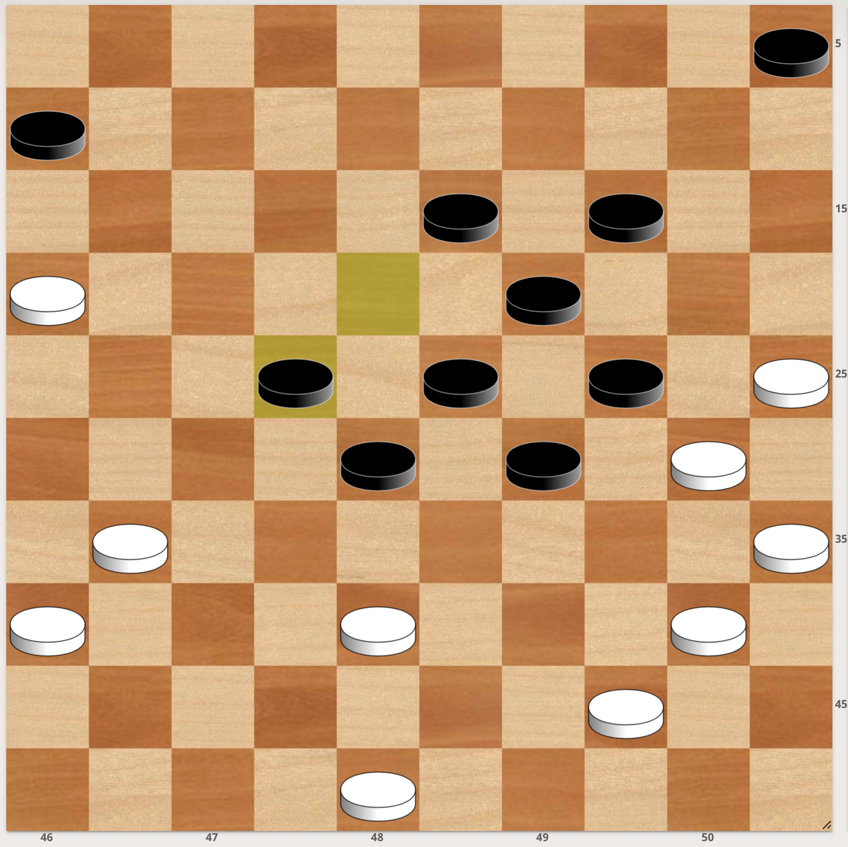
Architecture Design and Implementation of Draughts

# Goal

The goal to design and to implement an **object-oriented** software architecture for a **console-based Draughts** game between **two human players**. This includes following all best practices of object-orientation.

Base your entire work on the rule set used by the lidraughts site (cf. References).



1. Design a user interface for your game. Decide in particular about the following points as it may have a huge impact on other design aspects:
   1. How is the playing field and the respective men and kings (men / king = term for the two types of pieces in the draughts game) going to be presented to the user?
   2. How is the user going to enter moves?

Note that the coordinates of the figure above are not mandatory. You are allowed to define a different coordinate system, if you wish to do so.

1. Use ***UML class diagrams*** to design the internal structure of the program. Do not only show the classes/interfaces and their interconnections, but also the methods and most relevant attributes.
2. Use **UML activity and sequence diagrams** to model the internal steps of the most important methods. Include in particular the methods that allow players to move during each phase of the game.
3. Add any other UML diagrams that you deem relevant. (optional)
4. Implement your game in **C++**. The use of an engine or any libraries beside the standard C++ library is not allowed. It is, however, allowed to use the standard library and, for example, its container classes.

**Your UML diagrams and implementation must match. ☺**

**It is recommended, but not mandatory to leverage the FEN notation, which is basically a string that allows for storing the current game state.**

# Deliverable

**Two deliverables** will be due **15 June 2025**:

1. All your UML diagrams
2. The source code of your implementation

Submit both as a .zip file on Moodle.

Finally, there is going to be a **third deliverable for the day of the exam**. Prepare a **slide show** including your UML diagrams as well as explanations of all major design decisions. Highlight in particular which design patterns have been used. Also prepare a **demonstration** of your game prototype. Make sure to start with the demonstration. You will have **15 minutes for both**.

**The evaluation will be based on all deliverables.**

# References

* Overview of the Rules

<https://lidraughts.org/variant/standard>

* Class Diagrams:   
  <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>
* Activity Diagrams:   
  <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
* Sequence Diagrams:   
  <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
* Sequence Diagram vs Activity Diagram:

<https://www.geeksforgeeks.org/difference-between-sequence-diagram-and-activity-diagram/>