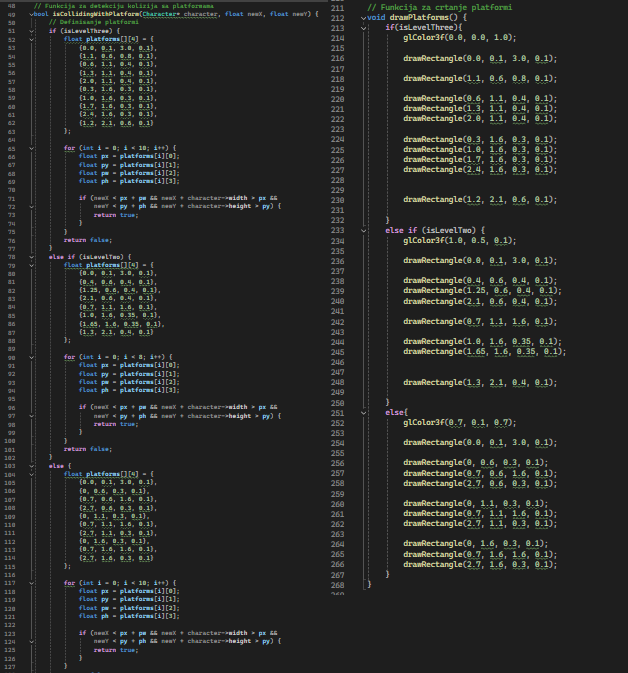
 **PROJECTILE\_SPEED** i **PROJECTILE\_DISTANCE** definišu brzinu i udaljenost koje projektil može da pređe.

 **MONSTER1\_SPEED, MONSTER2\_SPEED, MONSTER3\_SPEED, MONSTER4\_SPEED** postavljaju brzinu kretanja svakog čudovišta u odnosu na HERO\_STEP.

Ostale varijable predstavljaju stanja kretanja heroja kao i br. života i broj poena prikupljenih u igri. Funkcija render BitmapString služi za ispisivanje teksta na ekranu u igri.

  
*Slika 7 – LevelOne.cpp prvi deo*

Sledeći deo koda, funkcija isCollidingWithPlatform služi za detekciju kolizije bilo kog karaktera sa platformama sastoji se iz tri dela jer proverava za svaki nivo posebno jer se na sva tri nivoa nalaze različite platforme. Sa desne strane na slici je kod za iscrtavanje tih paltformi.

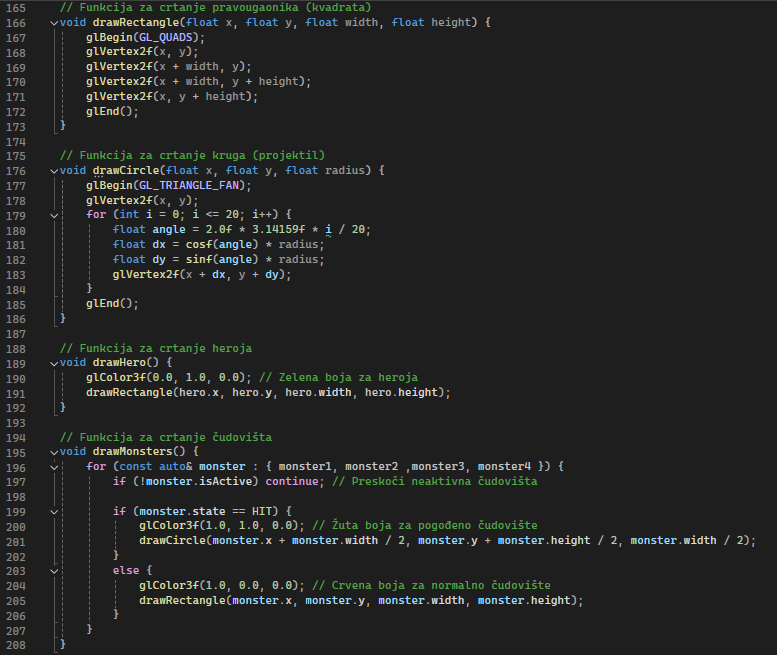


*Slika 8 – LevelOne.cpp drugi deo*

Slika ispod prikazuje sledeću funkciju u kodu koja definiše tastere upravljanja herojem i omogućava njego kretanje levo i desno kao i skok i pucanje:  


*Slika 9 – LevelOne.cpp treći deo*

U nastavku se definiše funkcija za crtanje pravougaonika drawRectangle, nakon nje funkcija za crtanje kruga. Korišćenjem te dve funkcije se nakon toga crta heroj drawHero kao zeleni kvadrat i čudovište kao crveni kvadrat ili žuti krug u slučaju da je pogođeno. Ako čudovište nije aktivno onda ne dolazi do crtanja tog čudovišta i njega nema na ekranu.



