Team NASA

•••

INF112

2nd obligatory assignment

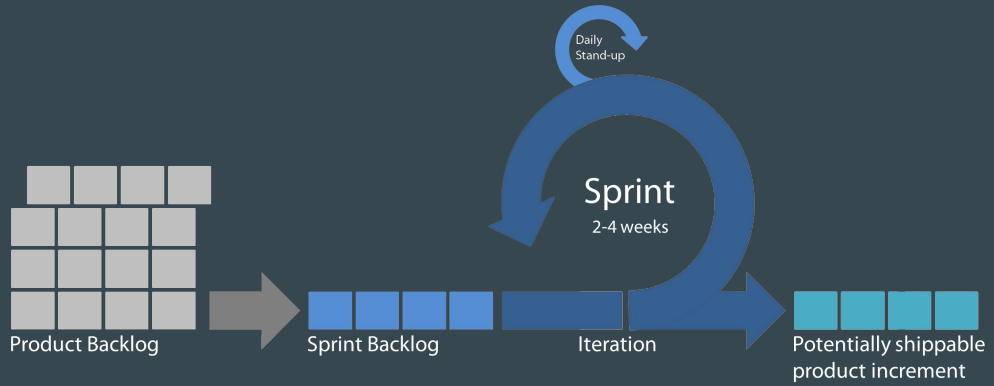
Agenda

- Development process and results
  - Process and development style
  - Rules
  - Product specification:
    - Graphics
    - Modelling
    - System requirements
- Challenges and solutions
- Retrospect

