Use Case	Make A Move
Primary Actor	Player
Stakeholder & Interest	• Player- Selecting the robots and the directions for them to move
Pre-conditions	The player has placed the smallest bid among all the players.
Post-conditions	The player moves the robot to its desired destination (marked region).
Main Flows	1. The system provides an opportunity for the user to select a robot to move. 2. The user selects the robot to move. 3. The system requests the user to select the direction for the robot to move. 4. The user selects the direction where they want the robot to move. 5. The system checks if it's a valid move [Alt 1: The move isn't valid] 6. The system confirms the validity of the move and moves the robot in the chosen direction. 7. The system increments the number of moves by the user [Alt 2: The number of moves exceeds the bid stated] 8. The system provides the user with the opportunity to make further moves. [Alt 3: The user chooses to make another move] or exit the use case [Use Case Ends].
Alternative Flows	Alt 1: The move is not valid. 1. The system informs the user that the chosen move isn't valid and instructs the user to select another square. Alt 2: The user has made more moves than the stated bid. 1. The system informs the user that they have exceeded the stated bid 2. The system returns the robots that were moved back onto their position markers

	3. The system chooses the user with the next lowest bid 4. Flow resumes at main flow 1 Alt 3: The user chooses to make another move. 1. Flow resumes at main flow 1
Exception	• If the program is closed, the system asks the user if they want to save the game or quit. If the user decides to save the game, the system will save game progress and the use case ends. Otherwise, the use case ends without saving.
Special Requirements	• Ensuring that each player makes their move within a reasonable time limit
Open Cases	• How do we determine an appropriate time limit for each move to keep game flow consistent