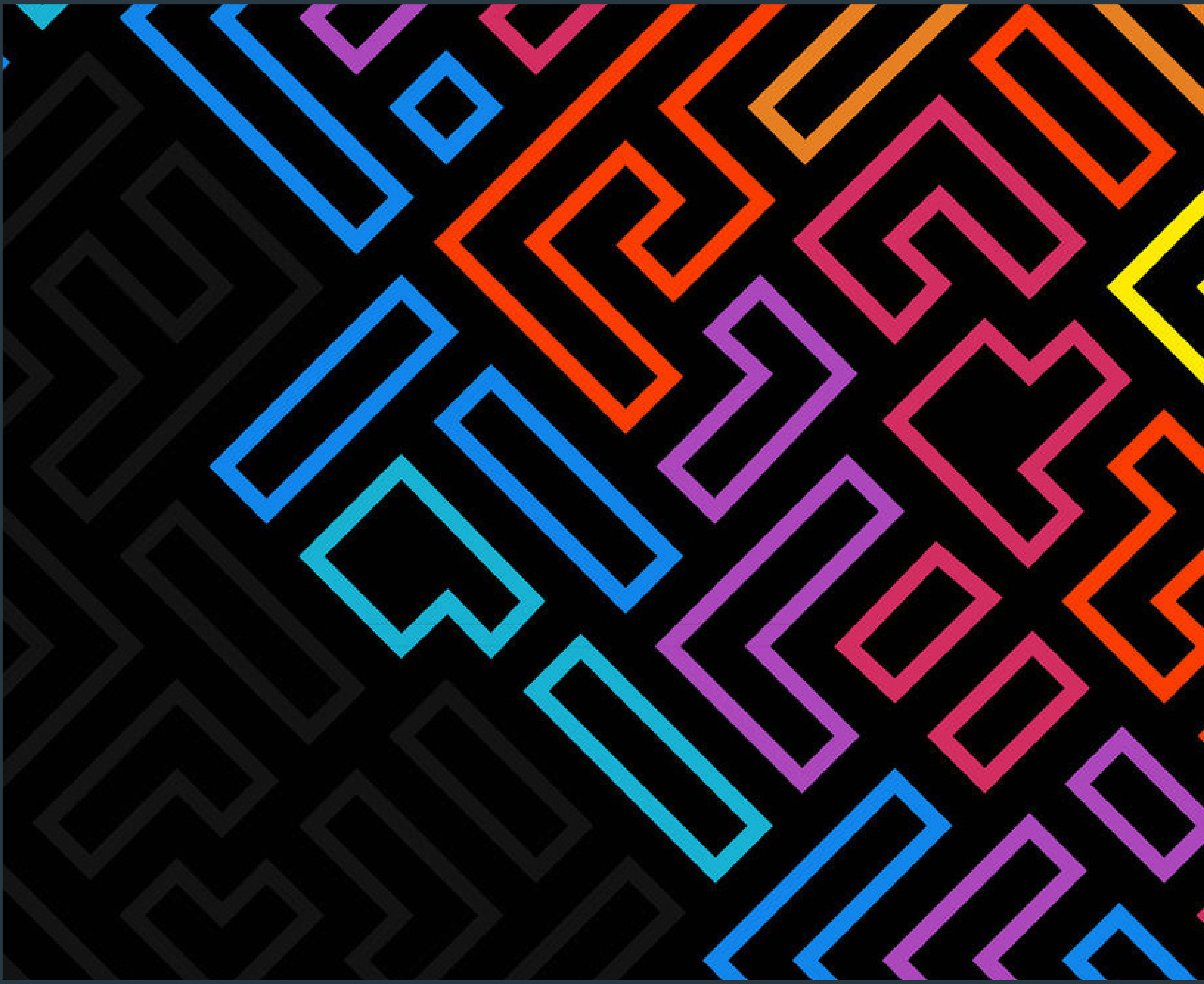
DATA STRUCTURES & ALGORITHMS

Open Ended Lab

Game Description: **TETRIS**



Submitted by:

- *Mohib Ud Din (CS-21142)*
- *Sameed Fayiz (CS-21127)*
- *M. Maaz Khan (CS-21019)*

B.E Computer Systems & Information Engineering

Problem Description

In this project, we developed a game of Tetris using Pygame library. The game features a human player who competes against the computer, with the computer acting as the second player or assessing the moves of the human player and generating playing conditions accordingly. The game utilizes various data structures, such as lists and dictionaries, to keep track of the game state and the computer's moves. Algorithms, such as collision detection and line clearance, were implemented to provide a smooth and challenging gameplay experience. The Pygame library was used to create an interactive user interface, allowing the player to easily control the game and view their progress. Overall, this project showcases the power of Python and its capabilities in developing engaging and interactive games.