*Slika 10 – LevelOne.cpp četvrti deo*

Sledeća funkcija drawLevelOneScene() je odgovorna za iscrtavanje scena različitih nivoa igre, uz prikaz trenutnog nivoa, rezultata i broja života igrača. Funkcija sadrži tri glavna dela, svaki za po jedan nivo igre.

**Struktura funkcije:**

1. **Iscrtavanje za nivo 3 (if (isLevelThree))**
   * **Pozadina:** Boja pozadine postavlja se na tamno plavu (glClearColor(0.0, 0.0, 0.1, 1.0)).
   * **Elementi na ekranu:** Poziva se niz funkcija da se iscrtaju različiti elementi scene:
     + drawPlatforms(); za platforme.
     + drawHero(); za heroja.
     + drawMonsters(); za čudovišta.
     + drawProjectiles(); za projektile.
   * **Tekst:** Prikazuje nivo, rezultat i broj života:
     + "LEVEL: 3" prikazuje trenutni nivo.
     + sprintf\_s(scoreText, "SCORE: %d", score); formira string za prikaz trenutnog rezultata.
     + sprintf\_s(livesText, "LIVES: %d", lives); formira string za prikaz broja života.
   * **Renderovanje i osvežavanje:** glutSwapBuffers(); se koristi za osvežavanje ekrana.
2. **Iscrtavanje za nivo 2 (else if (isLevelTwo))**
   * **Pozadina:** Boja pozadine je drugačija, postavljena na tamno narandžastu (glClearColor(0.2, 0.1, 0.02, 1.0)).
   * **Elementi na ekranu:** Sličan postupak kao za nivo 3, s time što se prikazuje "LEVEL: 2".
3. **Iscrtavanje za nivo 1 (else)**
   * **Pozadina:** Boja pozadine je tamno ljubičasta (glClearColor(0.1, 0.01, 0.1, 1.0)).
   * **Elementi na ekranu:** Sličan postupak kao za prethodne nivoe, s time što se prikazuje "LEVEL: 1".



*Slika 11 – LevelOne.cpp peti deo*

**Funkcija updateHero()**

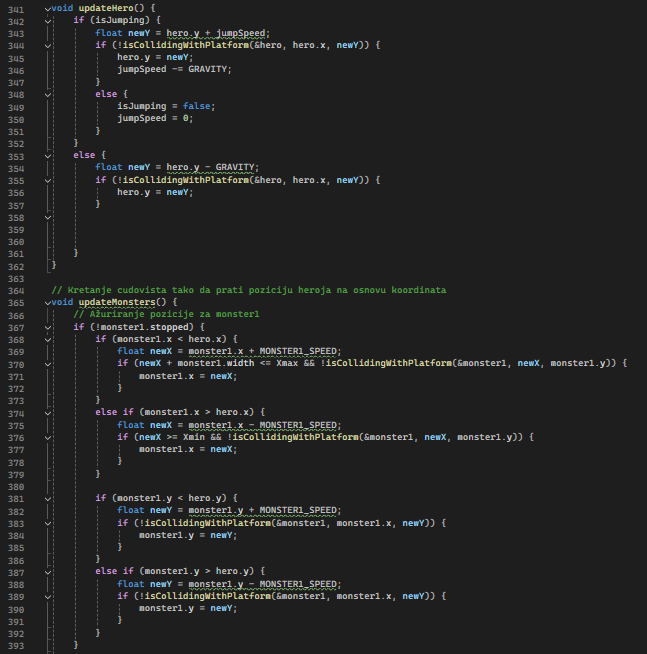
Ova funkcija ažurira poziciju heroja u igri, obrađujući kretanje heroja u vertikalnom pravcu (skakanje i padanje) i interakciju s platformama.

