ICGs Platform: Final Report

Group 7: Xuan Yang, Jiete Xue, Jiahao Liu

January 2026

Project Overview

We developed an interactive platform for playing and analyzing **Impartial Combinatorial Games (ICGs)** with automated winning strategy computation. The platform integrates multiple classic ICG games (Nim, Kayles, and two original games) with a graphical user interface, real-time strategy analysis, and an extensible algorithmic framework.

This project bridges theoretical combinatorial game theory with practical implementation, offering educational, research, and AI-development applications. Core theoretical foundations include the Sprague-Grundy theorem and SG function [1].

Background and Challenges

ICG theory is well-established [2], but practical implementation faces significant challenges:

- **State space explosion**: Exponential growth of game states with linear increases in game size.
- **Real-time requirements**: Need for sub-second response in interactive modes.
- **General framework design**: Accommodating diverse and complex game rules.
- **UI/UX design**: Creating an intuitive interface without prior experience.

Our Solution: Framework and Algorithm

We adopted a **three-tier modular architecture**:

1. **Game Logic Layer**: Implements rules and state transitions.
2. **AI Engine Layer**: Optimized recursion with memoization for strategy analysis [3].
3. **User Interface Layer**: PyGame-based interactive graphical interface.

Core Algorithm

We implemented a **recursive depth-first search** with memoization to classify positions as winning (N) or losing (P). The win function is defined as:

$$\mathrm{win}(S) = \begin{cases} \text{False} & \text{if } S \text{ is terminal} \\ -\bigwedge_{S' \in \mathrm{moves}(S)} \mathrm{win}(S') & \text{otherwise} \end{cases}$$

Key optimizations include:

- **State hashing**: Efficient immutable state representation.
- **Memoization**: Caching computed results to avoid redundancy.
- **Early termination**: Pruning search space when possible.

Results and Contributions

Completed Work

- Implemented four ICG games: **Nim**, **Kayles**, and two original games.
- Developed a fully functional PyGame UI with three modes: PvP, PvC, and strategy analysis.
- Achieved real-time response (<1 second) for medium-complexity states.

Team Contributions

- **Jiete Xue**: Core algorithm design, game logic (Nim, Kayles), system architecture.
- **Jiahao Liu**: Test case design, performance testing, state optimization, documentation.
- **Xuan Yang**: UI implementation, interaction flows, integration testing.

Platform Features

- Interactive game selection menu and help guide.
- Real-time move suggestions and position analysis.
- Customizable settings (sound, hints, etc.).
- Modular design for easy addition of new games.

AI Usage Statement

- **Core algorithm**: 100% self made.
- **Architecture**: 70% self made.
- **UI**: 95% AI generated.

Conclusion

We successfully built a functional and extensible ICG platform that demonstrates the practical application of combinatorial game theory. The system provides an educational tool for learning game strategies, a research platform for algorithm testing, and a foundation for future AI integration.

References

[1] Michael H. Albert, Richard J. Nowakowski, and David Wolfe. *Lessons in Play: An Introduction to Combinatorial Game Theory.* A K Peters, Wellesley, Mass., 2007. A widely used introductory textbook that covers basic techniques, impartial games, and the algebra of games[citation:2][citation:5].

[2] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for Your Mathematical Plays*, volume 1–4. A K Peters, Wellesley, Mass., 2nd edition, 2001–2004. The seminal and encyclopedic work that founded modern combinatorial game theory. It is known for its breadth and engaging style[citation:10].

[3] Aaron N. Siegel. *Combinatorial Game Theory.* American Mathematical Society, Providence, Rhode Island, 2013. A comprehensive and modern graduate-level text that covers both classical theory and recent advances like misère quotients[citation:3][citation:8][citation:10].