

Higher/Lower Card Game – GUI Implementation:

Overview

The Higher/Lower Card Game is an engaging and innovative project that combines technical rigor with creative problem-solving. Built using C++ and the .NET Framework, it features a user-friendly graphical interface (GUI), a competitive scoring system, and robust file handling for data persistence. Designed with scalability and maintainability in mind, the game leverages core object-oriented programming (OOP) principles—such as encapsulation, inheritance, abstraction, and composition—to create a modular and efficient architecture. Key components, including the Card and Deck classes, handle game logic, while the Game class integrates gameplay, scoring, and user interaction in a seamless workflow.

A research-driven approach informed the development process, focusing on accessibility and user convenience. This led to the inclusion of an executable file for easy distribution, while the C++ source code remains available for future exploration and customization. The GUI dynamically tracks player scores and rankings, fostering replayability through competitive elements. Feedback from classmates, who acted as testers, helped refine these features, making the game an engaging highlight of their day. File handling functionality ensures that player progress is saved and retrieved across sessions, adding to the game's depth and usability.

Development Process

The development process began with comprehensive research and planning to identify the most efficient methods for distribution and implementation. OOP principles were meticulously applied to define a scalable architecture that could accommodate future feature expansions. Each phase of implementation adhered to clean and maintainable coding practices, with abstractions ensuring a clear separation of concerns and reducing code complexity.

Iterative testing played a crucial role in the project's success. Classmates provided insightful feedback on the user experience, leading to meaningful improvements in scoring mechanics and the ranking system. Rigorous debugging and performance testing verified the reliability of features such as file handling and GUI responsiveness. This methodical, feedback-driven approach ensured that the game was both technically sound and user-centric.

Potential Areas for Improvement

While the current implementation successfully delivers a functional and engaging game, feedback from testers highlighted two key areas for enhancement:

1. **Increased Competitiveness:** Introducing a timer for player actions would add urgency and make multiplayer mode more competitive.
2. **Joker Mechanic Revision:** Changing the Joker rule to deduct points rather than end the game would enhance strategy without disrupting gameplay.

Although these features fall outside the current scope, I am eager to implement them in future iterations to further improve the player experience.

Conclusion

This project has been a rewarding journey, offering the opportunity to apply technical skills and creativity in equal measure. From implementing robust OOP principles to refining the game based on user feedback, I have ensured that every aspect of the project reflects attention to detail and a commitment to excellence. I am excited about the chance to bring these skills to Hawk-Eye Innovations and contribute to impactful, forward-thinking projects.