



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA



Predmet : Multiprocesorski sistemi

Napisati paralelni softver koji implementira igricu po uzoru na
Pac-Man Game.

Asistent : Anja Tanović

Student : Milin Ivan E1-79/2023

Na početku je igrica implementirana sekvencijalno, pomoću C++ programskog jezika i korišćena je **SFML biblioteka** za obradu slike, a zatim paralelizovana pomoću OpenMP biblioteke za paralelno programiranje.

Funkcije koje su kreirane u cilju realizacije igrice:

1. void copyMatrixAndModify() – kopira sadržaj iz matrice *mazeMatrix* u matricu *foodMatrix* i rubove matrice postavlja na nulu.

2. void movePacman(Sprite& pacmanSprite) – prima kao parametar referencu objekta *pacmanSprite* klase *Sprite*.

Ukoliko se pritisne neki od tastera W, A, S ili D pomera ikonicu *pacmana* gore, dole, levo ili desno. Kada se *pacman* nađe na ivici mape i ukoliko ne postoji prepreka u vidu zida prebacuje *pacmana* sa skroz leve tačke na skroz desnu tačku ili sa skroz gornje tačke na skroz donju tačku mape i obrnuto.

3. void moveGhost(Ghost& ghost, Sprite& ghostSprite) – prima kao parametar referencu na strukturu *Ghost*, kao i objekat *ghostSprite* klase *Sprite*. Pomoću randomizacije nasumično pomera duha po mapi.

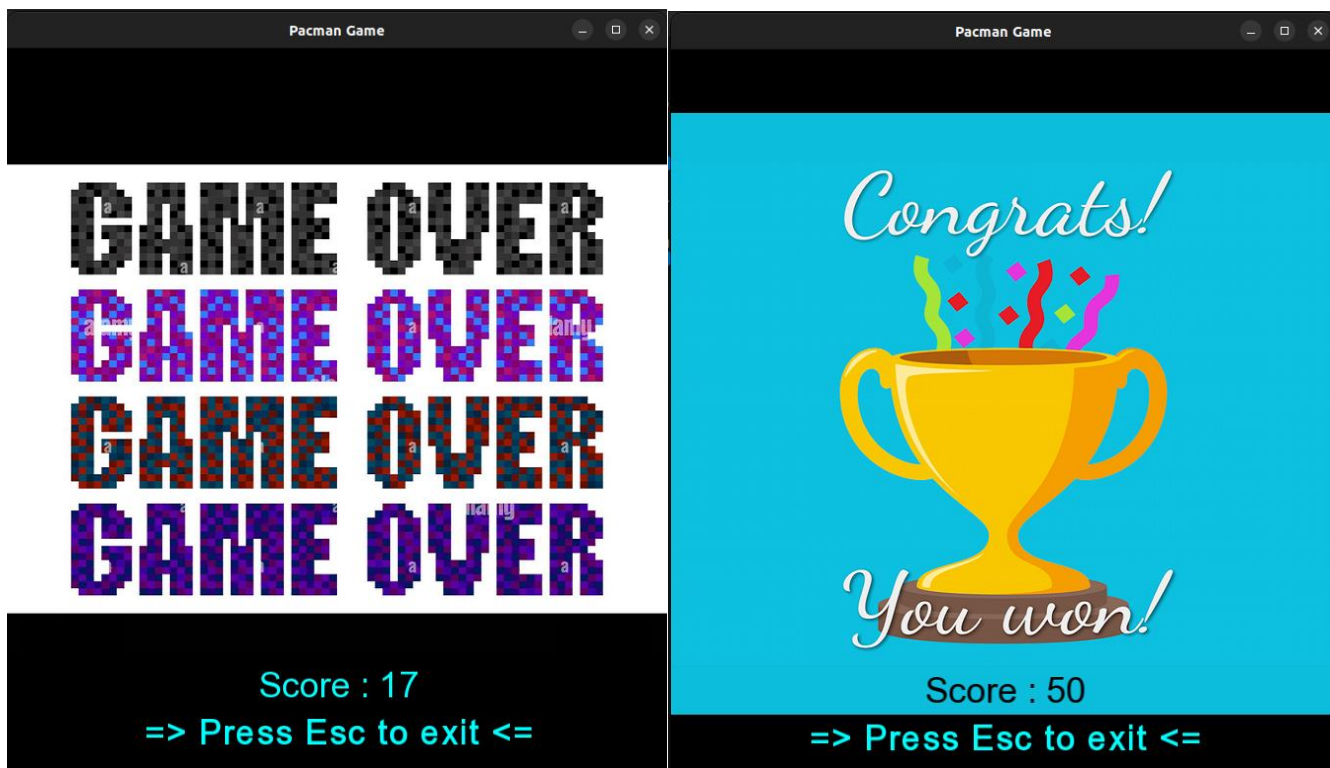
4. void initializeFruits(std::vector<Fruit>& fruits) – prima kao parametar referencu vektor *fruits* strukture *Fruit*.

