**InitializationController**

* 1. Class-responsibility-collaboration card

|  |  |
| --- | --- |
| Reponsibilities | collabrator |
| * Build the robots * Build Gameboard and tiles * Start the game * Terminate(shut down) the game * Select Mode * Select robots(Human/AI) * Download from and to JSON file | * slectMode Interface class * gameboard model class * tiles model class * robot model class |

1.2 Initialization Controller Class

|  |
| --- |
| InitializationController |
| int number\_Robert  string robort[];  Boolean isAI[];  Int size\_gameboard(); |
| +buildRobot(robot[]);  +buildGameBoard(size\_gameboard);  +setMode(isAI[]);  -initialization();  +download(robot\_id);  +upload();  -terminate(); |

1.3Function explanations

1.3.1

buildRobot(robot[]);

* build robots and pass their statics to Robot model.
* Pre: after the player choose the robot number.
* Post: the statics has been built and stored temporarily.
* Return: robots script

1.3.2

buildGameBoard(size\_gameboard());

* take the size of the board and build the board correspondingly.
* Pre: after the player choose the game board size.
* Post: game board has been built.
* Return: nothing

1.3.3

setMode(isAI[]);

* take the input of which robot is Human and which robot is AI, put these data into robot model.
* pre: after the player choose the robot or AI.
* post: corresponding data has been put in robot model.
* Return: Boolean, true if the data was put in robot model successfully, false otherwise

1.3.4

initialization();

* take the scrip from the model and start the game
* pre: the robot scripts have been built in the model and downloaded successfully. The game board has been built successfully
* Post: the game starts
* Return Boolean, true if the game starts successfully, false otherwise

1.3.5

download(robot\_id);

* download the script from model
* pre: script was in model object
* post: the script was downloaded successfully
* return: Boolean, true downloaded if downloaded successfully, false otherwise

1.3.6

upload();

* upload the script from model
* pre: data was successfully subtracted
* post: the script was downloaded successfully
* return: Boolean, true if uploaded successfully, false otherwise

1.3.7

terminate();

* shut down the game
* pre: terminate key was pressed.
* Post: the program terminates
* Return: nothing