Rationale for each chosen domain and their relationships

Player

Reason for choosing this domain

Since both HumanPlayer and ComputerPlayer will later have the same attributes (such as an arrayList of tokens) and methods (such as the basic moves of moving the token), Player is introduced as a parent class for both HumanPlayer and ComputerPlayer. It defines the standard for all the actions any type of player can take during the game, hence, Player will be defined as an abstract class so that there is no repetition of code in the HumanPlayer and ComputerPlayer classes. Having an abstract super class for Player, ensures that each type of player can have its own implementation details within the same domain.

Reason for the relationships

- Player-Move : Player can use Move to move their tokens at each turn to any valid position.
- Player-Token : Each Player has a total of 9 tokens at the start of the game. It might be reduced slowly one by one to a minimum of 2 tokens in which the game will end.

HumanPlayer

Reason for choosing this domain

The 9 Men's Morris game requires at least one HumanPlayer for it to be runnable. The HumanPlayer domain defines the implementation of any behaviours that any user of the game does

Reason for the relationships

- HumanPlayer-Player : HumanPlayer is chosen to be a child class to the Player in order to prevent repetition of items in multiple player classes. It extends from the abstract Player class.
- HumanPlayer-Game : A Game will have at least one HumanPlayer (ComputerPlayer Mode) and a maximum of 2 (HumanPlayer Vs HumanPlayer) depending on the game mode chosen.

ComputerPlayer (Advanced Feature)

Reason for choosing this domain

The user can select to play against a bot whose actions and behavioural implementation is defined by the ComputerPlayer domain.

Reason for the relationships

- ComputerPlayer-Player : ComputerPlayer is chosen to be a child class to the Player in order to prevent repetition of items in multiple player classes. It extends from the abstract Player class.
- ComputerPlayer-Game : A Game might have a ComputerPlayer is a user chooses to play with a bot.

Game

Reason for choosing this domain

Reason for the relationships

-

MessageBox

Reason for choosing this domain

A message box is required mainly for 2 functions. Message box is displayed when a mill is formed by either of the players including the computer as an indication for the player to remove a token from the opponent. Other than that, message box is displayed when the game ends.

Reason for the relationships

- Game-MessageBox : The Game will display a minimum of 8 messages (7 mills minimum to win a game formed by the same player + 1 winning message).

Token

Reason for choosing this domain

Token is the main element in the game for the players to move.

Reason for the relationships

- Token-MovablePositionFinder : Each Token when being selected will have to check for its movable positions before a move is made on it.

MovablePositionFinder

Reason for choosing this domain

MovablePositionFinder is used to locate the empty positions that a selected token can be moved to considering the types of Move valid at the point in game (Sliding,Placing,Flying).

Reason for the relationships

- MovablePositionFinder-Position : For sliding, MovablePositionFinder check through all the adjacent Position(minimum of 2) on the board if they are occupied.In the case of flying and placing, will have to check through all the 24 Position to see if they are valid for placement of tokens.

Position

Reason for choosing this domain

Position is required in the game to form the basic design structure of 9 Men’s Morris’s board so that tokens can be placed.

Reason for the relationships
Position-Token : Each Position can have either 0 or 1 Token at any point in the game.

Board
Reason for choosing this domain
The Board is the place where the positions are located so that all the tokens can move and where ultimately, the actual game will take place.

Reason for the relationships
- Board-Position : The Board has 24 Position where a token could be placed. Board and Position relationship is chosen to be represented as Composition since Position is part of the Board. Without the Board, the Position will not even exist as they are supposed to be placed on the Board at specific locations according to the the 9 Men's Morris's board design.

Display
Reason for choosing this domain
The game uses the Display to show the current state of the board.

Reason for the relationships
- No Relationship with any other domains

Move
Reason for choosing this domain

Reason for the relationships
- Move-MillChecker : After every move of sliding,placing and flying, the MillChecker checks if a mill has been formed or in the case of removing, it is used ensure that the token removed by player is not part of a mill . Since all the subclasses (PlacingMove ,SlidingMove ,RemoveMove ,FlyingMove) of the Move will eventually have to use the same implementation of MillChecker, the MillChecker is directly associated with abstract class Move instead of all the individual subclasses.
- Move-Position : Move updates the Position of a token when it is repositioned.
- Move-Token : Move changes the positions of Token

SlidingMove
Reason for choosing this domain
This move provides a way for tokens to move to adjacent positions, which is the basis for the game itself. SlidingMove is enabled in the game after all 18 tokens from the two players are placed on the board.

Reason for the relationships
SlidingMove-Move :

FlyingMove

Reason for choosing this domain

This move provides a way for tokens to move to any unoccupied position on the board. FlyingMove is enabled when players have three or less tokens can move their tokens to any currently unoccupied position on the board.

Reason for the relationships

FlyingMove-Move :

PlacingMove

Reason for choosing this domain

This is a way for tokens to be placed on the board during the start of the game until all 18 tokens from the two players are all placed.

Reason for the relationships

PlacingMove-Move :

RemoveMove

Reason for choosing this domain

This is a way for the tokens to be removed from the board after a mill is formed. A player that has formed a mill will get to remove one of the opponent's tokens.

Reason for the relationships

RemoveMove-Move :

MillChecker

Reason for choosing this domain

The MilChecker checks all positions to find if a mil exists.

Reason for the relationships

- MillChecker-Position : The MillChecker go through all 24 positions to find a straight row of 3 occupied positions aligned along the same board line.