

CS7330 Object Oriented Design I

Assignment #3 Checkers Game Data Model Concept

Huanhuan Xia, 02/12/2018, 14283803

1. Description

This assignment is to create a data model concept for the game of Checkers, called Draughts game. The data model is a representation of this game as data and methods, without any reference to UI elements. Aiming at gaining experience with creating data models.

2. Design

a. User Movements

- i. Users are asked to enter the players' name.
- ii. Users are asked to select game modes. There are three types of game modes, automatic, single player or two players.
- iii. Users are asked to select colors of pieces. The side with darker piece should make the move first.

b. Game Board Design

- i. The game board is designed as a 2-D array, with 8 columns and 8 rows.
- ii. Each element in the array represents a grid. Each grid has a 2D coordinate (x, y) , where $0 \leq x \leq 7, 0 \leq y \leq 7$.
- iii. To prevent illegal moves on one specific grid (x_0, y_0) , if $(x_0 + y_0) \% 2 == 0$, this grid is in red color, it should never be reached or initially put pieces on.
- iv. Each side begins the game with 12 pieces, within the area of three rows that are closer to him/her.

c. Coordination of Move and Jump

- i. If a lighter piece reaches a grid on the top of the game board, it will be a red king.
- ii. Vice versa, if a darker piece reaches a grid on the bottom of the game board, it will be a black king.
- iii. The table below shows the conditions of moves and jumps:

Possible Moves	Lighter Piece	Darker Piece	King
Move	$(x - 1, y - 1)$ $(x - 1, y + 1)$	$(x + 1, y - 1)$ $(x + 1, y + 1)$	$(x - 1, y + 1)$ $(x + 1, y - 1)$ $(x - 1, y - 1)$ $(x + 1, y + 1)$
Jump (if enemy pieces are on the neighbor diagonal grids)	$(x - 2, y - 2)$ $(x - 2, y + 2)$	$(x + 2, y - 2)$ $(x + 2, y + 2)$	$(x - 2, y + 2)$ $(x + 2, y - 2)$ $(x - 2, y - 2)$ $(x + 2, y + 2)$
Else	Cannot move or jump		

- d. Regulations of Moves or Jumps
 - i. All the possible moves and jumps should locate with the game board.
 - ii. If the grid that one side wants to move to or jump to already has a piece on it, it is an illegal move or jump.
 - iii. If another jump is available for one of the pieces, that jump must be taken. If more jumps are available for the same piece, continue jumping until no available jumping is detected. To satisfy this regulation, check recursively if there is another jump after this jump.
- e. Decisions of Winner
 - i. When there is no lighter piece exists, the darker side wins.
 - ii. When there is no darker piece exists, the lighter side wins.
 - iii. If there is no winner after 50 rounds, there is a draw.

3. UML Figure

