TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A PROJECT REPORT ON

Tetris: The Classic Puzzle Game

Submitted by:
AASHISH KARKI (078BCT004)
ANUP ARYAL (078BCT015)
APIL CHAUDHARY (078BCT017)

SUBMITTED TO:
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
IOE, Pulchowk Campus
Kathmandu Nepal

August 24, 2023

1 ACKNOWLEDGEMENT

We express our sincere gratitude to our teachers(Mr. Daya Sagar Baral, Mr. Aman Shakya and Mr. Rajad Shakya) who suggested building this game for sharpening our Object Oriented Programming skills and logic building capabilities.

Our special thanks goes to our lecturer, Daya Sagar Baral, for his guidelines, suggestions, and instructions, which have served as a contributor towards the inception of this project.

We sincerely thank the Department of Electronics and Computer Engineering, Pulchowk Campus, for giving us an opportunity to work on this project to expand our knowledge of Object Oriented Programming and working in a team.

2 ABSTRACT

This report presents the development process, design decisions, and outcomes of our Tetris game project. Tetris, a classic tile-matching puzzle video game, has been a popular choice for both casual gamers and enthusiasts since its inception. This project aimed to create a modern and engaging implementation of Tetris while exploring key aspects of game development, Object-Oriented Principles, graphics rendering, user interaction, and algorithm design.

The project involved designing and implementing the game mechanics, user interface, scoring system, and visual elements of Tetris. Leveraging the capabilities of SFML(Simple and Fast Multimedia Library), we built a playable version of Tetris that captures the essence of the original game while adding visual enhancements and a user-friendly interface.

In this report, we delve into the technical details of our game's architecture and discuss the challenges we encountered during development. We describe our approach to handling user input, managing game state, rendering graphics, and optimizing performance. Additionally, we highlight the design considerations that went into creating an intuitive user experience, including responsive controls and visual feedback.

Throughout the project, we had the opportunity to apply principles learned in our Object Oriented Programming course, gaining hands-on experience in software design, algorithms, and teamwork. We collaborated closely, leveraging each team member's strengths to overcome obstacles and make informed design choices. Our project supervisors, Mr. Daya Sagar Baral and Mr. Rajad Shakya, provided valuable guidance and feedback, ensuring the project's alignment with our learning objectives.

As a result of our efforts, we have successfully developed a functional and enjoyable Tetris game that showcases our technical skills and creativity. This project not only enhanced our understanding of game development but also provided us with a platform to apply theoretical knowledge to a practical context. We hope that our Tetris game brings joy to players. At last, we learned to use SFML library in our C++ projects and learned a lot about GIT and Version Control Systems while working in a team

Contents

3 OBJECTIVES

The objectives of building this game in C++ are listed below:

1. **Use Object-Oriented Programming (OOP):**
    - Create classes for Tetrominoes, game board, and game logic.
    - Organize code for easy maintenance and flexibility.
2. **Explore C++ Features:**
    - Manage game data efficiently using vectors, pairs, maps and other containers in STL.
    - Handle resources smartly to optimize performance.
3. **Graphics with SFML:**
    - Use SFML to draw Tetrominoes, game board, and animations.
    - Respond to player input using SFML's event system.
4. **Reusable Code:**
    - Create header files for sharing code across the game.
5. **Implement Game Mechanics:**
    - Make Tetrominoes rotate, move, and detect collisions.
    - Create a scoring system and difficulty levels.Create a scoring system and difficulty levels.Create a scoring system and difficulty levels.
6. **Add Sound Effects and Music:**
    - Integrate audio using SFML for a better gaming experience.
7. **Test and Debug:**
    - Thoroughly test the game to fix any issues.
    - Use logs and debugging tools for troubleshooting.
8. **Team Collaboration:**
    - Communicate well within the team.
    - Use Git for version control and collaboration.

4 INTRODUCTION

Tetris, an iconic and enduring puzzle game, has captured the hearts of gamers around the world since its inception in 1984. Created by Russian game designer Alexey Pajitnov, Tetris challenges players with a deceptively simple objective: fitting different-shaped blocks, known as tetrominoes, together to form complete lines. Yet, beneath its straightforward premise lies an addictive and exhilarating experience that has stood the test of time.

