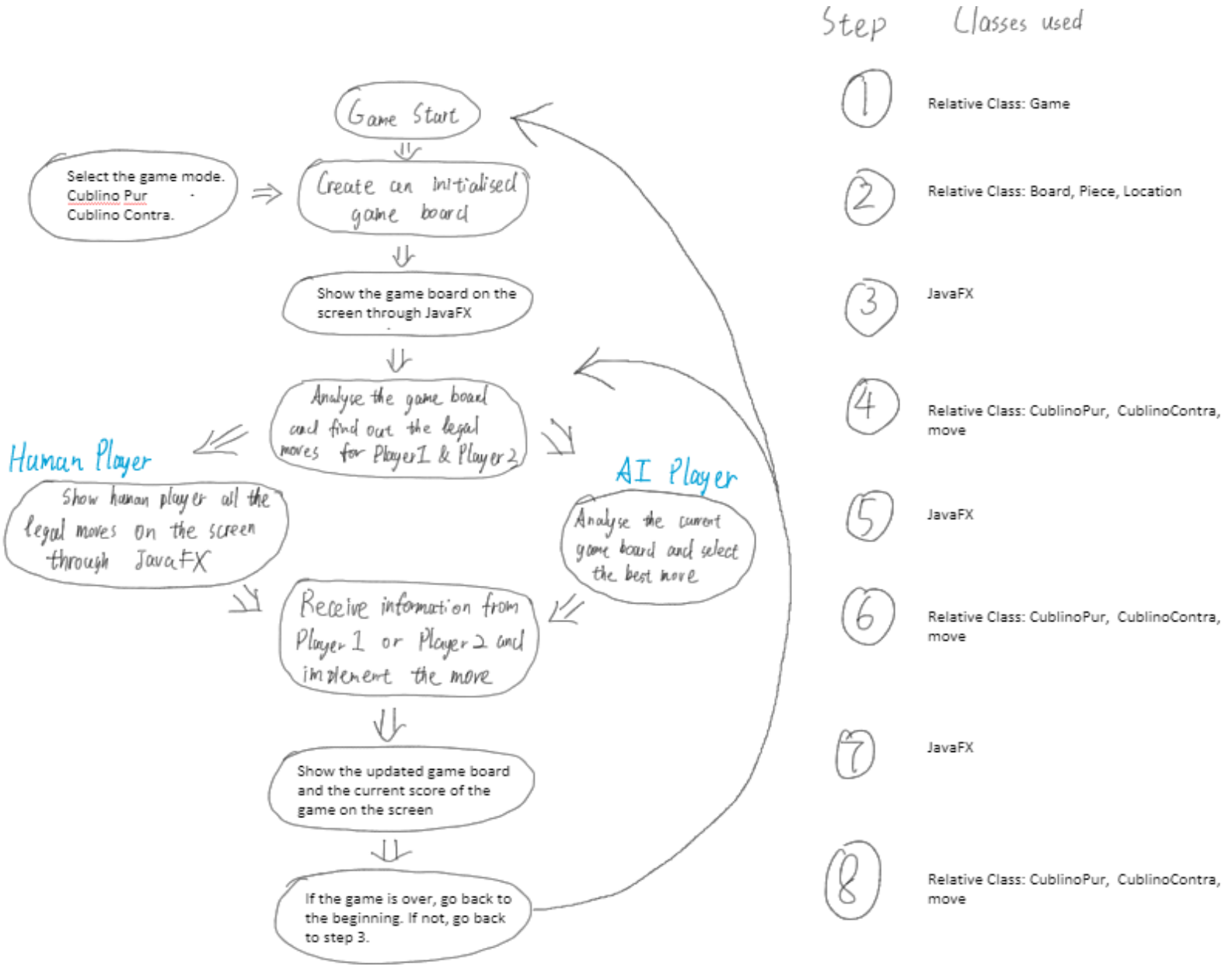
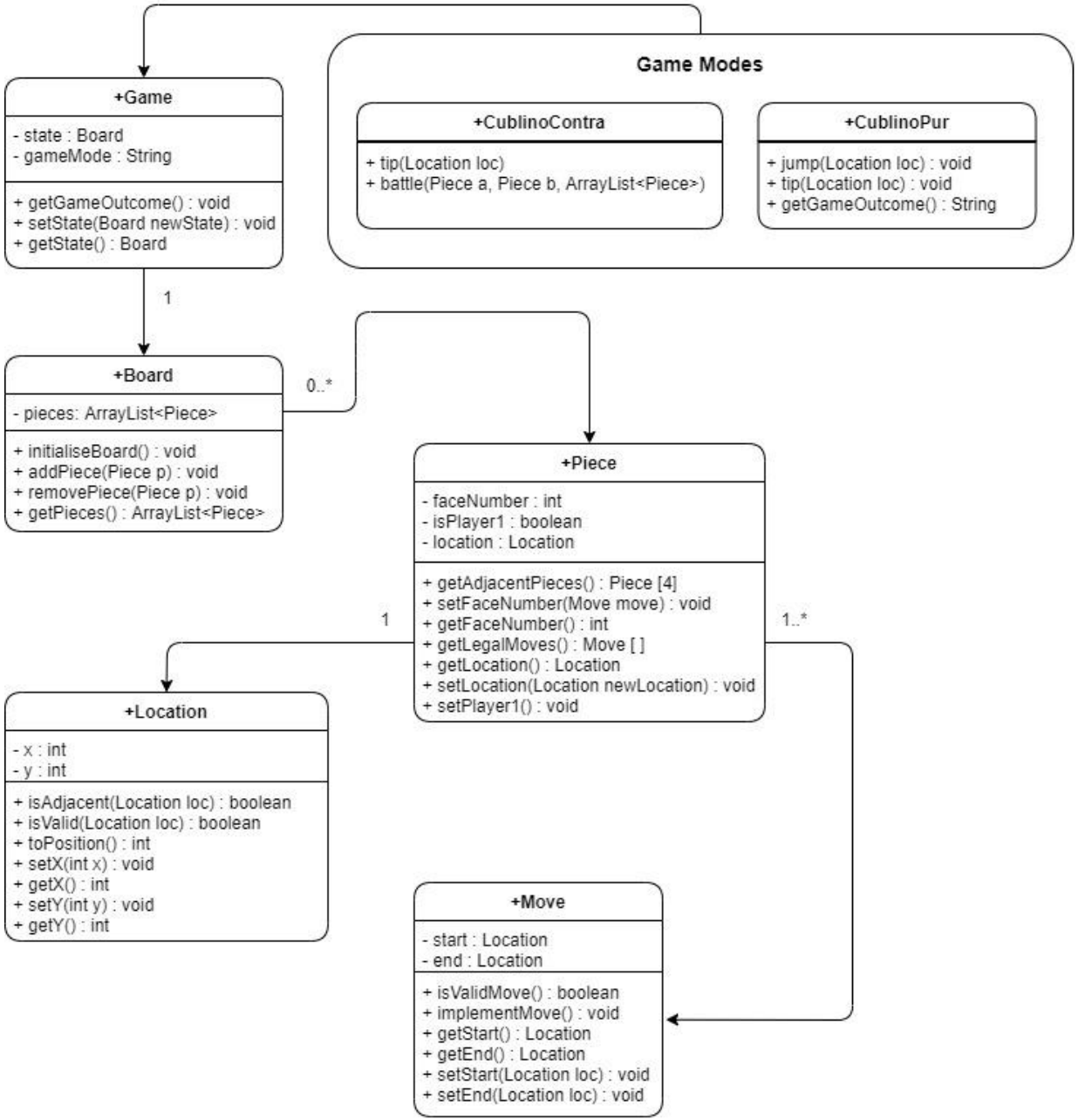
Assignment 2 - Design Document

