Criterion B – Design

Design of the Solution

Two objects:

- Piece
- Board

Board has the following properties:

- contains the location of pieces on the board that have not been eliminated
- a board can be constructed with no pieces on it, or with the state of another board.

Methods: (Mutators)

“addPiece” – adds a piece to an unoccupied location on the board.

“removePiece” – removes the piece currently at the specified location on the board.

“movePiece” – moves the piece from location1 to location2 and attacks the piece at the destination, if there is one.

(Accessors)

“getPiece” – returns the piece at the specified location of the board, if there the piece exists.

“getBoard” – returns the location of all pieces on the board.

Piece has the following properties:

- knows its rank
- knows its side that it belongs to
- knows the location of the image that represents the piece
- knows its name. Ex. Engineer
- knows its uniqueID (a system used to distinguish between multiple pieces of the same rank on the same side)
- a piece can be constructed given its side, rank, and uniqueID

Methods:

“compareTo” – used when 2 pieces collide, determines whether the attacking piece gets eliminated.

“equals” – determines whether 2 pieces are equal to one another

“getID” – returns the uniqueID

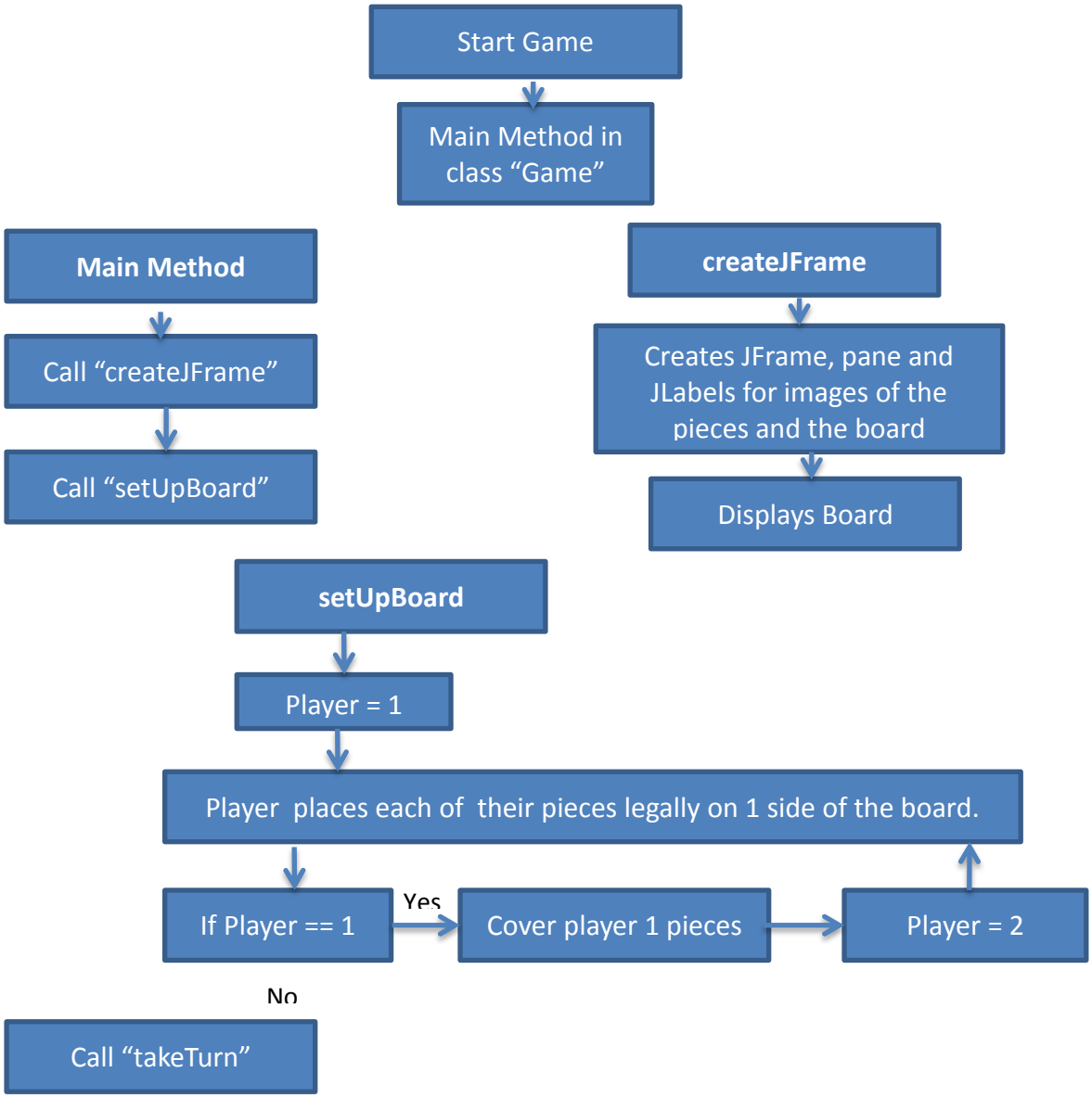
“getImageLocation” – returns the name of the image file