1. **Provera skakanja (if (isJumping))**
   * + **Računanje nove visine:** Visina heroja se ažurira pomoću trenutne brzine skakanja (jumpSpeed).
     + **Provera sudara:** Proverava se da li nova pozicija heroja kolidira s platformom koristeći isCollidingWithPlatform.
     + **Ažuriranje pozicije:** Ako nema sudara, heroj nastavlja da se pomera prema gore, a brzina skakanja se smanjuje za vrednost gravitacije (GRAVITY).
     + **Zaustavljanje skakanja:** Ako heroj udari u platformu, skakanje se zaustavlja, a brzina se resetuje na 0.
2. **Kretanje nadole kada heroj nije u skoku**
   * + **Simulacija gravitacije:** Visina heroja se smanjuje simulirajući gravitaciju.
     + **Provera sudara:** Ako nema sudara s platformom, heroj se spušta.

**Funkcija updateMonsters()**

Ova funkcija ažurira pozicije svih čudovišta u igri kako bi pratili heroja, uzimajući u obzir njihove individualne brzine i provere sudara sa platformama.

1. **Ažuriranje za svako čudovište (monster1, monster2, monster3, monster4)**
   * **Horizontalno kretanje:**
     + Čudovište se pomera prema ili od heroja zavisno od njihove trenutne horizontalne pozicije.
     + Nova pozicija se računa i proverava da li dolazi do sudara s platformom.
     + Ako nema sudara, čudovište ažurira svoju horizontalnu poziciju.
   * **Vertikalno kretanje:**
     + Slično kao i za horizontalno kretanje, čudovište se pomera prema gore ili dole, zavisno od toga da li je heroj iznad ili ispod njega.
     + Nova vertikalna pozicija se proverava i, ako nema sudara, čudovište se pomera.



*Slika 12 – LevelOne.cpp šesti deo*

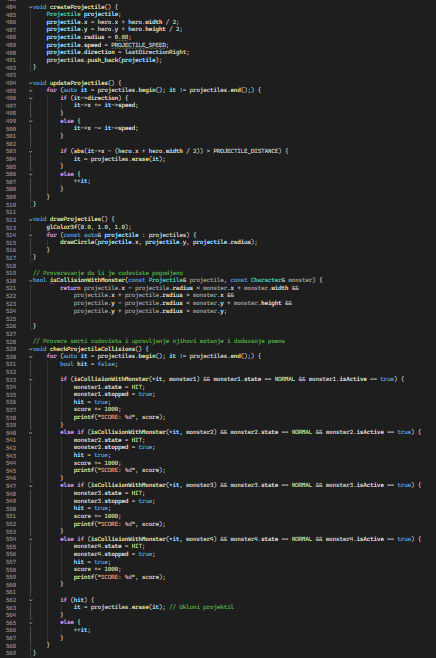
Funkcija createProjectile() kreira novi projektil u trenutku kada heroj ispaljuje hitac. Projektil se postavlja na centar heroja, a njegov pravac zavisi od poslednjeg smera kretanja heroja (lastDirectionRight). Novi projektil se zatim dodaje u listu projectiles.

Funkcija updateProjectiles() ažurira položaj svih projektila u igri. Svaki projektil se pomera levo ili desno, zavisno od njegovog smera. Ako projektil pređe određenu udaljenost od mesta ispaljivanja (PROJECTILE\_DISTANCE), uklanja se iz liste.

Funkcija drawProjectiles() crta sve aktivne projektile na ekranu koristeći funkciju drawCircle. Projektili su obojeni u svetloplavu boju.

Funkcija isCollisionWithMonster() proverava da li je došlo do sudara između projektila i određenog čudovišta. Sudar se proverava upoređivanjem pozicija projektila i čudovišta.

Funkcija checkProjectileCollisions() proverava da li je neki od projektila pogodio neko od čudovišta. Ako projektil pogodi čudovište koje je u normalnom stanju (NORMAL) i aktivno (isActive), stanje čudovišta se menja u pogođeno (HIT), čudovište se zaustavlja (stopped postaje true), a igrač dobija 1000 poena. Projektil koji je pogodio čudovište se zatim uklanja iz liste.



