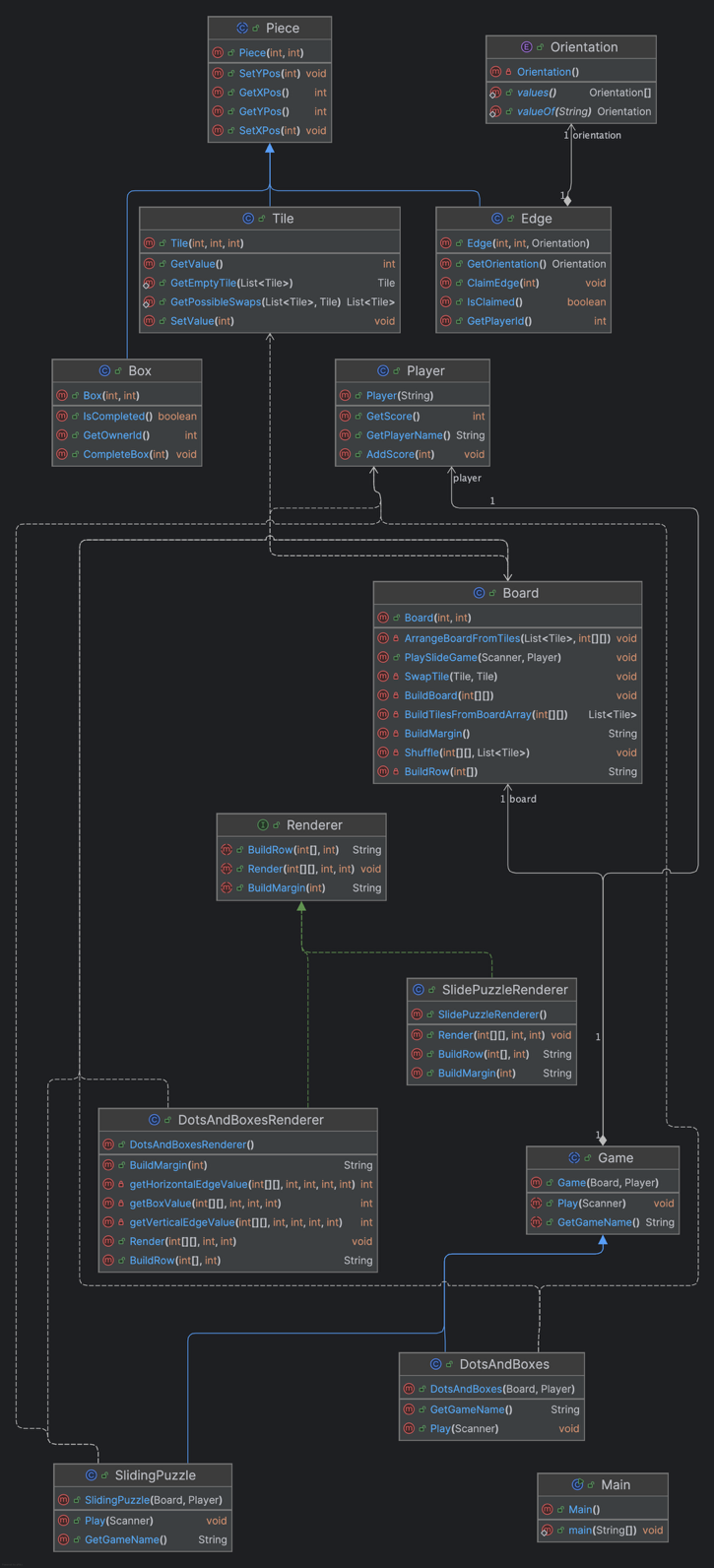
ARCADE DESIGN DOC

UML Diagram



Scalability and Extendibility

My arcade project is scalable because you can adjust the width and the height of the board to be whatever height the user enters. There is a minimum height and width of 2. If the user enters a height or width below the minimum height or width of 2, my program sets the height or width to a default value which is 3 for both. It also would write a friendly message to the console saying there was an invalid value for height or width and it’s setting the value to the default.

The puzzle is also extendable because many games will have a board, a player and play method. Additionally, the boards are always rendered and have rows and margins, which the Renderer interface has methods for doing. There is a SlidePuzzleRenderer and DotAndBoxesRenderer which implement this interface. Additionally, I have a Piece class which has an x-position and y-position. The Tile, Edge, Box class all inherit from this. This means if I wanted to extend this project by adding more, I would simply need to add another class that inherits from Game. If I render a board, it can implement the Renderer interface. If I have anything in the game that has a specific x- or y-position, I would create another class that extends the Piece class.

Changes October 5, 2025

* Added more classes for extendibility
* Slide game now has player name and keeps track of number of moves a player makes
* Dots and Boxes game is now implemented