Game Functions

- **Collision Detection**: This algorithm is used to determine if the current falling tetromino is colliding with any other blocks on the game grid. This algorithm is critical for ensuring that the player cannot place tetrominoes in invalid positions. It is typically implemented by checking if any of the blocks in the current tetromino have the same position as any blocks already on the game grid.

- **Line Clearance**: This algorithm is used to check if any lines on the game grid are fully filled with blocks, and if so, clears them from the game grid. This is typically done by iterating through each row of the game grid, and checking if every cell in that row is filled with a block. If a line is found to be fully filled, it is cleared by shifting all the rows above it down by one.

- **Scoring Algorithm:** This algorithm is used to keep track of the player's score, which is typically increased whenever a line is cleared. Some games will award more points for clearing multiple lines at once, or for clearing lines quickly.
- **Drop Algorithm:** This algorithm is used to move the current tetromino down the game grid over time, simulating the gravity in the game. The rate at which the tetromino drops can be adjusted to increase the difficulty of the game.

Challenges Faced

- **Difficulty balancing**: Another challenge faced was balancing the difficulty of the game. This could be caused by the game being too easy or too difficult, which could lead to players losing interest. To overcome this, we used playtesting to get feedback from players and make adjustments accordingly.

- **Bug fixing**: As with any software development project, bug fixing is an important task. This could include bugs in the game logic, graphical issues, or issues with the user interface. To fix these bugs, we used debugging tools and print statements to track down the source of the problem, and implement solutions to fix it.

- **Game Logic:** Game logic can be a complex task, especially when it comes to Tetris because of the several variations of the game. To overcome this, we broke down the game into smaller parts and worked on them individually.

