ICSP Final Presentation

Development of an Interactive Combinatorial Game Platform

X. Yang & J. T. Xue & J. H. Liu

December 31, 2025

7th Group

Contents

Project Overview

Project Overview

Objective
Develop an interactive platform for playing and analyzing Impartial Combinatorial Games (ICGs) with automated winning strategy computation.

Key Features

- Support multiple classic ICG games (Nim, Kayles, etc.)
- Real-time strategy analysis and optimal move suggestions
- Interactive graphical user interface
- Extensible framework for new games

Technical Challenges

Technical Challenges

- **State Space Explosion**: Exponential growth of possible states

Technical Challenges

- **State Space Explosion**: Exponential growth of possible states
- **Real-Time Requirements**: Need for sub-second response time

Technical Challenges

- **State Space Explosion**: Exponential growth of possible states
- **Real-Time Requirements**: Need for sub-second response time
- **General Framework Design**: Supporting diverse game rules

Technical Challenges

- **State Space Explosion**: Exponential growth of possible states
- **Real-Time Requirements**: Need for sub-second response time
- **General Framework Design**: Supporting diverse game rules
- **UI/UX Design**: Creating intuitive interface with no prior experience

Our Solutions

System Architecture

Three-Tier Architecture

- **Game Logic Layer**
  Implements rules and state transitions for each game
- **AI Engine Layer**
  Optimized recursion with memoization for strategy analysis
- **User Interface Layer**
  Pygame-based interactive graphical interface

Modular Design

- Each game implements standardized interfaces
- Core engine independent of specific game logic
- Easy to add new games

Core Algorithm

Optimized Recursion with Memoization

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

Key Optimizations

- **State Hashing**: Efficient state representation
- **Memoization**: Avoid redundant calculations
- **Early Termination**: Prune search space

Performance

- Solves medium states in < 1 second
- Scales to complex game configurations
- Memory efficient

Progress & Results

Completed Work

Implemented Games

- **Nim** with multiple pile configurations
- **Kayles** (bowling pins game)
- **Take coins, Split cards and subtract factors** our original creations
- All with complete rule validation

System Features

- Fully functional graphical UI using Pygame
- Three gameplay modes: PvP, PvC
- Real-time strategy analysis display

Platform Screenshots I

Main Menu Interface

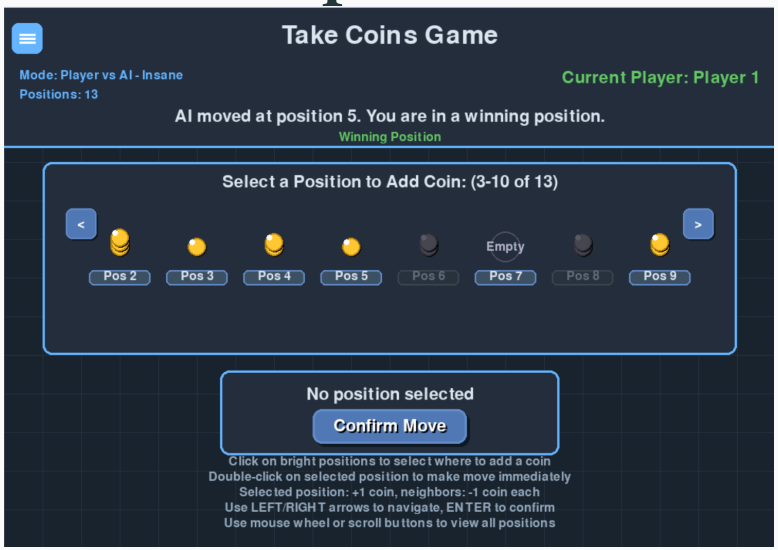
Main menu with game selection and options

Help Guide

Interactive help guide and rule explanation

Platform Screenshots II

Game Example 1



Take coins game interface with move visualization

Game Example 2

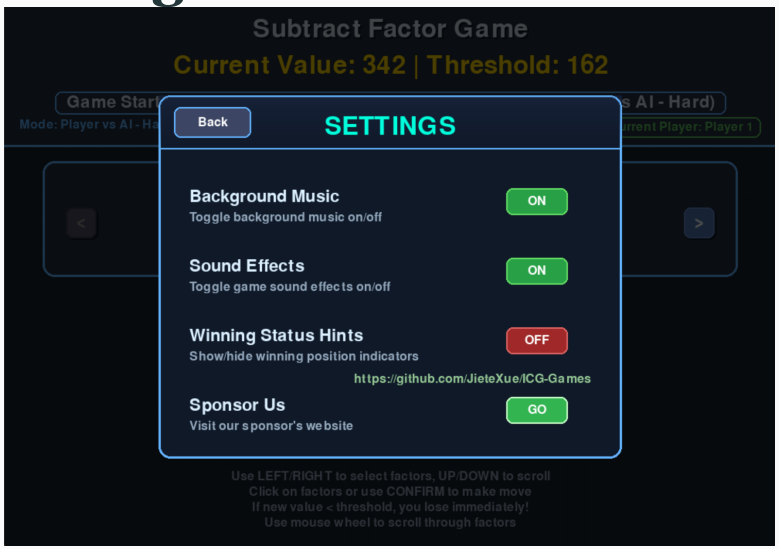
Split cards game interface showing strategy analysis

Platform Screenshots III

Sidebar Functionality

Sidebar with control options

Settings Interface



Settings panel for game customization and preferences

Team Contributions

Member	Contributions
J. T. Xue	Core algorithm design and implementation, Nim & Kayles game logic, system architecture design
J. H. Liu	Test case design, performance testing, state representation optimization, documentation
X. Yang	UI components implementation, user interaction flows, system integration testing

Table 1: Team member contributions and responsibilities

Significance & Applications

Significance & Applications

Educational Value

- Interactive learning of game theory
- Visual demonstration of winning strategies
- Research tool for combinatorial games

Technical Applications

- AI decision-making systems
- Algorithm design patterns
- Framework for game analysis

Research Contribution

Bridges abstract mathematical theory with practical computational implementation, making combinatorial game theory accessible and applicable.

Future Directions

Future Directions

- **Algorithm Optimization**: Parallel computation, heuristic evaluation
- **Game Expansion**: Support for more complex ICG variants
- **AI Integration**: Machine learning for strategy prediction
- **Cloud Deployment**: Web-based platform for broader accessibility

Conclusion

Conclusion

Key Achievements

- Successfully developed a functional ICG analysis platform
- Implemented efficient algorithm using memoization and state hashing
- Created intuitive graphical interface for multiple games
- Demonstrated practical application of combinatorial game theory

The platform provides valuable tools for education, research, and AI development

Questions?

Thank You!