Contract CO1: joinGame

Operation: joinGame(playerName)

Cross References: Use Cases: Join Game

Preconditions
1. Number of players < 2
2. Player name is valid

Postconditions
1. One more player added into the game
2. Total number of players in the game increases by one

Contract CO2: playGame

Operation: playGame( )

Cross References: Use Cases: In Game

Preconditions
1. 2 <= Number of players <5
2. Board is ready for the game
3. Tiles are ready for the game
4. Deck is ready for the game
5. Players are ready for the game

Postconditions
1. GameOn is set to true

Contract CO3: placeTile

Operation: placeTile(location, rotation)

Cross References: Use Cases: In Game

Preconditions
1. GameOn is true
2. Deck is not empty
3. Previous placement is not valid

Postconditions
1. Return the validity of this tile placement

Contract CO4: placeMeeple

Operation: placeMeeple(location)

Cross References: Use Cases: In Game

Preconditions
1. GameOn is true
2. Deck is not empty
3. Previous placement is not valid

Postconditions
1. Return the validity of this meeple placement

Contract CO5: getScore

Operation: getScore( )

Cross References: Use Cases: In Game

Preconditions
1. GameOn is True
2. Placement of tile is valid
3. Placement of meeple is valid

Postconditions
1. Score for this move is computed

Contract CO6: getResult

Operation: getResult( )

Cross References: Use Cases: After Game

Preconditions
1. GameOn is false
2. Deck is empty

Postconditions
1. Winner is obtained