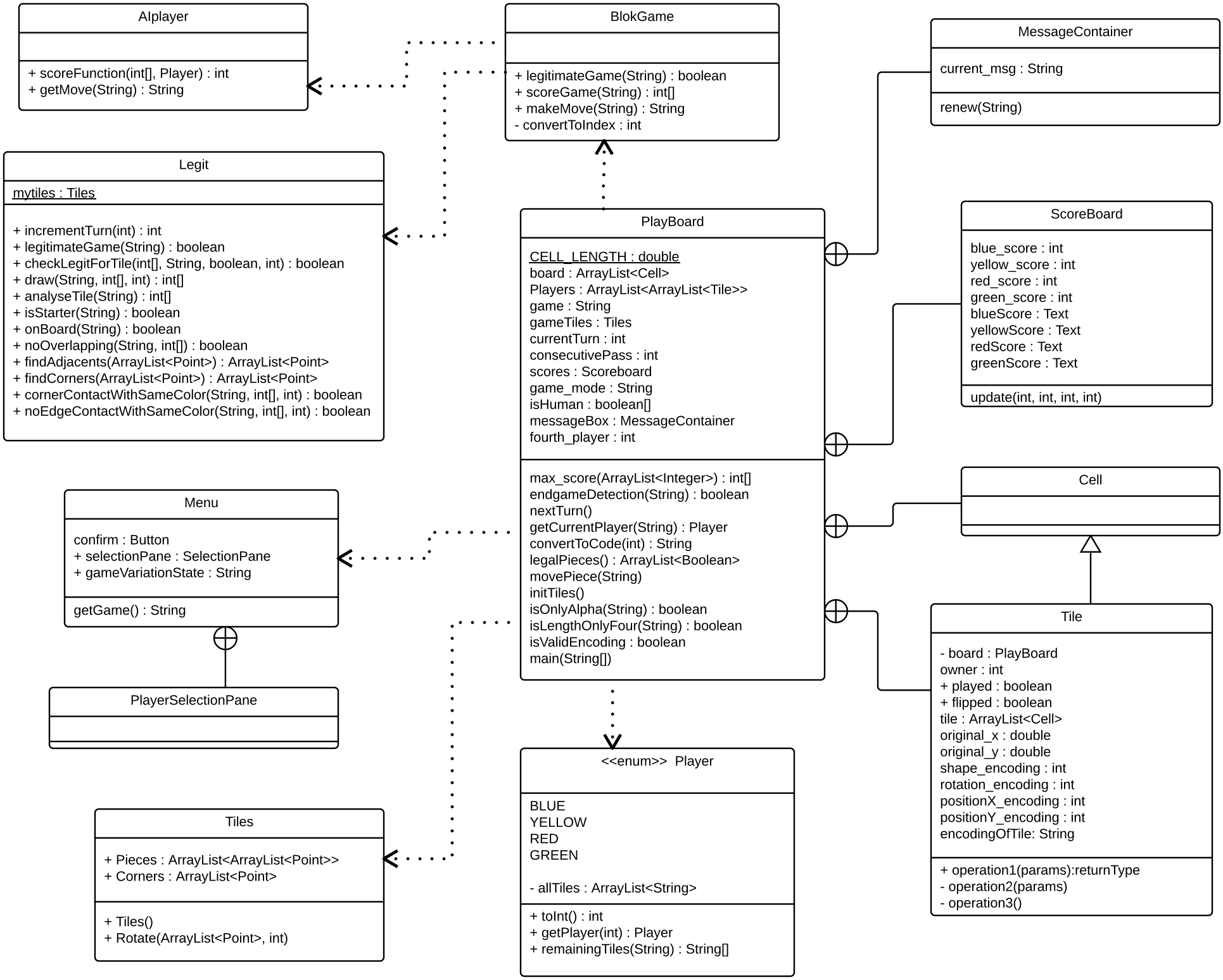
AIplayer
+ scoreFunction(int[], Player) : int
+ getMove(String) : String
BlokGame
+ legitimateGame(String) : boolean
+ scoreGame(String) : int[]
+ makeMove(String) : String
- convertToIndex : int
MessageContainer
current_msg : String
renew(String)
Legit
mytiles : Tiles
+ incrementTurn(int) : int
+ legitimateGame(String) : boolean
+ checkLegitForTile(int[], String, boolean, int) : boolean
+ draw(String, int[], int) : int[]
+ analyseTile(String) : int[]
+ isStarter(String) : boolean
+ onBoard(String) : boolean
+ noOverlapping(String, int[]) : boolean
+ findAdjacents(ArrayList<Point>) : ArrayList<Point>
+ findCorners(ArrayList<Point>) : ArrayList<Point>
+ cornerContactWithSameColor(String, int[], int) : boolean
+ noEdgeContactWithSameColor(String, int[], int) : boolean
PlayBoard
CELL_LENGTH : double
board : ArrayList<Cell>
Players : ArrayList<ArrayList<Tile>>
game : String
gameTiles : Tiles
currentTurn : int
consecutivePass : int
scores : Scoreboard
game_mode : String
isHuman : boolean[]
messageBox : MessageContainer
fourth_player : int
max_score(ArrayList<Integer>) : int[]
endgameDetection(String) : boolean
nextTurn()
getCurrentPlayer(String) : Player
convertToCode(int) : String
legalPieces() : ArrayList<Boolean>
movePiece(String)
initTiles()
isOnlyAlpha(String) : boolean
isLengthOnlyFour(String) : boolean
isValidEncoding : boolean
main(String[])
ScoreBoard
blue_score : int
yellow_score : int
red_score : int
green_score : int
blueScore : Text
yellowScore : Text
redScore : Text
greenScore : Text
update(int, int, int, int)
Cell
Menu
confirm : Button
+ selectionPane : SelectionPane
+ gameVariationState : String
getGame() : String
PlayerSelectionPane
Tile
- board : PlayBoard
owner : int
+ played : boolean
+ flipped : boolean
tile : ArrayList<Cell>
original_x : double
original_y : double
shape_encoding : int
rotation_encoding : int
positionX_encoding : int
positionY_encoding : int
encodingOfTile: String
+ operation1(params):returnType
- operation2(params)
- operation3()
<<enum>> Player
BLUE
YELLOW
RED
GREEN
- allTiles : ArrayList<String>
+ toInt() : int
+ getPlayer(int) : Player
+ remainingTiles(String) : String[]
Tiles
+ Pieces : ArrayList<ArrayList<Point>>
+ Corners : ArrayList<Point>
+ Tiles()
+ Rotate(ArrayList<Point>, int)