Project Report - Group 19

This project involved the design and development of a fully functional Tetris game using Python. The main components of the system included the board.py, piece.py, and main.py files, which together handled the game's logic, piece representation, and graphical user interface using Pygame. The project aimed not only to replicate the classic Tetris experience but also to enhance it with features like a hold-piece mechanic, competitive multiplayer mode, and a modern scoring system. As the team leader, my responsibilities included creating the GitHub repository for version control, managing the workflow among team members, and contributing to the development of the board.py file. The design of the project was modular, with each file handling specific responsibilities: piece.py: Defined the different tetromino shapes, their rotations, movements, and copies. This file provided the building blocks of the game. board.py: Managed the grid where pieces were placed, handled collision detection, line clearing, score calculation, level progression, and competitive garbage line mechanics. It also implemented the hold feature and ghost piece preview. main.py: Integrated the boards into a game loop, handled input for both single-player and multiplayer modes, and rendered the game using Pygame through a TetrisRenderer class. This separation of concerns allowed for easier debugging, testing, and collaboration. The design also made the game extensible, with potential for additional features such as new game modes or visual themes. Throughout the project, both I and my team encountered several technical and organizational challenges:
1. Version Control with GitHub: At the start, most of us were unfamiliar with GitHub workflows. Issues like merge conflicts and improper branch management slowed down progress. I overcame this by studying GitHub tutorials and guiding my team through proper branching, pull requests, and commit practices. By the end of the project, everyone was more confident with collaborative version control.
2. Collision Detection and Piece Rotation: One of the toughest coding challenges was ensuring that tetromino pieces interacted correctly with the grid boundaries and other pieces. Rotating pieces near walls often resulted in unexpected behavior. We solved this by implementing thorough collision checks and testing edge cases using pytest, which I also learned and introduced to the team.
3. Multiplayer Synchronization: Designing the competitive mode introduced complexity because both boards had to interact through garbage line mechanics. Ensuring fairness while avoiding desynchronization was difficult. We tackled this by clearly defining how garbage lines would be generated and added, and by simulating multiple games during testing.
4. Team Coordination: As with any team project, aligning everyone's pace was a challenge. Some members worked faster than others, while others needed help with debugging. I made sure to hold regular check-ins, divide tasks clearly, and assist teammates when they were stuck.

My main contributions were creating and managing the GitHub repository, implementing key logic in the board.py file such as line clearing, scoring, and the hold-piece feature, and helping to debug collision and movement issues. I also contributed to the design of the competitive mechanics by suggesting and implementing garbage lines to make multiplayer more engaging.

While the current project achieved its core goals, there are several directions for future improvement: 1. Improved Graphics and Animations: Adding smoother transitions, visual effects for line clears, and customizable themes could make the game more visually appealing.
2. Online Multiplayer: Currently, multiplayer is local. Extending the game to support online play would make it more accessible and competitive.
3. AI Opponent: Implementing a computer-controlled opponent could enhance single-player mode and help players practice.
4. Mobile Port: Adapting the game for Android or iOS would allow wider accessibility.
5. Expanded Testing: Writing more extensive unit and integration tests using pytest would further improve the game's stability and maintainability.

Overall, this project was a significant learning experience. It gave me the opportunity to strengthen my technical skills in Python, GitHub, and automated testing, while also developing my leadership and teamwork abilities. Despite the challenges, the final outcome was a functional and enjoyable Tetris game that reflected the combined effort of the entire team. With future improvements, this project has the potential to evolve into an even more professional and versatile game.