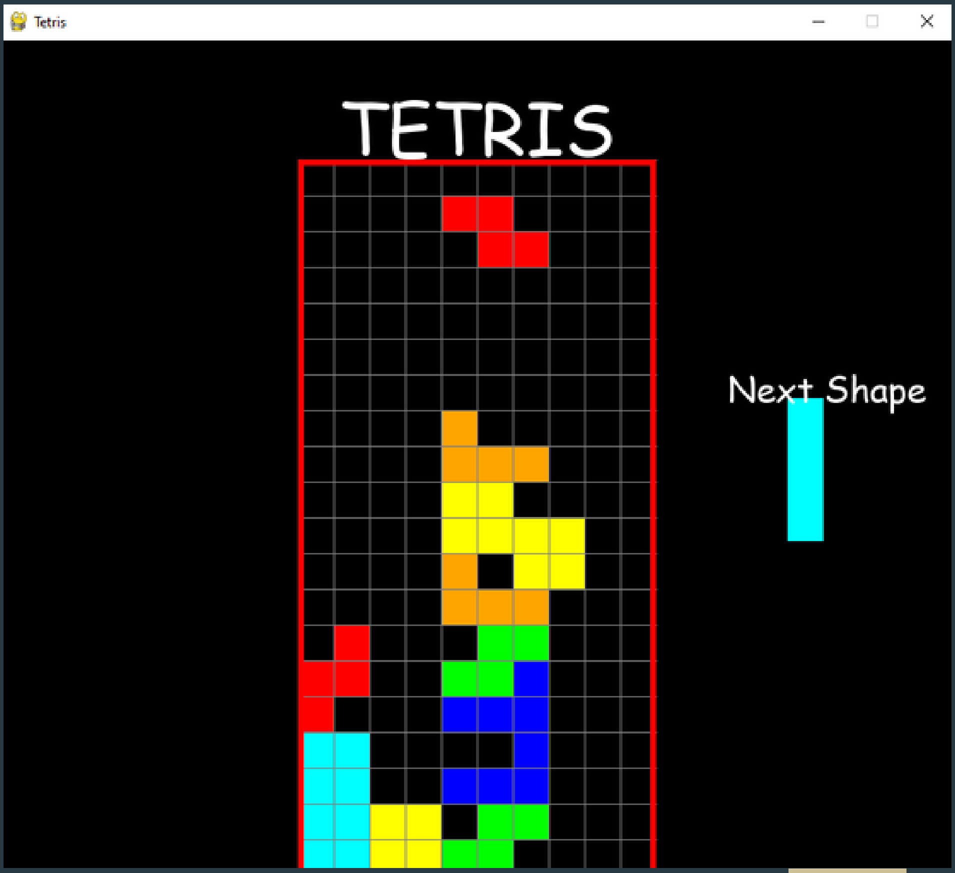
Takeaways

We were able to learn how to develop complex applications, the use of pygame, deploy optimizing algorithms and how to select the right data structure for the data. Additionally, looking out for each other in team and working through the complexities made us professionally relevant and able.

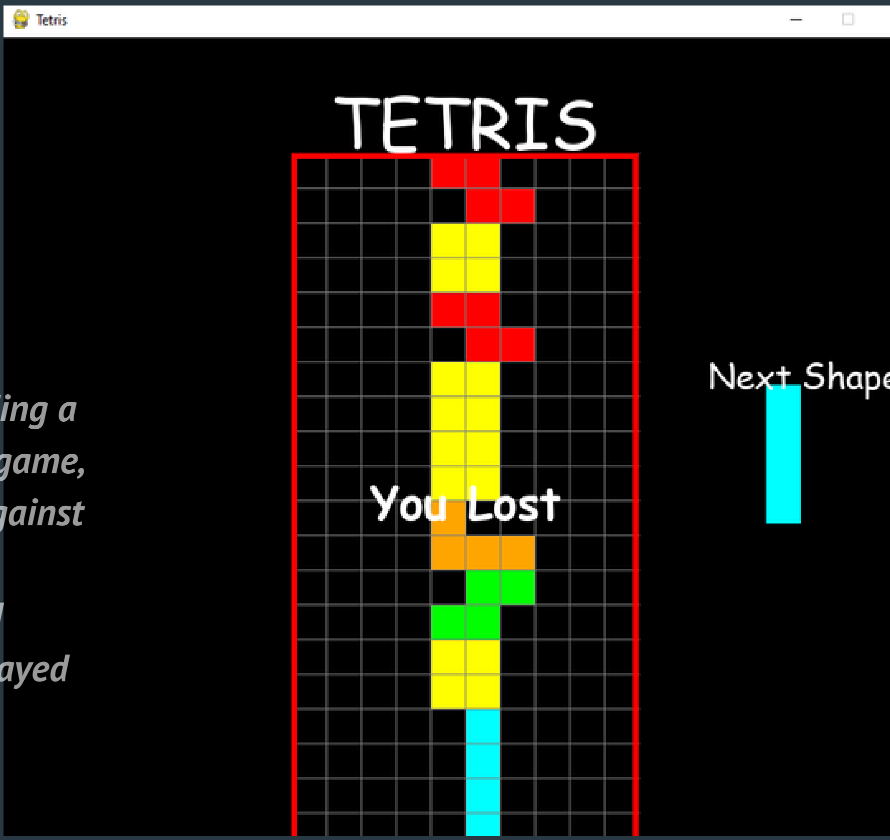
Data Structures Used

1. **List**: In the game, the falling tetromino blocks are stored in a list, which is updated in real-time as the blocks move and rotate. The list is used to keep track of the current state of the game and to render the blocks on the screen.
2. **Dictionary**: The various tetromino shapes and their rotations are stored in a dictionary. Each key in the dictionary represents a unique tetromino shape, and the corresponding value is a list of rotations for that shape. This data structure is used to retrieve the correct rotations for each tetromino as it falls and rotates, allowing the game to dynamically change the shape of the falling blocks. The dictionary is also used to keep track of the color of each tetromino.

Snippets of Game Running



Future Innovations



1. *Multiplayer: Another potential future innovation could be adding a multiplayer component to the game, allowing players to compete against each other online.*
2. *Virtual Reality and Augmented Reality: Tetris could be also played in virtual reality, providing an immersive and interactive experience for players.*
3. *Mobile and Web support: Developing Tetris as a mobile app or web-based game would allow players to access it from any device with a web browser or mobile app store.*
4. *Extra Features: adding new features such as power-ups, game modes, or achievements to the Tetris game could make it more engaging and enjoyable for players.*
5. *Game Analytics: Collecting and analyzing data on how players interact with the game to improve the gameplay experience and adjust the game's difficulty.*