

*Prolećni semestar, 2024/25*

*PREDMET: IT230 INTERAKCIJA ČOVEK-RAČUNAR*

Projektni zadatak:

**Tetris igrica**

Ime i prezime: **Andrija Milovanović**

Broj indeksa: **6310**

Datum izrade: **14/09/2025**

# Sadržaj

Sadržaj

[Sadržaj 2](#_Toc208793183)

[Uvod 2](#_Toc208793184)

[Analiza 2](#_Toc208793185)

[Cilj programa 2](#_Toc208793186)

[Elementi programa 3](#_Toc208793187)

[Glavne funkcije za rad sa matricama 3](#_Toc208793188)

[Pomoćne funkcije 3](#_Toc208793189)

[Izrada programa 3](#_Toc208793190)

[Funkcionisanje programa 4](#_Toc208793191)

[Zaključak 7](#_Toc208793192)

[Literatura 7](#_Toc208793193)

# Uvod

Ovaj dokument predstavlja tehničku dokumentaciju kompletne implementacije klasične Tetris igre razvijene u programskom jeziku C++ koristeći objektno-orijentisan pristup. Program je implementiran pomoću Raylib grafičke biblioteke i omogućava potpuno funkcionalno iskustvo Tetris igre sa podrškom za različite platforme, uključujući web platforme kroz Emscripten.

Implementacija se sastoji od šest glavnih komponenti: main.cpp (ulazna tačka programa), TetrisGame (glavna logika igre), GameGrid (upravljanje igralnom tablelom), TetrisPiece (logika tetris figura), GameConstants (konstante igre) i odgovarajuće header fajlove. Program implementira sve standardne Tetris mehanike uključujući gravitaciju, rotaciju figura, uklanjanje kompletnih linija, i sistem bodovanja.

# Analiza

## Cilj programa

Glavni cilj programa je kreiranje potpuno funkcionalne Tetris igre koja implementira sve standardne mehanike i funkcionalnosti originalne igre. Program je dizajniran da:

* Pruži autentično Tetris iskustvo sa standardnim tetris figurama (tetrominode)
* Implementira osnovne kontrole: pomeranje levo/desno, rotaciju, ubrzano padanje
* Omogući detekciju i uklanjanje kompletnih linija sa animiranim efektima
* Implementira sistem pauze i ponovno pokretanje igre
* Podrži različite platforme uključujući desktop i web
* Prikaže sledeću figuru i trenutni broj obrisanih linija
* Održi glatko gameplay sa 60 FPS-a

Program ima edukacijski značaj za učenje objektno-orijentisanog programiranja, game development principa, kao i praktičnu vrednost kao zabavna igra.

## Elementi programa

A screen shot of a computer

AI-generated content may be incorrect.

*Slika 1. Osnovna enum struktura*

**Klasa GameGrid:**

* Upravlja tablelom igre (12x20 polja)
* Implementira detekciju kolizije za sve pravce
* Omogućava pomeranje figura i detekciju kompletnih linija
* Upravlja konverzijom MOVING u FULL stanja
* Implementira algoritam brisanja kompletnih linija

**Klasa TetrisPiece:**

* Predstavlja tetris figuru u 4x4 matrici
* Implementira rotaciju figura pomoću algoritma transpozicije
* Generiše nasumične figure (7 standardnih tetromino oblika)
* Upravlja pozicioniranjem i pomeranjem figura

**Klasa TetrisGame:**

* Glavni kontroler igre koji koordiniše sve komponente
* Implementira glavnu game loop sa Update/Draw ciklusom
* Upravlja user input-om i game state-om
* Implementira timing sistem za različite aspekte gameplay-a
* Renderuje grafiku pomoću Raylib-a

**Konstante igre (GameConstants.h):**

* SQUARE\_SIZE: veličina jednog polja (20 piksela)
* GRID\_HORIZONTAL/VERTICAL\_SIZE: dimenzije table (12x20)
* Timing konstante za brzinu pomeranja i rotacije

## Izrada projekta

Program je razvijen koristeći napredne C++ koncepte i objektno-orijentisan dizajn:

**Objektno-orijentisan dizajn:** Program koristi enkapsulaciju sa privatnim podacima i javnim interfejsom. Svaka klasa ima jasno definisanu odgovornost: GameGrid za logiku table, TetrisPiece za logiku figura, TetrisGame za koordinaciju.

**Upravljanje memorijom:** Program koristi std::array umesto sirovih nizova za bezbednost tipa i automatsko upravljanje memorijom. Nema eksplicitne alokacije memorije što eliminiše mogućnost memory leak-ova.

**Cross-platform podrška:** Program je dizajniran da radi na desktop platformama i web-u kroz Emscripten. Koristi se uslovni kod kompajliranja za različite platforme.

**Timing sistem:** Implementiran je sofisticiran timing sistem sa različitim counterima za:

* Gravitaciju (prirodno padanje figura)
* Lateralno pomeranje
* Rotaciju
* Ubrzano padanje
* Animaciju brisanja linija

**State machine:** Program implementira jednostavan state machine sa stanjima: igra u toku, pauza, game over, brisanje linija.

## Funkcionisanje programa

Program funkcioniše kroz glavnu game loop koja se izvršava 60 puta u sekundi:

**Inicijalizacija:** Program kreira prozor 800x450 piksela, inicijalizuje game grid sa granicama, postavlja početne vrednosti varijabli i priprema prvi par figura.

**Input handling:**

* **Strelice levo/desno:** Lateralno pomeranje aktivne figure
* **Strelica gore:** Rotacija figure za 90 stepeni u smeru kazaljke na satu
* **Strelica dole:** Ubrzano padanje figure
* **P:** Pauza/unpauza
* **Enter:** Restart nakon game over

**Game state management:**

1. **Kreiranje nove figure:** Kada nema aktivne figure, kreira se nova na vrhu table
2. **Pomeranje figure:** Figura se pomera na osnovu korisničkog input-a i gravitacije
3. **Kolizija i sletanje:** Kada figura dotakne dno ili drugu figuru, konvertuje se u FULL stanje
4. **Detekcija linija:** Program proverava da li su formirane kompletne linije
5. **Animacija brisanja:** Kompletne linije se animiraju blikenjem pre brisanja
6. **Brisanje i pomeranje:** Kompletne linije se brišu, a sve figure iznad se pomeraju dole

**Rendering:** Program renderuje:

* Glavnu tabelu sa različitim bojama za različita stanja polja
* Preview sledeće figure
* UI informacije (broj linija)
* Game over i pauza poruke

**Algoritamska složenost ključnih operacija:**

* Detekcija kolizije: O(n) gde je n broj MOVING polja
* Rotacija figure: O(1) - fiksna 4x4 matrica
* Detekcija kompletnih linija: O(w×h) gde su w,h dimenzije table
* Brisanje linija: O(w×h) u najgorem slučaju

**Tetromino shapes:** Program implementira svih 7 standardnih tetromino oblika:

* I-piece (linija)
* O-piece (kvadrat)
* T-piece
* L-piece i J-piece
* S-piece i Z-piece

# Zaključak

Implementirana Tetris igra predstavlja kompletnu, profesionalnu implementaciju klasične igre sa svim standardnim funkcionalnostima. Program uspešno demonstrira napredne C++ koncepte i game development principe.

Program predstavlja odličnu osnovu za dalje proširivanje i može se koristiti kao referentna implementacija za učenje game development-a u C++. Kod je dobro organizovan, dokumentovan kroz jasne nazive funkcija i varijabli, i lako se održava.

# Literatura

1. LAMS
2. Raylib - <https://www.raylib.com/>
3. Google
4. StackOverflow