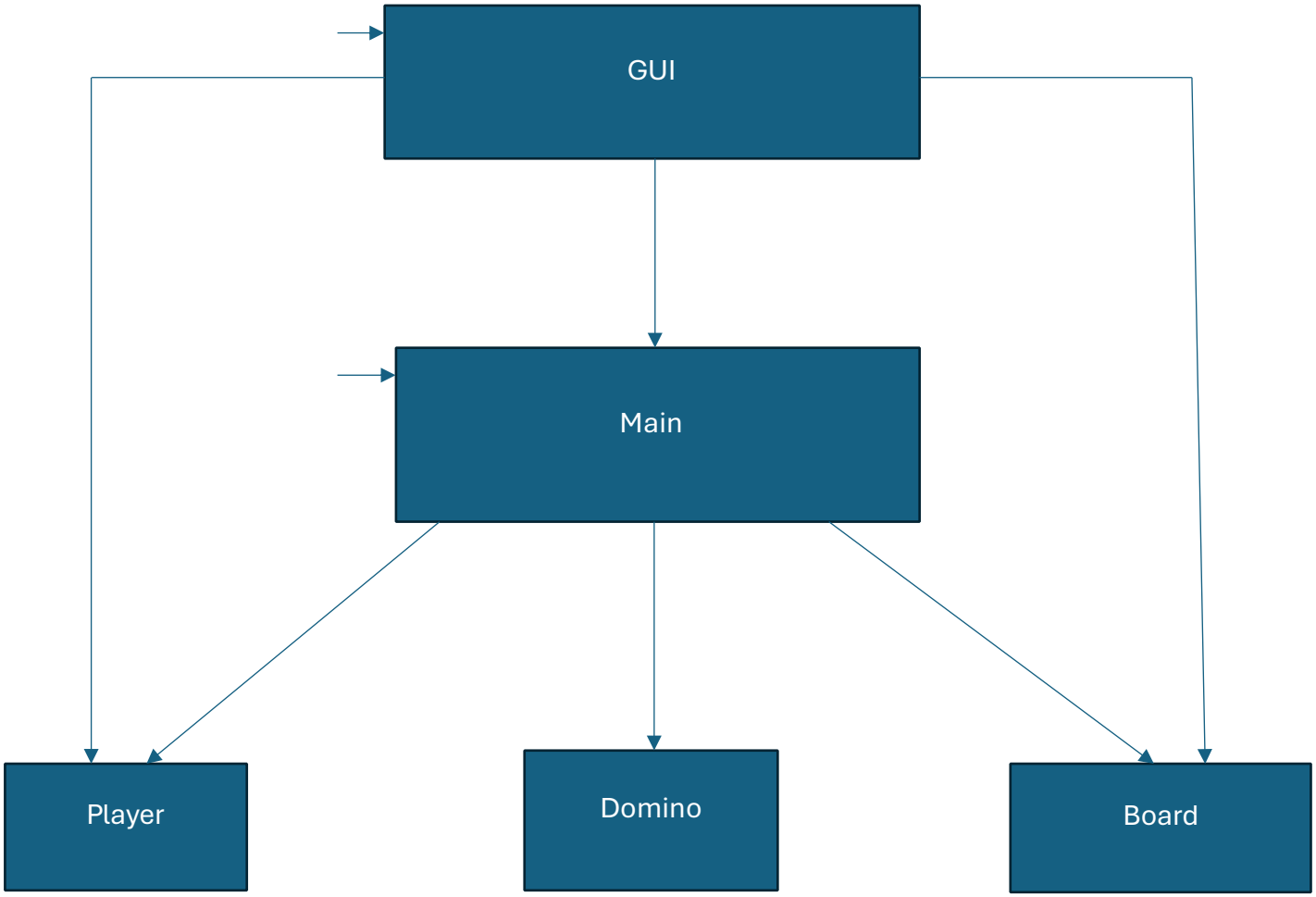
Object Diagram



Description:

GUI: The GUI class extends JavaFX's Application and serves as the main entry point for the domino game's user interface. It sets up a BorderPane-based layout that organizes the game board at the center, the player's tray at the bottom left, and option panels and game counts on the sides and top. The class manages key game components like the Board and Players and creates buttons, labels, and image views to handle user interactions such as playing dominoes and drawing from the

boneyard, as well as managing computer moves and pop-up notifications. It also processes an optional command-line argument to configure the maximum number of dots in the domino set.

Main: The Main class orchestrates the gameplay for the Domino game. It initializes the game board and creates human and computer players, then distributes a fixed number of dominoes to each player. During the game loop, it alternates turns between the human and computer, displaying the current board and player trays, handling human input for playing or drawing dominoes, and automating the computer's moves. It continuously checks for valid plays and, when neither player can move, it determines the winner based on whose tray is empty or by comparing the total remaining dots.

Board: The Board class manages the game board by handling the boneyard and played dominoes. It initializes, shuffles, and distributes dominoes based on a set maximum dot count. It provides methods to draw and place dominoes while maintaining the game state, including retrieving available dominoes, played sequences, and boneyard size.

Domino: The Domino class models a single domino tile with two numbered sides. It is constructed with specific dot counts for the left and right sides, and includes a method to rotate the domino by swapping these values. Getter methods provide access to the dot counts on each side, and the overridden toString method returns a neatly formatted representation of the tile. This class forms the fundamental unit for building and manipulating domino sets in the game.

Player: The Player class represents an individual participant in the domino game—either a human or a computer, as defined by the Players enum. It maintains a tray, which is an ArrayList of Domino objects, that holds the dominoes the player currently owns. The class provides methods to add a domino to the tray, remove a domino by its index, and retrieve either the entire tray or a specific domino from it. This design allows the game to manage each player's set of dominoes independently during gameplay.