Interaction: The Minotaur player attempts to move its worker.

Operation: moveWorker(newPosistion: BoardPosition)

Preconditions:

1. Valid worker: currentWorker is not null.
2. Valid Game state: Game is in the 'MOVE' state, not in any other game state.
3. Current player turn: It must be the turn of the current player who owns the worker identified by worker's workerID.
4. Valid player action state: It must be the 'MOVE' action state, not 'BUILD'.
5. Valid worker selection: The Minotaur worker is owned by the current player.
6. Valid godCard and modifyMoveValidation: The godCard value is not null, and godCard's modifyMoveValidation should return true.
7. Valid move distance: the newPosition must be in neighboring spaces to the Minotaur worker's current position.
8. Incomplete Towers: The newPosition must not contain a complete tower (i.e., a tower with a dome)Occupied.
9. Valid target square: If the target square is unoccupied, do the standard move. Else if the target square is occupied, it must be occupied by the opponent player's worker.
10. Valid behind square: If the target move square is occupied by the opponent player's worker, the square one space straight backward can be at any level, but has to be unoccupied, not out of bounds, also not having a dome.
11. Level difference constraint: the level of the newPosition is at most one level higher than the Minotaur worker's current position, but lower levels are all fine.

Postconditions:

1. Worker's position updated: Minotaur worker is now in newPosition.
2. Square occupancy updated: the square at the new position is now occupied by this Minotaur worker, and the square at the previous position is no longer occupied.
3. Win condition checked: if the Minotaur worker moves from two-level to three-level, check for a win condition for this Minotaur worker's player.
4. Opponent worker updated: if the Minotaur worker performs a push, the pushed opponent worker now should be on the square that one space straight backward.
5. Updated Game State: If the move results in a win condition, the game state should be 'GAME_OVER' and declare the winner. Else the game state should turn to 'BUILD'.
6. Turn Action updated: If the move does not result in a win condition, then the player action turns to the 'BUILD' phase.s