*Slika 13 – LevelOne.cpp sedmi deo*

Poslednje tri funkciju u LevelOne.cpp fajlu su:

Funkcija **void resetLevel()** vraća sve elemente igre na početno stanje kad se nivo resetuje:

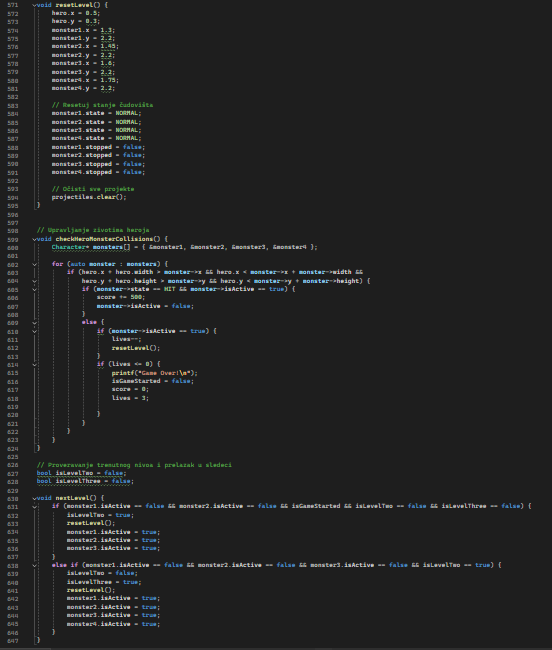
1. **Pozicije Heroja i Čudovišta:**
   * Pozicija heroja (hero.x i hero.y) se postavlja na (0.5, 0.3).
   * Pozicije čudovišta (od monster1 do monster4) se postavljaju na različite koordinate u igri.
2. **Resetovanje Stanja Čudovišta:**
   * Stanje svih čudovišta (monster1, monster2, monster3, monster4) se postavlja na NORMAL, što označava da nisu pogođeni.
   * Sva čudovišta se postavljaju na aktivna i nestopirana (stopped = false).
3. **Čišćenje Projektila:**
   * Svi projektili koji su trenutno u igri se brišu pomoću projectiles.clear(), čime se lista projektila prazni.

Funkcija **void checkHeroMonsterCollisions()** proverava da li je došlo do sudara između heroja i čudovišta i obavlja potrebne akcije:

1. **Provera Sudara:**
   * U ovoj funkciji se koristi niz monsters koji sadrži pokazivače na čudovišta.
   * Proverava se da li su koordinate heroja i čudovišta u preklapanju.
2. **Akcije pri Sudaru:**
   * Ako je čudovište pogođeno (state == HIT), igrač dobija 500 poena, a čudovište se označava kao neaktivno (isActive = false).
   * Ako čudovište nije pogođeno, a i dalje je aktivno, igrač gubi jedan život (lives--) i nivo se resetuje koristeći resetLevel().
3. **Kraj Igre:**
   * Ako broj života (lives) padne na 0 ili ispod, igra se završava. i postavke igre se vraćaju na početne vrednosti (score=0 i lives =3).

Funkcija **void nextLevel()** upravlja prelaskom između nivoa u igri:

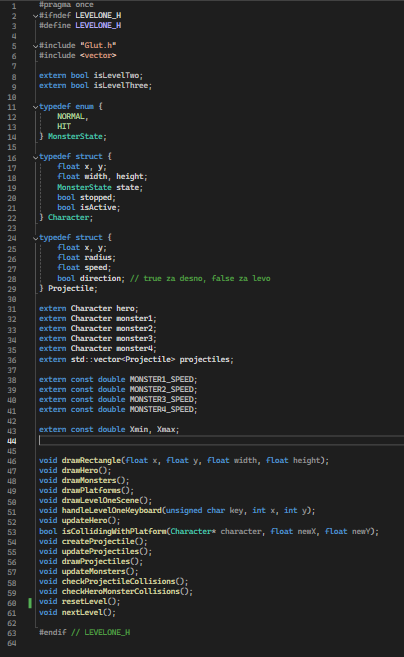
1. **Prelazak na Drugi Nivo:**
   * Ako su monster1 i monster2 neaktivni i nivo je aktivan, a isLevelTwo i isLevelThree su oba false, igra prelazi na drugi nivo (isLevelTwo = true).
   * Pozicija svih čudovišta se ponovo postavlja i čudovišta se postavljaju na aktivna (isActive = true).
2. **Prelazak na Treći Nivo:**
   * Ako su monster1, monster2, i monster3 neaktivni i trenutno je drugi nivo aktivan (isLevelTwo = true), prelazi se na treći nivo (isLevelThree = true).
   * Nivo se resetuje, postavljaju se čudovišta na aktivna, i dodaje se četvrto čudovište



*Slika 14 – LevelOne.cpp osmi deo*

Fajl LevelOne.h je zaglavlje odnosno header fajl koje definiše sve ključne komponente i funkcije vezane za LevelOne.cpp fajl. Sadrži direktive koje obezbeđuju da se sadržaj fajla uključuje samo jednom prilikom kompilacije, sprečavajući dupliranje definicija i greške koje mogu nastati zbog više uključivanja istog fajla(#pragma once i #ifndef LEVELONE\_H … #endif).