Poziva funkciju *copyMatrixAndModify()* i zatim svugde gde je matrica *foodMatrix* jednaka jedinici na to mesto u vektoru *fruits* postavlja jedinicu.

5. void removeEatenFruit(std::vector<Fruit>& fruits, int x, int y) – prima kao parametar referencu vektor *fruits* strukture *Fruit*, kao i koordinate x i y koje zapravo predstavljaju koordinate hrane koje treba da se uklone kada pacman „pojede” hranu sa mape.

6. void displayImageAndText(RenderWindow& window, Sprite& game_overSprite, Text& instructionToExitGame, Text& currentScoreText, int score) – prima kao parametar referencu na *window* koja je objekat klase *RenderWindow*, isto važi i za *game_over*, *instructionToExitGame*, *currentScoreText* i promenljivu *score*.

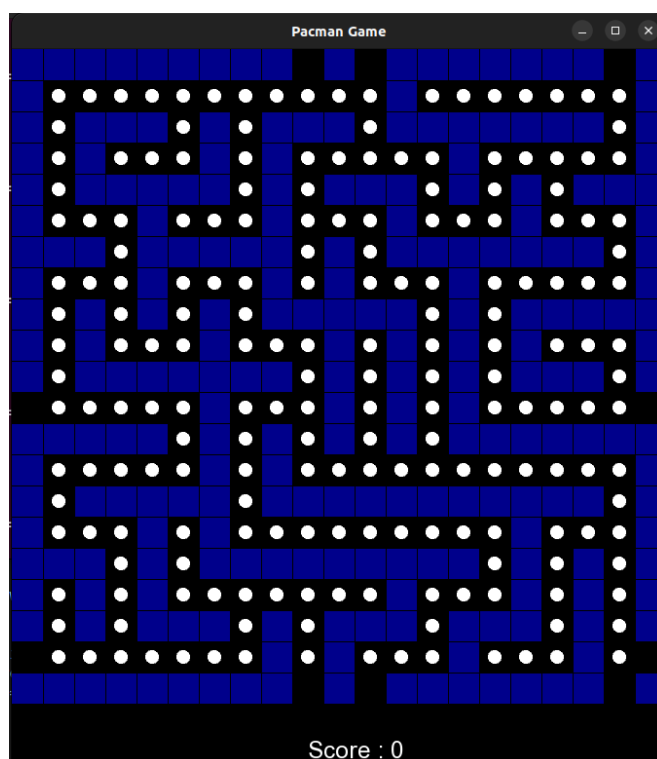
Ova funkcija se poziva ili kada *pacmana* pojede duh i treba da prikaže sliku *game_over.png* ili kada *pacman* pojede svu hranu sa mape tada treba da prikaže sliku *you_won.png*, takođe prikazuje i ukupan broj hrane koju je *pacman* pojeo.



7. `void drawMap(RenderWindow& window, RectangleShape& block, RectangleShape& line_vertical, RectangleShape& line_horizontal, CircleShape& fruitShape)` - prima kao parametar referencu na *window* koja je objekat klase *RenderWindow*, isto važi i za *block*, *line_vertical*, *line_horizontal* i *fruitShape*.

