Use Case Name: "Take a turn"

Primary Actor: User(s)

Stakeholders and Interests:

-The player should be presented with a clearly defined board and know what color robot piece belongs to them.

-The player should know when it's their turn to play

-The player should be able to maneuver their game piece around the board

Preconditions:

-The board should be set up in a random configuration of 4 sections, 4 robot pieces and target squares all distributed randomly. Each of the robot pieces should be placed on the board such that they do not start on a target square.

-A target square has been selected

-All human players are aware of which target square has been selected

-The system knows the location of all robot pieces

Success Guarantee:

-The robot piece belonging to the player whose turn it is has moved to the square as a result of the path chosen

-Results of the selected path (number of turns) is recorded for the player whose turn it is

-The system indicates it is the turn of the next player or end of the round

Main Success Scenario:

1. The system indicates whose turn it is and gives control to the user to move their robot piece *[Alt 1: The user would like to save the game and exit; Alt 2: The user would like to reset the game; Alt 3: The user decides to exit the game]*
2. The player whose turn it is selects a path they presume is the shortest path by means of up, down, left, and right directions until a target square is reached
3. The system moves the player's piece across the board, following the user's steps
4. The system keeps track and records of the number of times the robot piece changes direction based on the path chosen
5. Once the target square is reached, the system ends the user's turn and the user can no longer move their robot piece
6. The system records the new location of the robot piece
7. The system gives the results (number of steps) of the player's turn. *Use case ends*

Alternate Flows:

Alt 1:

-Save the game:

1. The user selects to save the game's progress and exit the game
2. The system requests the user confirm their choice and notifies the user that all progress will saved until the user elects to load a saved game
3. The user elects to save the game
4. The system automatically saves all progress: location of robot pieces, tally of scores, location of tokens and exits the game. *Use case ends*

Alt 2:

-Reset the game:

1. The user selects to reset the game.
2. The system asks the user to confirm their selection and notifies the user that all progress will be lost if they choose to continue
3. The user confirms their choice
4. The system resets the game configuration and all progress is lost. *Use case ends*

Alt 3:

-Exit the game

1. The user selects to exit the game
2. The system asks the user to confirm their choice and notifies the user that all progress will be lost if they choose to continue
3. The user confirms their selection
4. The system ends the game, losing all progress. *Use case ends*

Exceptions:

If the user selects an invalid path, they will be prompted to choose again

Special Requirements:

-The game should give a hint of where to first move the robot piece. This can be enabled/disabled by the user.

-The user should be able to save a game and return to it later to complete it.

Open Issues:

-How will a token be rewarded if two or more players choose the same path with the same number of moves?