Process and development style

- Slack, Google Drive, Facebook

- Scrum - agile methods
  - Smaller groups
  - Group meetings
  - Review

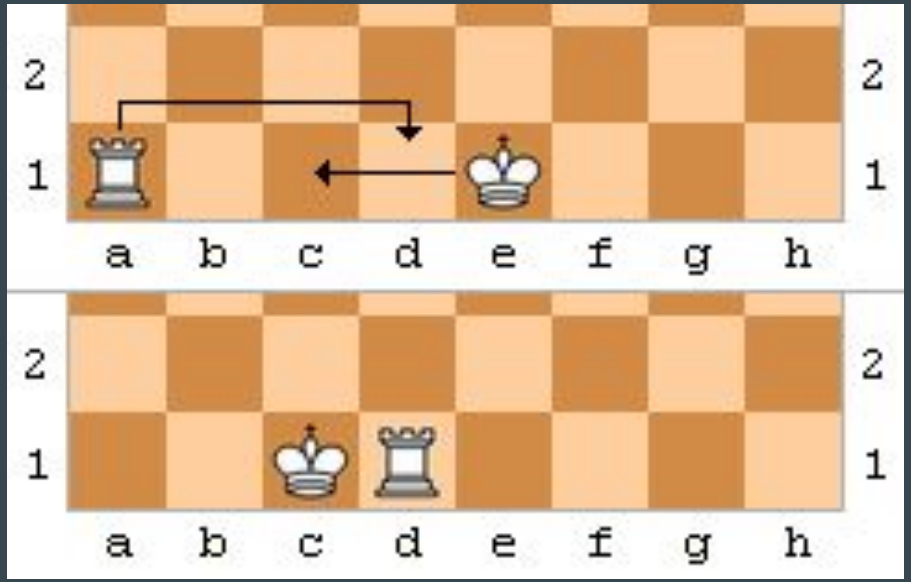


- Git -> two branches
  - dev
  - master

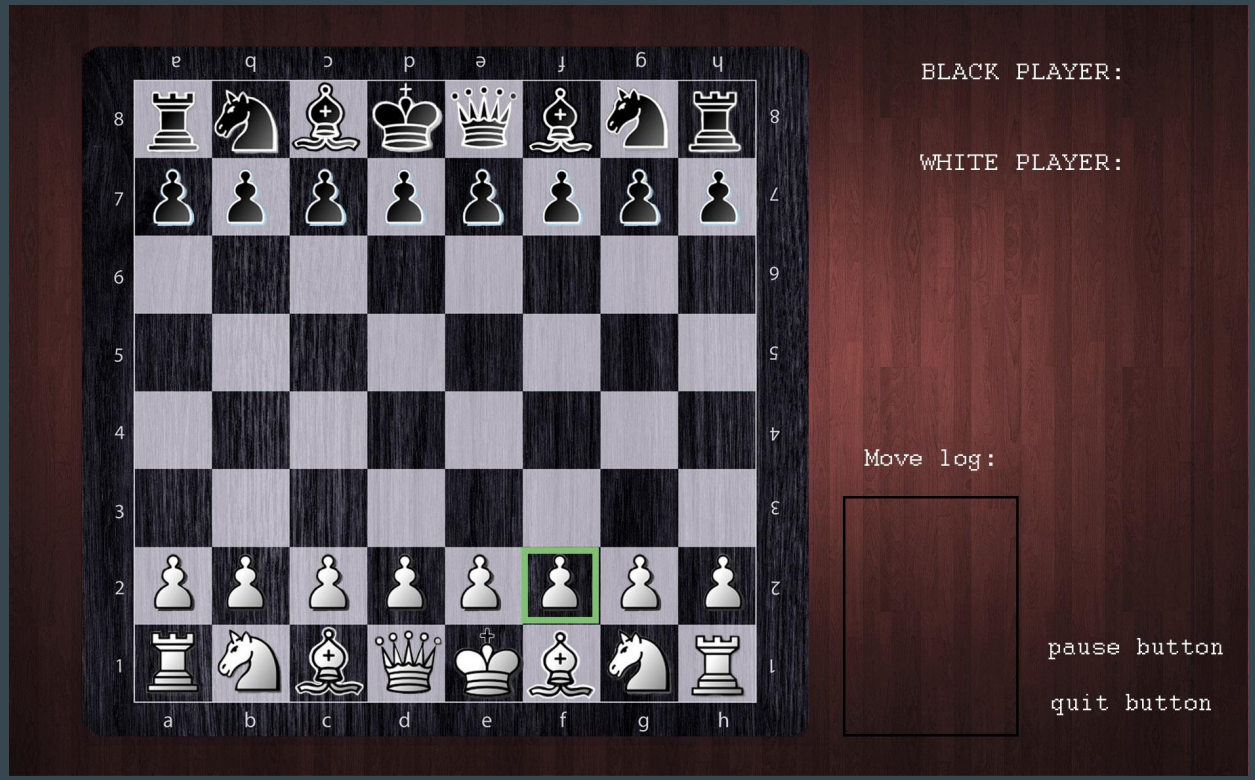
GitLab

Rules

- This iteration: Focus on traditional rules
- Onwards: Easily expandable code



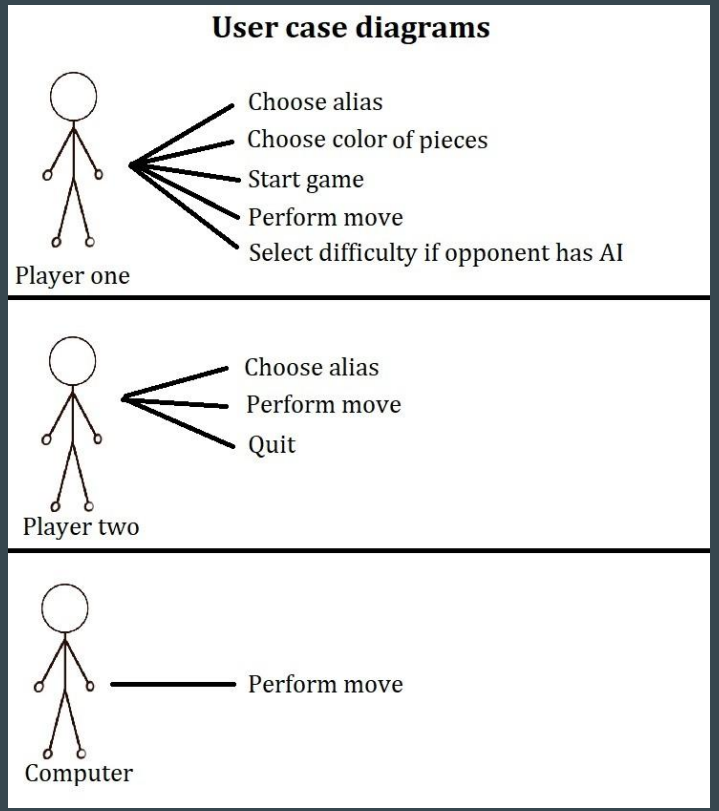
Graphics suggestion



Interactive modeling: user case diagrams

- Abstraction of interactions and user operations
- Often used early in a development process

- Advantages:
  - Identify user requirements
  - Communication



Interactive modeling: fully dressed user case

- Purpose?

- Cases:
  1. Multi player
  2. Single player (AI)
- Structure
  - Main Success Scenario
  - Extensions

Main Success Scenario

Use Case Name: Playing multi player game
Scope: Chess Game
Primary actor: Player
Stakeholders and interests:
- Player: Wants to play a game with an opponent. Only legal moves. Score should be saved. Time is kept track of.
- Opponent: Wants to play a game against player. Only legal moves. Score should be saved. Time is kept track of.
- High score holders: All scores are safely kept track of.