Gameplay :

Tetrominoes: The game consists of various tetrominoes, each made up of four connected squares arranged in different shapes. The seven tetrominoes are: I, J, L, O, S, T, and Z.

Falling Blocks: Tetrominoes fall from the top of the playing area one at a time. You have the ability to move and rotate the tetrominoes as they descend.

Movement and Rotation: You can move the falling tetrominoes horizontally (left or right) using the arrow keys or buttons. You can also rotate them clockwise or counterclockwise to fit them into available spaces using the up arrow key or designated rotation buttons.

Clearing Lines: The primary goal is to create complete horizontal lines by filling all the spaces within a row. When a line is entirely filled, it clears from the screen, and you earn points.

Scoring: The more lines you clear at once, the higher the points you earn. Clearing multiple lines simultaneously is known as a ”Tetris” and yields the highest points.

Game Over: The game ends when the stack of falling tetrominoes reaches the top of the playing area. If you can’t clear lines fast enough, the stack will reach the top, and the game will be over.

Controls: The controls for Tetris are usually straightforward:
Left Arrow: Move tetromino left
Right Arrow: Move tetromino right
Down Arrow: Accelerate the fall of the tetromino
Up Arrow: Rotate tetromino clockwise or counterclockwise

5 APPLICATIONS

Tetris was originally inspired by a puzzle game called ”pentominoes,” in which different wooden shapes made of five equal squares are assembled in a box. Hence, Tetris is also a puzzle in itself and thus a “mind” game. Although, video games are often dismissed as unsophisticated or the domain of couch potatoes, but many common elements of these simulated worlds can provide tangible benefits in real life. Benefits of Tetris for both children and adults include:

- Healthy Brain Simulation
- Development of Problem Solving Skills
- Stress Relief

Taking these benefits into account, application of Tetris are discussed below:

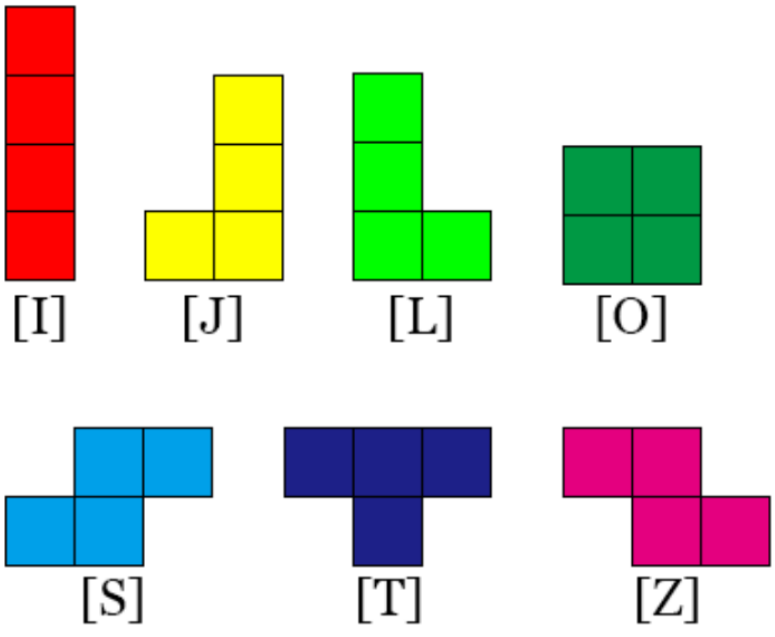
1. **Problem Solving:** As already mentioned above, Tetris is a puzzle game and thus, is a tool to develop problem solving skills. Placing each block requires a strategy so to get that neat high score everyone craves for. This, in turns develops a habit of searching of the best route for the required outcome.
2. **Quick Decision Making:** Tetris does not only require a gamer to develop a strategy but it also requires them to make a decision to take a certain strategy in a quick manner. There is so little time between the clearing of a line and the falling of the next block that the gamer must choose an optimum decision quickly.
3. **Mental Health:**With something as concerning as mental health, it is only right to search for a remedy to it that boosts morale and mood. Tetris, and video games, in general, are a great tool that can be very helpful.
4. **Learning:**Although video games are portrayed as something that is an obstruction for learning, in truth, the portrayal is anything but far from truth. Tetris can be a great tool for learning as one is required to use their memory, spatial intelligence, problem solving, decision making, etc.
5. **Balancing Time and Priority:**Tetris can be a great teacher if you’re having difficulty prioritizing. To keep up with the game, you must change your priorities very quickly. You can apply this pattern in the real world when you encounter situations that disturb your routine and require you to switch tasks.