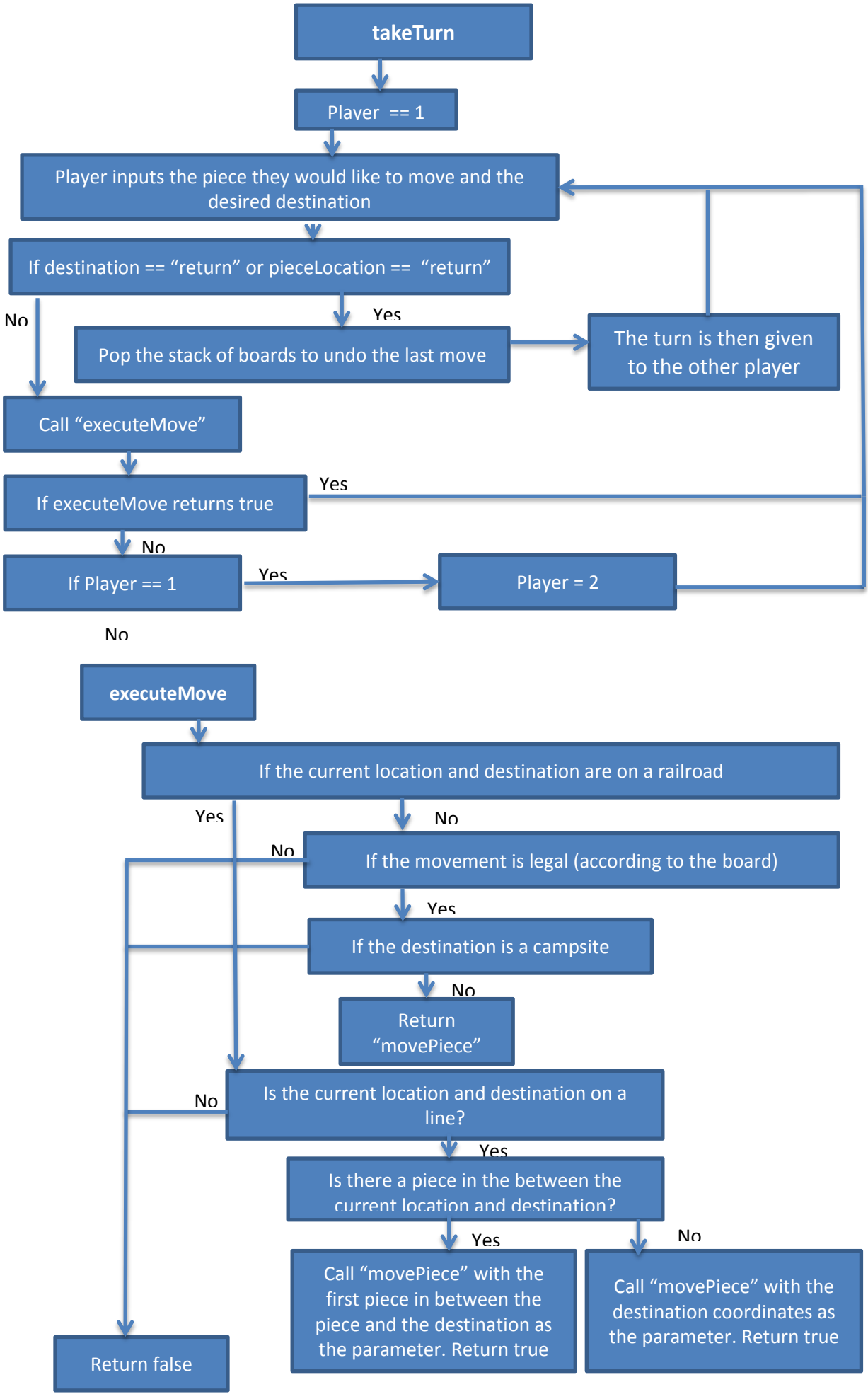
“getName” – returns the name of the piece