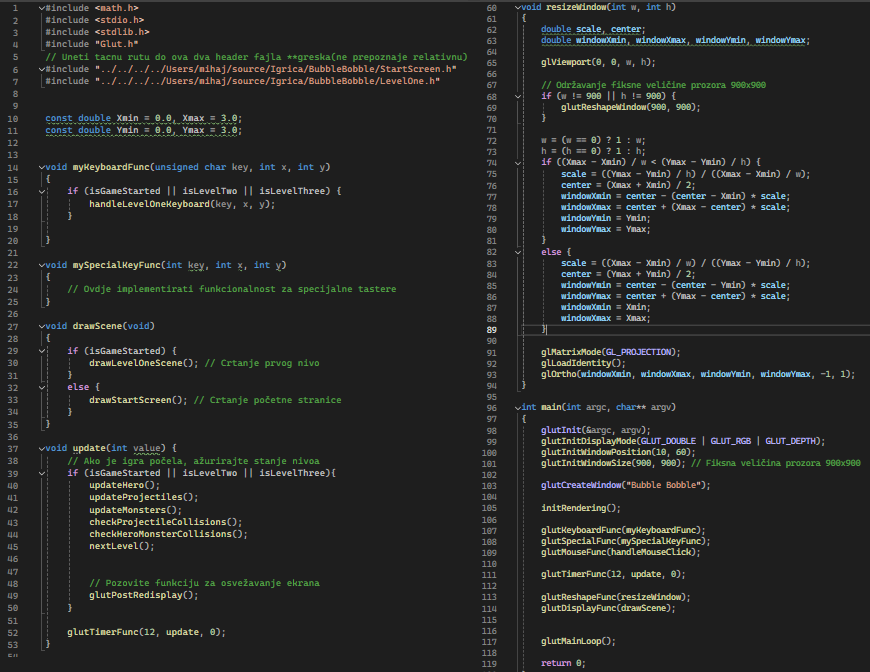
Nakon toga se uključe potrebne bibilioteke, a to su Glut.h i vector koja se koristi za rad sa dinamičkim nizovima. Zatim se definišu globalne promenljive za trenutni nivo i enumeracija koja definiše stanje čudovišta, struktura koja opisuje karaktere i nakon nje struktura za projektile odnosno mehuriće koje ispaljuje heroj. Nakon toga idu redom globalne promenjlive, a nakon njih i funkcije kako bi se sve to upotrebilo u drugom fajlu radi pozivanja ili izmena.



*Slika 15 – LevelOne.h*

Poslednji fajl je izvršni fajl main.cpp gde je potrebno pozvati sve funkcionalnosti kako bi se inicijalizovale u aplikaciji.

Prvo se pozivaju sve biblioteke, a nakon njih se inicijalizuju promenjive za proporcije ekrana. Prva funkcija je za omogućavanje već definisanog rada tastera za upravljanje herojem, a nakon toga se iscrtava scena odnosno ekran početni ili ekran za prikaz nivoa. Funkcija update() ažurira stanje igre, heroja, projektila, čudovišta, kolizija kao i trenutnog nivoa. Funkcija resizeWindow() podešava veličinu prozora na fiksnu dimenziju 900x900 piksela, a int main() je glavna funkcija koja inicijalizuje GLUT, postavlja prozor i funkcije za upravljanje događajima.