6 LITERATURE SURVEY

It all began with a puzzle-loving software engineer named Alexey Pajitnov, who created ”Tetris” in 1984 while working for the Dorodnitsyn Computing Centre of the Soviet Academy of Sciences, a research and development center in Moscow created by the government. Pajitnov was inspired by a puzzle game called ”pentominoes,” in which different wooden shapes made of five equal squares are assembled in a box. Pajitnov imagined the shapes falling from above into a glass, with players controlling the shapes and guiding them into place. Pajitnov adapted the shapes to four squares each and programmed the game in his spare time, dubbing it ”Tetris.” The name combined the Latin word ”tetra” — the numerical prefix ”four,” for the four squares of each puzzle piece — and ”tennis,” Pajitnov’s favorite game. And when he shared the game with his co-workers, they started playing it — and kept playing it and playing it. These early players copied and shared ”Tetris” on floppy disks, and the game quickly spread across Moscow. When Pajitnov sent a copy to a colleague in Hungary, it ended up on display in a software exhibit at the Hungarian Institute of Technology, where it came to the attention of Robert Stein, owner of Andromeda Software Ltd., who was visiting the exhibit from the United Kingdom. ”Tetris” intrigued Stein. He tracked down Pajitnov in Moscow, but ultimately the game’s fate lay in the hands of a new Soviet agency, Elektronorgtechnica (Elorg), created to oversee foreign distribution of Soviet-made software. Elorg licensed the game to Stein, who then licensed it to distributors in the U.S. and the U.K. — Spectrum HoloByte and Mirrorsoft Ltd — The New York Times reported in 1988. According to the Times, ”Tetris” was the first software created in the Soviet Union to be sold in America. In Brown’s book, the unusual story of ”Tetris” is interwoven with an exploration of gaming: why people do it, how it changes them and how it brings people together. Pajitnov himself began this journey simply because he loved games and puzzles and wanted to share them with the world. And in the process, ”Tetris” took on a life of its own.

7 EXISTING SYSTEM

Tetris is a puzzle video game created by soviet software engineer Alexey Pajitnoovin 1984. It has been published by several companies for multiple platforms. The vanilla version of the game consisted of a board of dimension 10*18. It consisted of seven blocks of different shapes called tetromino blocks which were dropped continuously from the top of the board in a random order. It is this version of the game that we are taking as reference. The player has to smartly place each block in the board so as to fill a line. Filling a line will clear that particular row and all the blocks above that row will drop on that row. Blocks above this row will drop on this row and so on. Basically, the player has to prevent the board from filling otherwise the game will end displaying the final score. The vertical motion of the dropping block is automated and updates on each game tick so players cannot control the motion of the block along the column but they can rotate blocks in all directions and move them left and right along a row as they fall from the top. Blocks can't be moved once they stop dropping. Players can also see the preview of the next block in the side window. Today, Tetris can be found in a wide range of devices from PCs to mobile phones, with each adding new features to the already existing system.



(Tetrominos)

8 METHODOLOGY

This project aimed to develop a Tetris game using Object-Oriented Programming principles and the SFML library for graphics. The project was implemented using the CLion Integrated Development Environment and utilized CMake for cross-platform build support. Version control and collaborative coding was facilitated through Git and GitHub.
For detailed history of development of this game,
Link to our GitHub Repository:
https://github.com/Aashish079/Tetris

To ensure a structured and maintainable codebase, we have designed the class hierarchy for the game. Different classes are created by inheriting from the parent class State, to represent game objects such as the GamePlayState class, GameOverState class, HighScoreState tetrominoes, and FileManager. We have employed OOP concepts such as encapsulation, inheritance, and polymorphism to achieve modularity and code reusability in this project. The game logic is implemented within the GamePlayState class. Each class have private and public member variables and member functions that handle specific aspects of the game, such as moving and rotating tetrominoes, checking for line clears, and updating the score. The SFML library is utilized for rendering graphics and handling user input.

To ensure efficient functioning, smooth transistioning and good development experience we have used starter code of SFML named TheStateMachine which stores and renders different states such as MainMenuState, GameplayState and GameOverState.

