

Project - SENG 271

Goal: Create a simple board game simulation with Java / Swing. You will learn to apply some of the design patterns (Strategy, MVC, ...) and gain working knowledge of GUI programming with Java Foundation Classes/Swing.

The following project will be carried out in groups of 3-4 students. Groups are assigned and posted on Moodle.

Ludo Game Description

The Ludo board consists of a directed 40 field ring in form of a cross (see Fig. 1).

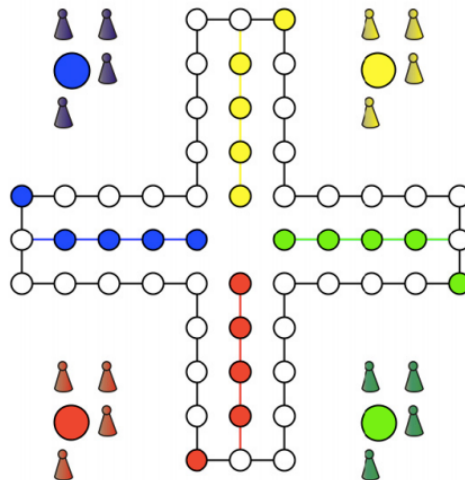


Fig. 1: The Ludo playing board

The rules are as follows:

1. There are four players: traditionally, red, blue, yellow and green. Every player has four pawns, which are not in the game initially (they are "at home").
2. Every 10th field serves as *entry field* for a player. Note that this imposes a cyclic order over the players. In addition, directly preceding each entry field is a junction to four consecutive *goal fields* of the same player.
3. At every point in time, it is the turn of one of the players. Turns rotate according to the cyclic order of players.
4. The player whose turn it is throws a *six-sided die*, and moves one of his pawns according to one of the following rules, if any is applicable. If no rule is applicable, no pawn is moved.
 - 4.1. **Entry:** If the die shows a six and the player still has pawns at home, and the player's entry field is not already occupied by a pawn of his own, he must put one pawn from his home to his entry field.
 - 4.2. **Forward:** If no entry move is possible, the player must select one of his pawns on the board and move it forward by the exact number of eyes on the die. In doing so he may not pass (i.e., overtake) or end on his own entry field (instead he must take the

junction to his goal fields) and may not end on a field that is already occupied by a pawn of his own. Moreover, a forward move may not pass any pawn already on a goal field.

If there is already a pawn (of another player) on the target field of a move, then this pawn is *kicked* and returns to the other player's home.

5. If the die roll was a six, the same player takes another turn; otherwise, the next player (in the given order) gets his turn.

The game ends when one of the players has occupied all his goal fields. This player has won the game.

Strategies

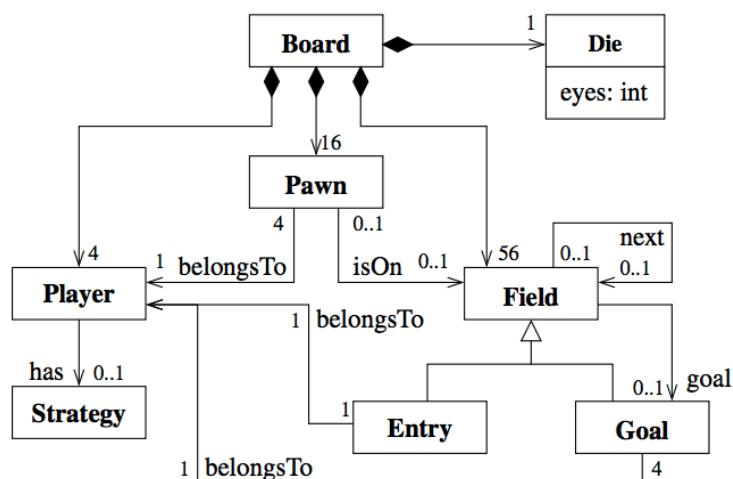
As with any game, an interesting question is to determine strategies for playing that are likely to win the game. Without going into formal game theory, one can easily identify several global strategies (global in the sense that they do not change during the game).

- *Aggressive*: Give preference to a move that kicks a pawn;
- *Cautious*: Give low priority to a move that kicks a pawn (so as not to anger the other player);
- *Defensive*: Give preference to a move to a target field where the pawn cannot be kicked;
- *Move-first*: Give preference to moving the foremost pawn;
- *Move-last*: Give preference to moving the hindmost pawn.

More sophisticated strategies can be defined by taking the moves (or the strategies) of other players into account.

Task description

Your overall goal is to design and implement a simulation for the LUDO board game in Java / Swing. The following class model should get you started with your overall class design.



You are asked to implement the following requirements:

- A. The software should allow a human and computer players. Users can choose the "strategy" of the computer players (see above).
- B. The Java UI should be graphical, based on Swing (Java GUI).

Deliverables - Milestone 1

due: July 5th

- I. Use Case Diagram and textual description
- II. Class diagram of application model classes (not GUI classes)
- III. Sequence diagram of a typical complete round of play (four players), including a textual documentation that explains what happens during that round.
- IV. A “wireframe” mock-up of your GUI design.
- V. A short description of which team member contributed to which deliverable

Deliverables - Milestone 2

due: August 1st

- I. Source code and instructions on how to compile and run.
- II. An update of the models (I-III) submitted in Milestone 1 to reflect the end-result, pointing out any changes that were made.
- III. Demo of project in class (either Aug. 1 or 6th) including a short presentation of the architectural structure and behaviour (class diagram and sequence diagram)
- IV. A short description of which team member contributed to which deliverable

Submit your deliverables in form of a zip archive to the Moodle Web site. (No other archiver please.)

Evaluation

Milestone 1 deliverables will receive 40% of the overall project mark, while milestone 2 deliverables will receive 60%.

Detailed break down:

Milestone 1:

- Use Case model (10%)
- Structural model (Class model) (10%)
- Behavioural model (Sequence diagram) (10%)
- GUI Wireframe (10%)
- (The last submission item will not be formally graded, but only used to check for anomalies, such as a group member not significantly contributing to the project.)

Milestone 2:

- Source code and instructions (30%)
- Model updates (20%)
- Presentation (10%)
- (The last submission item will not be formally graded, but only used to check for anomalies, such as a group member not significantly contributing to the project.)