*Slika 16 – main.cpp fajl*

# **Izgled igrice**

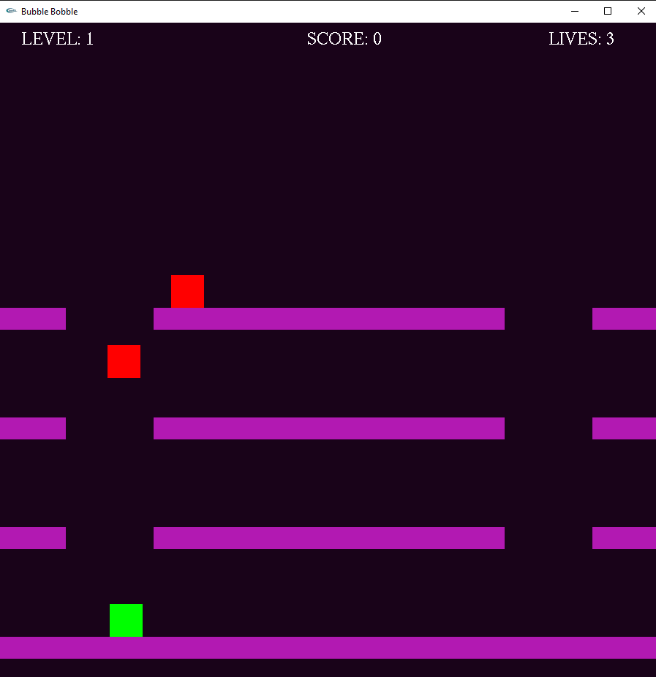
Prilikom pokretanja igrice pojavljuje se početni ekran sa plavom pozadinom na kojoj se nalazi naziv igrice ,,Bubble Bobble“ i ispod koje se nalaze dva dugmeta žute boje koji predstavljaju dve opcije Start Game i Exit i on izgleda ovako:



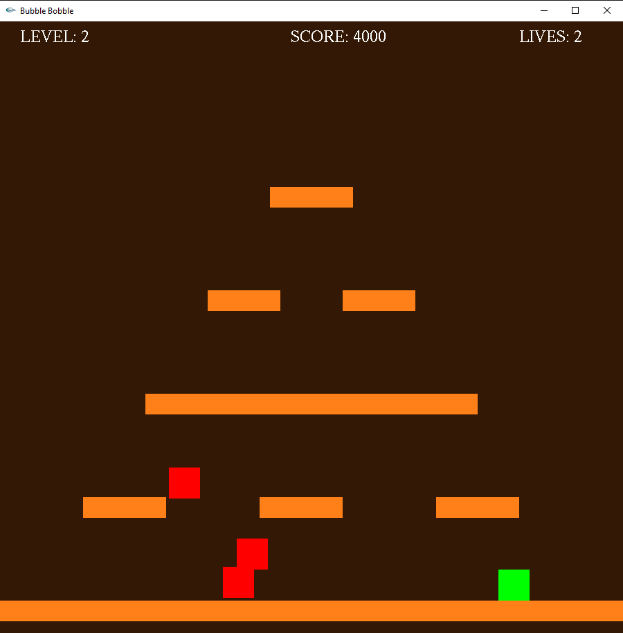
*Slika 17 – Početni ekran*

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 trenutni nivo, skor odnosno skupljeni poeni i broj preostalih života heroja. Na slici ispod se prikazuje prvi nivo aplikacije. Nivo je obojen u nijansama roze odnosno ljubičaste boje sa jedinstvenim rasporedom platformi. Heroj se nalazi inicijalno na najnižoj dok se čudovišta pojavljuju na vrhu. Na ovom nivou su samo dva čudovišta(dva kvadrata crvene boje) od kojih je jedno brže i cilj im je da uhvate glavnog heroja(kvadrat zelene boje).



*Slika 18 – Prvi nivo*

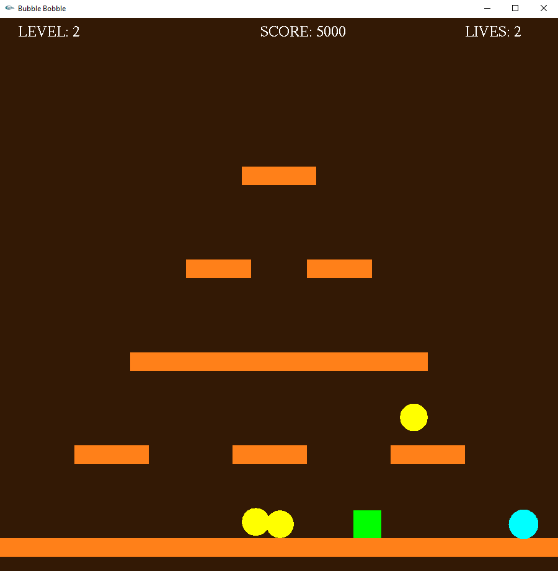
Nakon što se reši oba čudovišta automatski se prelazi na sledeći nivo broj 2 koji izgleda ovako:  
  


*Slika 19 – Drugi nivo*

Kao što se vidi i na slici drugi nivo je u nijansama braon odnosno narandžaste boje sa novim rasporedom platformi i dodatnim čudovištem koje je brže od prethodna dva, tako da se na ovom nivou na poslednjoj platformi stvaraju 3 čudovišta tj. crvena kvadrata koja jure glavnog heroja.