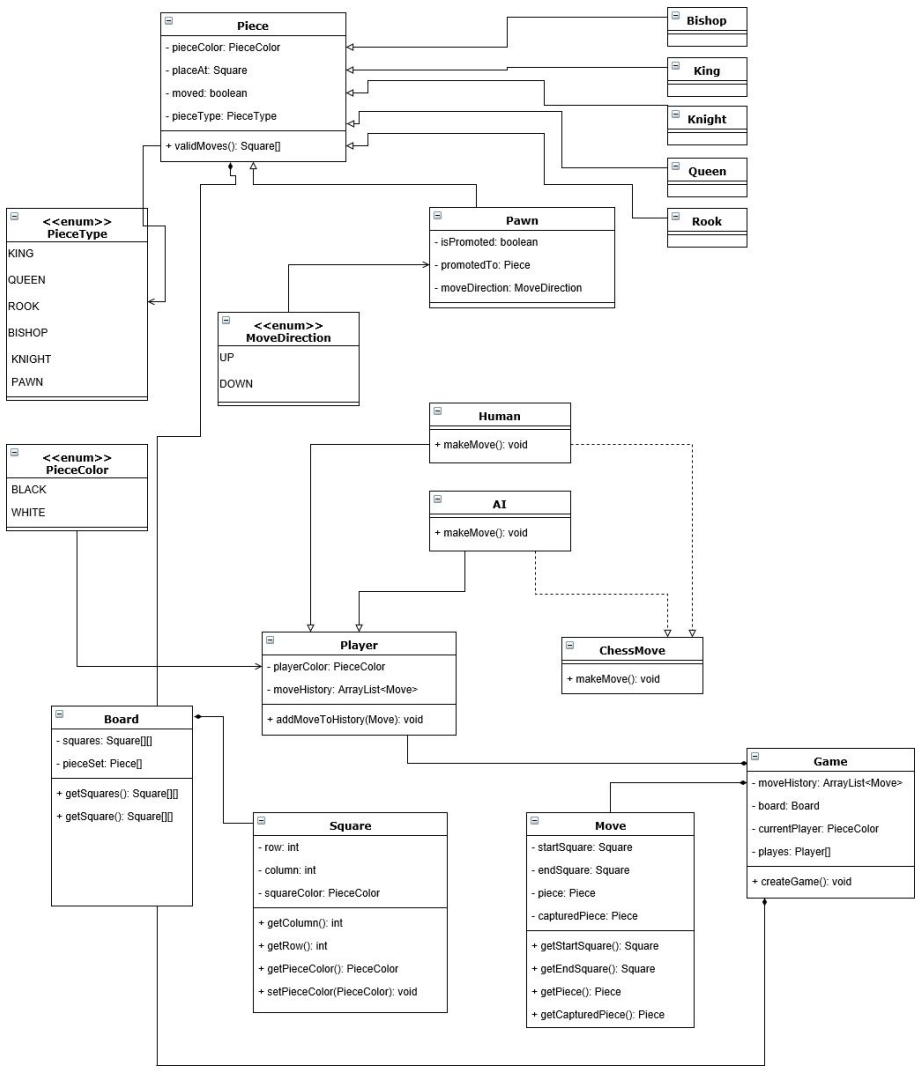
Precondition: Player has access to game, and is identified if in high score list.
Postcondition: Player has played a game, score is recorded in system.

Extensions

6a Time runs out for Player/Opponent

6a1 The Player with time left wins. Game is finished and scores are saved. Go to step 9

Structual modeling → class diagram



System Requirements - Scope

Scope:

- Multiplayer/Single player
- 3 Intelligence level
- 2D Chess board - standard chess rules
- Each piece with valid moves
- Necessary notifications
- Records each moves and results of each games

System Requirements - Functional

1. **Getting ready for a chess game:**
    - Multiplayer / Single player option
    - Single player - 3 Levels of intelligence
    - User registration
    - Selecting the color of the pieces
    - Displaying the chess board

System Requirements - Functional

2 . Playing the chess game

- Enable the moves of each chess piece (Pawn, knight, bishop, rook, queen and King)
- Enable castling, en-passant moves.
- Displaying the valid moves on selection of pieces.
- Notification for invalid moves.
- Last move UNDO option.
- Notification for check, stalemate and checkmate(winner)
- Time keeping for each move.
- Displaying the captured pieces.

System Requirements - Functional

3. Ranking of players

- Points calculation based on games win and intelligence level.
- Recording the results of each game played.
- Ranking of players based on the points they earned.
- Quit the game option in case of boredom.

System Requirements - Non-Functional

- Reliability
- Performance
- Response time.
- Maintainability
- Scalability

Challenges and solutions

- How to organize the team and communicate in an efficient manner
- How to properly define and distribute tasks

- Scrum methodology combined with communication tools such as Slack
- Inclusion of all team members and even workload
- Frequent meetings - tasks defined collectively and distributed

Retrospective

- What went well?
- Planned adjustments of the next iteration:
  - Stricter quality control
  - Class diagram