To facilitate collaboration among team members, we utilized Git and GitHub for version control. We had total of 7 different branches for different features. Regular commits were performed to the repository and timely pull request were initiated from different branches into main branch. GitHub's issue tracking feature was also utilized to manage and prioritize tasks. Proper documentation was maintained to provide an overview of the project structure, class hierarchy, and function specifications..

We tried our best to take the most systematic and effiecient development approach throughout the development of this project.

9 IMPLEMENTATION

10 RESULTS

After the final program was coded, the result obtained was close to what was expected. Game uses keyboard keys for function. Program starts with a short Intropage (for 2 sec) and then Mainmenu loads-up, with Play, HighScore, and Exit buttons. When played, the game goes into play state and game runs.

When the user gets out, GameOverState loads-up, and he is asked his name for storing his scores in a File. Then he can view top-ten scores in by pressing HighScore button and can press backspace to return back to the mainmenu state.
Then pressing the Exit button will close the gameplay window.

Figure 1: IntroState

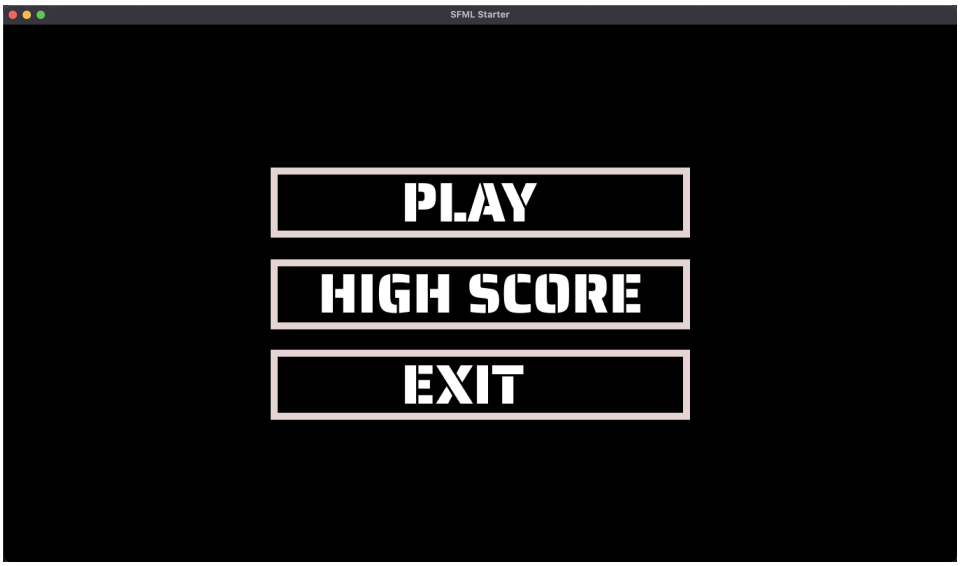


Figure 2: MainMenuState

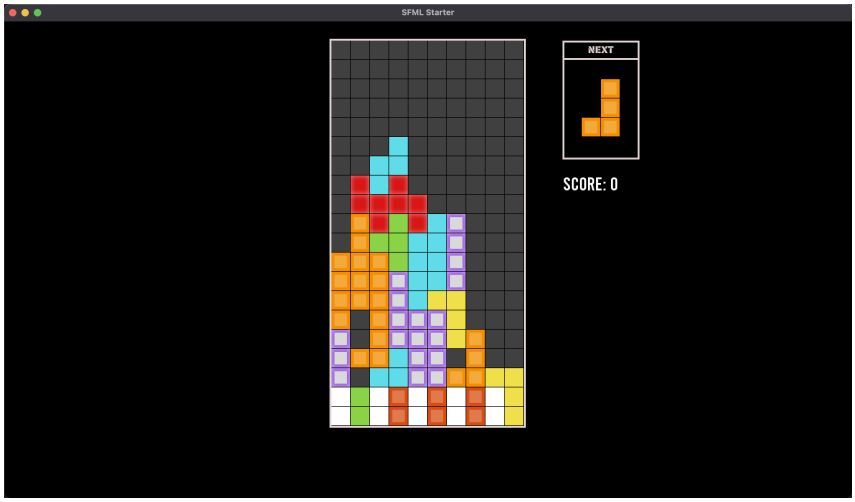


Figure 3: GamePlayState

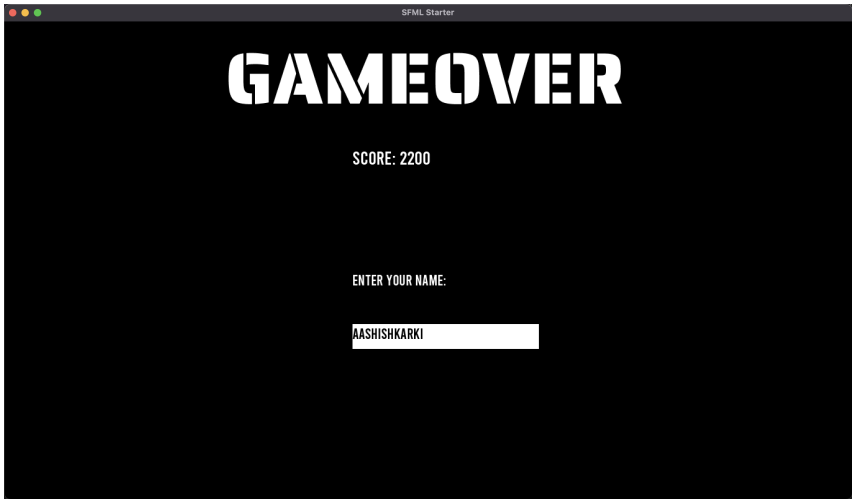


Figure 4: GameOverState

Figure 5: S

11 PROBLEMS FACED AND SOLUTIONS

1. **Problem:** IDE setup using CMake for cross-platform(Windows, Linux, MacOS)
   Solution: We used Clion IDE

2. **Problem:** Managing Different Pages efficiently for better developer experience
   Solution: We researched and found SFML Starter-Kit called TheStateMachine

3. **Problem:** Getting Familiar with SFML
   Solution: We went through comprehensive SFML Tutorial and bunch of Tutorial Videos to get good grasp of SFML

4. **Problem:** Connecting Different Classes
   Solution: We used Inheritance and Composition by creating user-defined headerfiles

5. **Problem:** Using STL for optimal Data and Memory Management
   Solution: We learnt STL from Course Book and referenced about it from the internet on the go.

6. **Problem:** Getting and Storing HighScore and User's Name
   Solution: We used Pair and Map data structures for storing and retriving Score and Name of Player.