Basic Overview of Game Implementation Logic



Java Class Relationship Diagram



Basic Classes' Specifications

Location

- **Fields:**
  - Int x
  - Int y
- **Methods:**
  - isAdjacent(Location loc) *// checks if a location is adjacent*
  - isValid (Location loc) *// checks if a location is valid*
  - toPosition() *// returns an integer value for the current coordinate*
  - *Setters and getters*

Piece

- **Fields:**
  - Int *// The current number facing up, default: 6.*
  - (Final) Boolean isPlayer1 *// checks if piece in player 1*
  - Location *// current location of the piece*
- **Methods:**
  - getAdjacentPieces() *// gets an array of pieces around current piece*
  - setFaceNumber(Move m1) *// changes face number of piece*
  - getFaceNumber()
  - getLegalMoves() *// gets all possible moves/jumps (depends on the gamemode) for the current piece*
  - Setters and getters

Game

- **Fields:**
  - Board *// current state of the game*
  - String *// stores the current game mode*
- **Methods:**
  - isGameOver() *// game is over or draw*
  - Setters and getters

The game starts with creating an initialised board.

Board

- **Fields:**
  - Piece [ ] *// pieces on the board*
- **Methods:**
  - initialisedBoard() initialise a board
  - Setters and getters

Move

- **Fields:**
  - Location start
  - Location end
- **Methods:**
  - isMoveValid()
  - implementMove()
  - Setters and getters

CublinoPur

A class containing methods of implementing a Cublino Pur move

- **Methods:**
  - jump(Location loc) *// output a new location after jumping*
  - tip(Location loc) *// output a new location after tipping*
  - getGameOutcome() *// gets the game outcome (win, loss, draw)*

CublinoContra

A class containing methods of implementing a Cublino Contra move

- **Methods:**
  - tip(Location loc) *// output a new location after tipping*
  - battle(piece, piece, pieces[]) *// remove or keep a piece after a battle and update piece position*