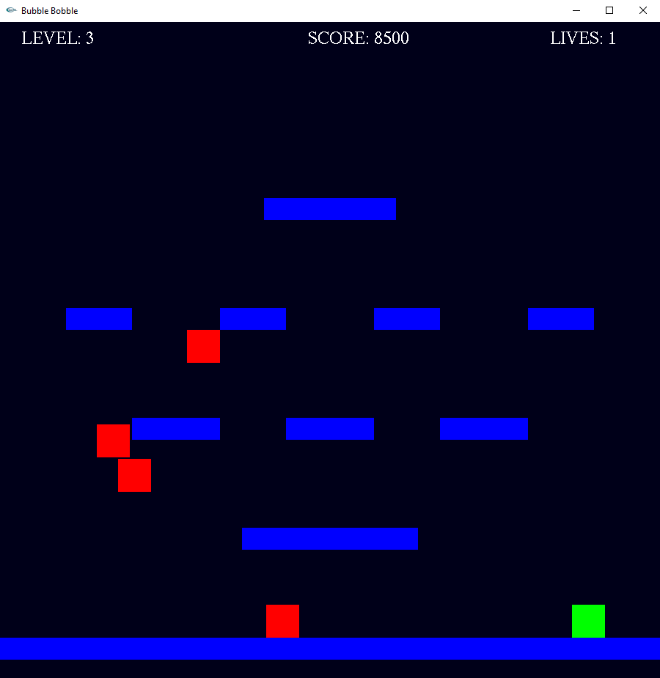
Da bi se prešao nivo potrebno je da heroj odnosno zeleni kvadrat pogodi svojim projektilima odnosno mehurićima sva čudovišta nakon čega se ona pretvaraju u žuti krug. Zelenim kvadratom je potrebno pokupiti i sve žute krugove nakon što se pogode sva čudovišta. Poeni se dobijaju i za pogodatk(1000 poena) i za skupljanje žutih krugova(500 poena).

Na slici ispod se vidi kako izgledaju sva tri čudovišta nakon što ih pogodi projektil koji je na slici krug plave boje i ispaljuje ga zeleni kvadrat:



*Slika 20 – Drugi nivo objašnjenje*

Nakon što pokupi svaki žuti krug koji je ostao od svih ubijenih čudovišta automatski se prelazi na sledeći nivo. Poslednji nivo je u nijansama plave boje sa novim rasporedom platformi i sa još jednim dodatnim čudovištem koje je još brže od prethodna 3, tako da ih na ovom nivou ima 4. Ovaj treći nivo je poslednji i nakon njega heroj ostaje sam na ekranu sa ispisanim brojem poena i preostalim životima.



*Slika 21 – Treći nivo*

Kada čudovište uhvati heroja on gubi život. Kada se započne igrica on ima tri život i potrebno je ne izgubiti više od dva života do kraja, ali ako dođe do toga da igrač izgubi sva tri života pre nego što pređe celu igricu, onda se on vraća na početni ekran i ima izbor da počne opet od prvog nivoa ili da izađe iz igrice.

# **Zaključak**

Razvoj igrice "Bubble Bobble" korišćenjem OpenGL-a predstavlja izazovan, ali veoma edukativan proces koji omogućava programerima da se upoznaju sa osnovama grafičkog programiranja. OpenGL, kao moćan alat za kreiranje grafike, pruža širok spektar mogućnosti za pravljenje aplikacije ovog tipa. Iako je OpenGL i dalje moćan alat za grafičko programiranje, razvoj igara danas je značajno evoluirao, sa modernim alatima i tehnologijama koje omogućavaju brži, jednostavniji i vizuelno impresivniji razvoj igara,ali s obzirom da je ova igrica stara oko 40 godina, nije bilo preterano teško napraviti igricu po uzoru na ,,Bubble Bobble“.

# **Literatura**

* Moodle: Računarska grafika, 10.08.2024.

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

* Wikipedia: Bubble Bobble, 10.08.2024.

<https://en.wikipedia.org/wiki/Bubble_Bobble>

* OpenGL, 10.08.2024.

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