12 LIMITATIONS AND FUTURE ENHANCEMENTS

We have tried to include most of the features in our game but with limited time, we could not include everything. In our version of this game, we could have created multiplayer mode but this remains a limitation to our program primarily because we did not have the time to include it in our project. This feature will certainly be added in the next update to the game.

Also we had thought of adding special feature like drop of bomb after clearing multiple rows and could have made UI more responsive adaptable to different window sizes but we couldn't add them due to lack of time.

Further, we believe that we could further organize our code and make it much cleaner and easier to read. We could have used the object-oriented approach more effectively.

In the future, we can further re-organize the code and make a much faster game. On top of that, we could allow multiplayer over the internet. This would be an exciting feature but we believe it's outside the scope of the project.

13 CONCLUSION AND RECOMENDATION

We designed and built this project using the basic principles of Object Oriented Programming along with Simple and Fast Multimedia Library. The use of OOP easily organized our code into different modules and implement those modules at the required time.

The use of game class showcases this idea, where we have created a game class and just updated and rendered it in the main class. But as we go deeper into the implementation in game class, we see the actual commands. This abstraction is visible throughout the program and it made coding it much easier. Further, the use of a graphic library came in quite handy as well. The design of the UI of the game became very much easier and it saved us from coding thousands of lines of code. We also learned the need for planning, execution, and testing. The game logic is the same as the classical Tetris game, but the design and implementation are different.

14 REFERENCES

Websites :

1. https://www.sfml-dev.org/documentation/2.5.1
2. https://github.com/kiswa/SFMLStarter
3. https://geeksforgeeks.com
4. https://stackoverflow.com
5. https://youtube.com

Books:

1. The Secrets of Object Oriented Programming in C++ -Daya Sagar Baral, Diwakar Baral
2. Object-Oriented Programming in C++ - Robert Lafore (Fourth Edition)