“getRank” – returns the rank of the piece

“getSide” – returns the side of the piece

“toString” – returns a string representation of the piece




takeTurn
Player == 1
Player inputs the piece they would like to move and the desired destination
If destination == “return” or pieceLocation == “return”
No
Yes
Pop the stack of boards to undo the last move
The turn is then given to the other player
Call “executeMove”
If executeMove returns true
Yes
No
If Player == 1
Yes
Player = 2
No
executeMove
If the current location and destination are on a railroad
Yes
No
No
If the movement is legal (according to the board)
Yes
If the destination is a campsite
No
Return “movePiece”
No
Is the current location and destination on a line?
Yes
Is there a piece in the between the current location and destination?
Yes
No
Call “movePiece” with the first piece in between the piece and the destination as the parameter. Return true
Call “movePiece” with the destination coordinates as the parameter. Return true
Return false

These tests need to be performed by playing the game several times

Action to test a function of the game	**Methods of testing result and evidence that the test is successful**
Players place their pieces legally	Play the game and place pieces. Flags are can only be placed on headquarters, landmines only on the last 2 rows of one side of the board. Other pieces cannot be placed on the campsite.
Incorrect input coordinates are detected and players have an opportunity to input the information again.	Input any miscellaneous string, out of bounds coordinates, coordinates of locations that are already occupied by an allied piece, or no input nothing at all. This should results in the player inputting their information again, until it is valid.
Each move is saved correctly and can be undone	Input "return" in the field where the destination or location of the current piece is required. This should results in the last movement of pieces being undone and the turn given back to the last player who made a turn.
Opponent pieces are undisclosed	Play the game and notice transition periods between turns. During every transition period where players switch looking at the screen, a neutral board where all pieces are covered is displayed. During a player's turn,

	only pieces belonging to the player is displayed. The screen is also cleared after each player sets up the positions of their pieces.
Winner is announced when the opponent flag is captured or bombed	Play and finish the game. When a side's flag is captured or even bombed, after the player's turn is finished, the console should indicate the side that has won and there will be no more turn taking after that. The game is over.
The correct screen is displayed after every move	Set up a side of the board and take turns playing the game. After every movement of a piece or the placement of a piece, the display changes to reflect the screen change.
There is an accurate determination of which piece(s) gets eliminated when pieces collide	Input valid coordinates of a red side piece and a blue side piece. Observe and verify which piece has been eliminated. The piece with the lower rank should be eliminated, with the exception of engineers (rank 0) who eliminate landmines (rank 9) and bombs.
Interface is representation of the board showing each piece in its intended position	Play the game, take turns and place pieces. The image of the pieces and board should be updated correctly after every piece placement or move.