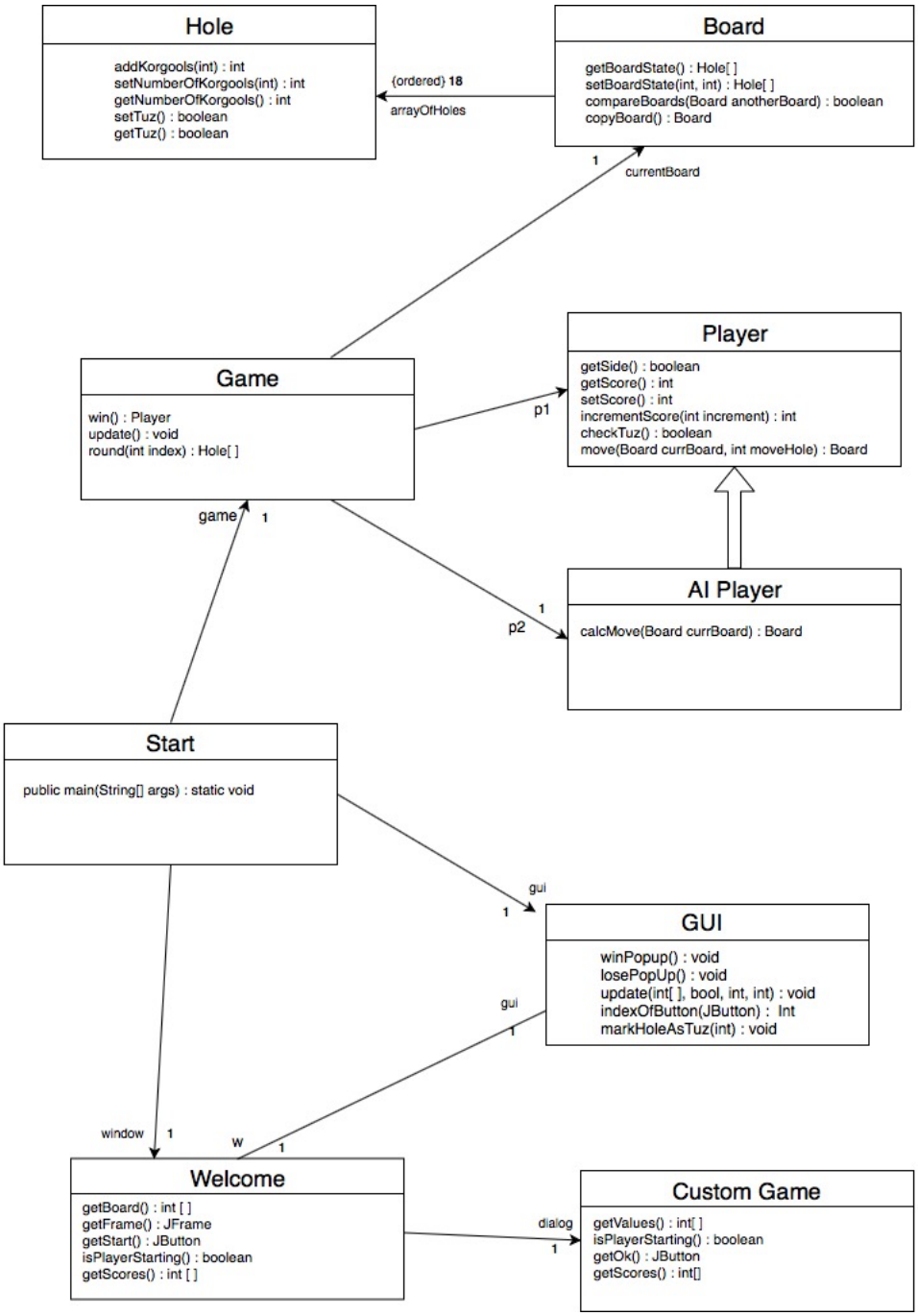
Design – Toguz Korgool Game

Class Diagram



Our application consists of nine classes:

Hole – The Hole class contains an array of integers which correspond to the number of korgools in the hole. It consists of methods to get and set the number of korgools in the hole and also to set and identify the hole as a tuz.

Board – The Board class contains its state - an array of exactly 18 holes. The holes which have an index from 0 to 8 in the array belong to player one, and those which have an index from 9 to 17 belong to player two. It contains methods to get a board state and also to set a board state. You can also call methods to copy a board state and to compare two board states.

Player - The Player class represents a player in the game. A player possesses and access their side, has methods to get and to set their own score, and can check if they have a tuz. The player class also contains a method called “move” which distributes korgools to different holes on the board depending on the hole chosen by the player.

AIPlayer – The AIPlayer class extends the Player class. Consequently, it has the same methods, moreover, it has a method called “calcMove” which executes its move from a randomly chosen, valid hole.

Game – The Game class creates a new board, one player, and one AI player, and decides which side starts. It has methods to play a round of the game called “round” and updates the board and score using the “update” method. It also has a method called “win” to check if the game has been won by any player.

Welcome – This class creates the initial frame/screen of the application and creates buttons to play a custom game or standard/default game. A custom game allows the user to set the initial board state. Clicking the “OK” button will either call the GUI class or the CustomGame class in order to construct the corresponding user interface.

GUI – The GUI class creates the in-game interface. It creates buttons which correspond to the holes in the game and labels each button with a number which corresponds to the number of korgools in the respective Hole. The background consists of two colours in order to differentiate between the two sides. All methods are private, and it calls these methods on construction.

Custom Game – The user can set the board state using the interface provided by this class. It allows the user to input the number of korgools to be set in each hole of the board consistent with the Toguz Korgool rules. The user can also decide who starts the game and the initial score for each player.

Start – The Start class launches the game and calls methods from the Welcome class and GUI class to display the user interface. The start class resides in-between the back end and front-end components of the application – it accesses the game data (the score, the board state etc) and passes this information on to the GUI classes so that they can be displayed to the user.

Another class we implemented was the InvalidMoveException which just extends Exception. It is called whenever a player tries to execute an invalid move.