Is crtava mapu, na pozicijama gde je matrica *mazeMatrix* jednaka jedinici te blokove će obojiti u crno i postaviti hranu na mapi u vidu belog kruga, dok na pozicijama gde je matrica *mazeMatrix* jednaka nuli te blokove će obojiti plavo.

Takođe radi estetike dodate su horizontalne i vertikalne crne linije *line_vertical* i *line_horizontal*, na kraju funkcije ukoliko *pacman* pojede hranu sa te pozicije se briše beli krug.



- *RenderWindow window(VideoMode(w,h+2*blockSize),"Pacman Game");*

Na početku int main() dela programa prvo se poziva konstruktor kojim se kreira prozor dimenzija *w* i *h+2*blockSize* i na gornjem delu prozora ispisuje se *Pacman Game*.

$N = 21px$, *blocksize* = 32px, $w = h = blocksize * N$

- *RectangleShape block(Vector2f(blockSize, blockSize));*
- *CircleShape fruitShape(blockSize / 4.5);*
- ...
- *fruitShape.setFill(Color::White);*
- *line_vertical.setFill(Color::Black);*

Zatim se kreiraju oblici za *block*, *line_vertical* i *line_horisontal* i dodeljuju im se boje.

Nakon toga pomoću mehanizma *Texture* i *Sprite* učitavaju se slike iz foldera */images* kojima će biti predstavljeni *pacman*, duhovi (crveni i plavi) i slike *game_over* i *you_won*.

Kako bi mogao da se ispisuje tekst unutar igrice neophodno je da se učita font iz foldera */font*, zatim se pomoću klase *Text* kreiraju objekti *pauseText*, *currentScoreText* i *instructionToExitGame* kojima će se ispisati predviđeni tekst.

Pomoću promenljive *delay* podešava se koliko se brzo kreću *pacman* i duhovi.

Nakon toga postavljaju se koordinate *pacman* karaktera, kao i za oba duha.

Programer sam može da podesi koordinate, dok je kretanje duhova randomizovano.

Unutar petlje *while(window.isOpen())* proverava se prvo da li je *pacman* pojeo svu hranu sa mape, ako jeste prozor igrice se briše, poziva se funkcija *displayImageAndText* koja prikazuje sliku *you_won* i ispisuje broj hrane koju je *pacman* pojeo. Ako nije prvi uslov ispunjen proverava se druga if-naredba koja proverava da li je *pacman* „pojeden” od strane prvog ili drugog duha, ako jeste prozor igrice se briše poziva se funkcija *displayImageAndText*, ovaj put se prikazuje slika *game_over* i ispisuje se koliko je hrane *pacman* pojeo.

Ove dve if naredbe su podeljene u dve sekcije kako bi se brže proverilo da li je jedna od if-naredbi tačna.

Zatim unutar petlje *while(window.pollEvent(e))* proverava se da li je mišem pritisnuto dugme *X* na prozoru ili taster *Escape*, ako jeste zatvara se igrice, takođe se proverava da li je pritisnuto dugme *Space* na tastaturi, pri pritisku ispisuje se na ekranu „*Paused*” i igrice se zaustavlja.

U narednoj if-proveri proverava se koji je taster za kontrolu *pacmana* pritisnut i na konto toga se *pacman* pomera u tom smeru, samo jedan pritisak tastera rezultuje da se *pacman* pomeri samo jednu poziciju.

Naredna if-provera je zadužena za paralelnu implementaciju *pacman* igrice, svaka nit ulazi u jedan task, a svaki task izvršava različite zadatke:

- **Prvi task** zadužen je za crtanje mape kao i crtanje pacman karaktera kao i oba duha
- **Drugi task** zadužen je za kretanje prvog duha
- **Treći task** zadužen je za kretanje drugog duha
- **Četvrti task** zadužen je za kretanje *pacmana*

Nakon izlaska iz te if-provere dodatno se na dnu mape ispisuje trenutni broj hrane koju je *pacman* pojeo i na kraju svih izmena prozor se prikazuje pomoću *window.display()*.