**1. Game Controller Class**

1.1 Class-Responsibility-Collaborator card

|  |  |
| --- | --- |
| Game Controller | |
| Responsibilities | Collaborator |
| * Identify the turn & round * Identify if it’s a human robot or AI * Play function * Move a robot * Shot a robot * Rotate a robot * Update the robot statistics in robot model * Update the game board if needed * Exit to the main menu | * Gameboard Interface class * Robot model class * Words model class * Mailboxes model class |

1.2 Game Controller Class

|  |
| --- |
| Game Controller |
|  |
| + whose\_turn()  +identify\_HumanOrAI(robot\_id)  + play(robot\_id)  + move()  + shot()  + rotate()  - update\_robot\_stat()  - update\_gameboard()  + exit\_to\_mainmenu() |

1.3 Function explanations

1.3.1 whose\_turn()

* identify which robot should be in the battle field for a specific turn.
* pre: the start button is clicked on the interface of ‘selecting robot’
* post: identify which robot should be played in the current turn
* return: the robot\_id of the robot

1.3.2 identify\_HumanOrAI(robot\_id)

* identify the robot that is going to be played should belong to a human team or AI team
* pre: after the turn is identified
* post: identify the robot team
* return: nothing

1.3.3 play(robot\_id) ---- draw the logic diagram already drew yesterday

* if the robot is played by the human, call functions move/attack/rotate directly inside the game controller to finish play; if the robot is played by AI, call interpreter class to translate
* pre: the turn and team of a robot has already been identified
* post: pass the robot to interpreter if it’s AI, or just call functions move/attack/rotate directly inside.
* return: nothing

1.3.4 rotate()

* turn the direction of a robot
* pre: the function play is called. The robot already chooses its destination to move to.
* post: the robot rotates to the desired direction, update the gameboard interface
* return: nothing

1.3.5 move()

* move a robot on the game board
* pre: the function play is called. The robot has at least one move point
* post: the robot moves to the desired destination, update the gameboard interface
* return: nothing

1.3.6 attack()

* attack a robot if in range
* pre: the function play is called. The robot doesn’t use its attack point
* post: the robot finishes it attack action, update the gameboard interface in needed.
* return: nothing

1.3.7 update\_robot\_stat()

* keep track the statistics of robots in each turn. Collaborate with robot model.
* pre: each time a robot rotates, moves or shoots.
* post: change the robot statistics in robot model.
* return: nothing

1.3.8 update\_gameboard()

* update the game board interface in each turn. Collaborate with game board interface.
* pre: turn change, robot dead or
* post: change the robot statistics in robot model. Update the game board interface.
* return: nothing

1.3.9 exit\_to\_mainmenu()

* exit to the main menu on clicking the exit button