Collapse_5

Game and Group

This project implements the **Collapse** board game using Prolog. The group, **Collapse_5**, from class T01, includes **Alexandre Gonçalves Ramos** (up202208028) and **Francisco Miguel Pires Afonso** (up202208115), each contributing 50%.

Francisco initiated the game logic and started the Computer functionality, while Alexandre refined and finalized the logic, improved the board display, and optimized input handling. Both worked collaboratively, reviewing each other's contributions to deliver a cohesive and polished final product.

Installation and Execution

Prerequisites

- Install **SICStus Prolog 4.9**.

Font Installation

To install the custom font (`font.ttf`):

- **Linux**: Copy the font to `~/.fonts/` and refresh the font cache using `fc-cache`.
- **Windows**: Right-click the font file and choose **Install**.

Game Installation

1. Extract the project ZIP file (`PFL_TP2_T01_Collapse_5.zip`).
2. Navigate to the `src` folder.
3. Launch **SICStus Prolog** and consult the `game.pl` file.
4. Start the game using `play.` if the menu does not load automatically.

For Linux, use terminal commands:

- Extract: `unzip PFL_TP2_T01_Collapse_5.zip`
- Navigate: `cd PFL_TP2_T01_Collapse_5/src`
- Launch SICStus: `sicstus`

For Windows:

- Open SICStus, choose **File > Consult**, and load `game.pl` from the `src` folder.

Description

Collapse is a two-player strategy game played on a **9x5 Fanorona board**. The objective is to force your opponent into a position where they cannot make a valid move. Gameplay revolves around capturing opponent pieces through strategic positioning and movement.

Game Rules

1. **Setup**: Pieces are placed around the board’s perimeter and Players choose to play as **white** (first) or **black** (second). In **Player vs Bot**, the player is always **white**; in **Bot vs Player**, the player is **black**.

2. **Movement**: Pieces move in cardinal (north, east, south, west) or ordinal (northeast, southeast, southwest, northwest) directions. Movement depends on the **rule**: with normal rule, Ordinal moves are limited by the lines of the Faronara Board, on the other side, **Easy Rule**: Ordinal moves are allowed on all cells.

3. **Capturing**: A move must capture exactly one opponent piece and the capturing piece stops one space before the captured piece, which is removed from the board.

4. **Game End**: The game ends when a player has no valid moves or no pieces left and the other player is declared the winner.

Links for Reference

- Collapse Official Page
- Fanorona Board Information

Considerations for Game Extensions

During the development of **Collapse**, we added flexibility with **optional rules** to cater to different skill levels while preserving the game's strategic essence.

1. Normal Rule

Diagonal moves are restricted to cells where the sum of coordinates `(X + Y)` is even, reflecting the Fanorona board's design. This rule adds strategic depth by challenging players to plan moves within these constraints.

2. Easy Rule

Initially, diagonal moves were allowed on any cell, but this was later identified as inconsistent with Fanorona's design. However, the "easy rule" was kept as a beginner-friendly option for simpler gameplay.

Variable Board Sizes

Variable board sizes were considered but rejected. The 9x5 board ensures a balanced, pre-defined piece arrangement crucial to the game's strategy. Larger boards would disrupt the intended gameplay dynamics.

These considerations ensure **Collapse** remains accessible for beginners while offering depth for seasoned players.

Game Logic

The implementation of **Collapse** in Prolog focuses on robust and user-friendly gameplay, with careful design decisions for game configuration, state representation, moves, and user interaction.

Game Configuration Representation

The game configuration includes:

- **Player Types**: `human` or `computer(Level)`.
- **Game Rule**: `1` for Normal Rule (restricted diagonals) or `2` for Easy Rule (unrestricted diagonals).

Example: A Player vs Bot configuration with the Normal Rule is represented as `[human, computer(2), 1]`.

The `initial_state/2` predicate uses this configuration to set up the board, starting player, and selected rule.

Internal Game State Representation

The game state tracks the board, current player, and rule (`game_state(Board, CurrentPlayer, Rule)`)

- **Board**: A 9x5 grid with `black`, `white`, or `empty` cells.
- **CurrentPlayer**: `player1` (White) or `player2` (Black).
- **Rule**: `1` or `2`.

Examples:

1. **Initial State**: `game_state(InitialBoard, player1, 1)`
2. **Intermediate State** (after a few moves):`game_state(UpdatedBoard, player2, 1)`
3. **Final State**: `game_over(game_state(Board, player2, 1), player1)`

Move Representation

Moves are represented as: `move(Row, Col, Dir)`

- **Row**: 1-5 (Piece's row).
- **Col**: 1-9 (Piece's column).
- **Dir**: Direction (`north`, `south`, `east`, `west`, `northeast`, `northwest`, `southeast`, `southwest`).

The `move/3` predicate validates moves, updates the board, and switches players.

User Interaction

Menu System

Players navigate through game modes, rules, and bot difficulties using `display_main_menu/3`.

Input Validation

1. **Moves**:
   - Format: `move(Row, Col, Dir)`.
   - Validated against `valid_moves/2`.
   - Invalid inputs prompt retry.
2. **Menu**:
   - Valid options checked with `member/2`.
   - Invalid inputs trigger a re-prompt.

Error Handling

Recursive calls and `repeat` loops ensure graceful handling of invalid inputs without disrupting gameplay.

Conclusions

Developing **Collapse** provided a valuable opportunity to refine our Prolog skills while implementing a strategic board game. The project achieved its goals of accurately representing the game's mechanics and offering an engaging user experience.

Achievements

1. **Rule Flexibility**: Supporting both **Normal** and **Easy** rules caters to players of varying experience levels.
2. **AI Implementation**: The inclusion of two AI levels (random and strategic) adds depth to solo gameplay.
3. **Seamless Interaction**: A structured menu system and robust input validation ensure smooth gameplay.
4. **Improved Aesthetics**: Custom fonts and board styling enhance the game's visual appeal.

Limitations

1. **Board Size**: The game is limited to the standard 9x5 board, as varying sizes could disrupt balance.
2. **AI Complexity**: The Level 2 AI, while strategic, could be more sophisticated.
3. **Interface**: The text-based interface may not appeal to players accustomed to graphical games.

Future Improvements

1. **Advanced AI**: Implementing algorithms like minimax could enhance AI complexity and challenge.
2. **Graphical UI**: Developing a GUI would improve accessibility and user engagement.
3. **Custom Rules**: Allowing users to define their own rules could increase replayability.
4. **Analytics**: Adding features like move history or win rates could enrich player insights.

Bibliography

Resources Consulted

- **Moodle Slides**: Prolog lecture slides provided on the PFL Moodle platform.
- **Wikipedia**: Fanorona Board Information
- **Game Official Page**: Kanare Abstract - Official Collapse Page

Tools and Assistance

- **ChatGPT (OpenAI)**: Assisted in refining AI logic and suggesting future enhancements.
    - Queries: "Improve Prolog AI logic for strategic gameplay." and "Suggest future features for a Prolog board game."