Escape

Alpha & Beta Release Design

Isabel Morales Sirgo

CS 4233

Alpha

Board

For the first release of the assignment we were tasked with handling the instantiation of multiple types of boards that hold a specific type of coordinate. Both Square and OrthoSquare boards are made up of squares, however I decided to make them separate classes since they each have different movement behaviour.

I added some private static methods within these board classes to handle boar instantiation errors. For Square and OrthoSquare I ensure that the coordinates given are within the board's defined limits. For Hex boards this method was not implemented since boards are infinite.

Coordinate

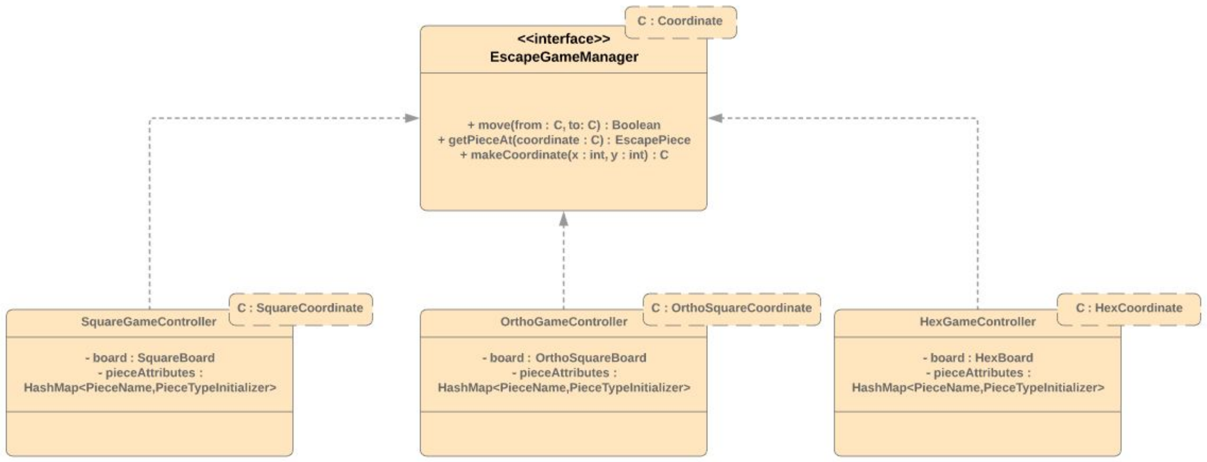
Additionally, I created 3 different coordinate classes, each implementing the coordinate interface. I made this decision to simplify the implementation of the distanceTo() method that has different behaviour for each game board.

Beta

Game Manager

The game supports three different types of boards. In my design I implemented 3 different classes to act as the corresponding game manager for each of the board types. I decided to implement these classes as controllers. The client will be able to send commands directly to the controller and the controller will handle the specific requests internally.

The structure of the controllers is shown below. The controller holds the game board as a private attribute. When instantiated the controller creates and initializes the board using the information provided by the game XML file. The corresponding controller is returned when calling the makeGameManager() method.



C : Coordinate
<<interface>>
EscapeGameManager
+ move(from : C, to: C) : Boolean
+ getPieceAt(coordinate : C) : EscapePiece
+ makeCoordinate(x : int, y : int) : C
SquareGameController
C : SquareCoordinate
- board : SquareBoard
- pieceAttributes :
HashMap<PieceName,PieceTypeInitializer>
OrthoGameController
C : OrthoSquareCoordinate
- board : OrthoSquareBoard
- pieceAttributes :
HashMap<PieceName,PieceTypeInitializer>
HexGameController
C : HexCoordinate
- board : HexBoard
- pieceAttributes :
HashMap<PieceName,PieceTypeInitializer>

Path Finding & Game Rules

Given my implementation of the game managers, I wanted to keep all of the external pathfinding logic separate from the move method. Therefore I implemented a pathfinding class within the escape.rules package. This class holds a canMove() static method that takes in the specific piece attributes and current game board. Within this class all movement type variations are handled as well as any specific piece attributes. A depth first search is performed eliminating any paths that are invalid for the current piece. This method will return true as soon as a single valid path is found.

My initial intention was for this class to be able to handle both Board and Coordinate generics. However we were not allowed to modify the interfaces for this assignment. Therefore I had to cast to the corresponding Board and Coordinate types to be able to use any of my additional methods within these subclasses. This limitation led me to implement three pathfinding classes, one corresponding to each of the game boards.