**Test cases**

The red color font means that test case does not pass and need to be finished next sprint

Consider the sprint task #1 – Develop GUI portion for drawing the size of the block(environment)

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 1 | Scenario | Input(s) | Expected output |
| 1 | User draws the normal(eg.10\*10) size of block by using mouse | Drawing using mouse | The block environment is visible with certain size |
| 2 | User draws the extreme small size 1\*1 of block by using mouse | Drawing using mouse | The block environment can not be set up |
| 3 | User draws the size of block reach the border by using mouse | Drawing using mouse | The block side which reaches the border will stop |
| 4 | User views block view with full size window | None | The layout of block view shows correctly |
| 5 | User views block view while changing the size of window | None | The layout of block view shows correctly |

Consider the sprint task #2 – Generate corresponding block data while user drawing environment

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 1 | Scenario | Input(s) | Expected output |
| 1 | User draws the normal(eg.10\*10) size of block by using mouse | Dragging using mouse | The data will be generated |
| 2 | User draws the extreme size 1\*1 of block by using mouse | Drawing using mouse | The data will not be generated |
| 3 | User draws the size of block reach the border by using mouse | Drawing using mouse | The data will be generated |

Consider the sprint task #3 – Develop GUI portion for Develop GUI portion for constructing the regions

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint1 | Scenario | Input(s) | Expect Output |
| 1 | User draw the region outside the block | Mouse click and drag | No response |
| 2 | User draw the region within the block | Mouse click and drag | The regions displayed in the block view |
| 3 | User draw the region with part of it out of the block | Mouse click and drag | The region drawn within the block. When the mouse out of the block, it should not get response. When the mouse within the block, it can response correctly. |
| 4 | User draw the region by moving his mouse along the box diagonal | Mouse click and drag | The region clicked at first. There is no response for the dragging. |

Consider the sprint task #4 – Generate corresponding regions data while user drawing environment

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint | Scenario | Input(s) | Expect Output |
| 1 | User draw the region within the block | Mouse click and drag | The matrix with 0 filled in the region positions, 1 filled in the block area. |
| 2 | User draw the region outside the block | Mouse click and drag | No data generated |
| 3 | User draw the region with part of it out of the block | Mouse click and drag | The matrix with 0 filled in the region positions within the boundary, 1 filled in the block area. |
| 4 | User draw the region by moving his mouse along the box diagonal | Mouse click and drag | The matrix with 0 filled in the position clicked at first, 1 filled in the block area. |

Consider the sprint task #5 – Generate ID for agents

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint1 | Scenario | Inputs | Expect Output |
| 1 | User click on the regions area | Mouse click | Ids generated for Agents |
| 2 | User click on the area out of the regions area | Mouse click | None |

Consider the sprint task #6 – Develop GUI to set agents position

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint1 | Scenario | Inputs | Expect Output |
| 1 | User click on one open space in the region area less than 5 times | Mouse click | The agents appeared at the clicked position as colored blocks each one of which represents an agent |
| 2 | User click on one open space in the regions area more than 4 times | Mouse click | The agents appeared at the clicked position as number which shows how many agents at this box. |
| 3 | User click on the area out of the regions area | Mouse click | None |
| 4 | User click once on multiple open space in the region area | Mouse click | Each clicked position appears an agent |

Consider the sprint task #7 – Develop data structure to store agents and their positions

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case # | Scenario | Inputs | Expect Output |
| 1 | Invoke addAgent() method after the user set the agents | Agent’s ID and Initial position of agent | None |

Consider the sprint task #9– Develop GUI to show the graphical view

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 2 | Scenario | Input(s) | Expected output |
| 1 | User select a particular region (eg.region#1) | Mouse click a particular region in sketch(top left corner) | The GUI shows the particular region by corresponding graphical view |
| 2 | User select a particular region (eg.region#1) | Mouse click a particular region in environment canvas(center) | None |
| 3 | User select the region only has one open space | Mouse click a particular region in sketch (top left corner) | The GUI only shows one node without edge |
| 4 | User select the region has lots of open space(eg.100) | Mouse click a particular region in sketch (top left corner) | The GUI shows the whole region and it can be change to fit the screen with reduce or magnify the size. |

Consider the sprint task #10– Develop code to show the number of agents at each node

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 2 | Scenario | Input(s) | Expected output |
| 1 | User select a particular region (eg.region#1) | Mouse click a particular region in thumbnail(top left corner) | The GUI shows the particular region by corresponding graphical view with correct current agent |
| 2 | User runs one step in graphical view | Mouse click the button “run one step”( top right corner) | The GUI shows the correct agents in corresponding node |
| 3 | User runs multiple step in graphical view | Input particular steps(integer) and mouse click the button “run steps” ( top right corner) | The GUI shows the correct agents in corresponding node |
| 4 | User runs multiple step in graphical view | Input a non integer steps and mouse click the button “run steps” ( top right corner) | The GUI shows error message |

Consider the sprint task #11 – Develop GUI to show the details about the agents in each node in graphical view

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 11 | Scenario | Inputs | Expect Output |
| 1 | Single click on a node which has been visited | Mouse click | All agents which have visited this node is shown on a board. |
| 2 | Single click on a node which has not been visited | Mouse click | None |
| 3 | double click on a node which has been visited | Mouse click | All agents which have visited this node is shown on a board. |
| 4 | double click on a node which has not been visited | Mouse click | None |
| 5 | Mouse hovering on a node which has been visited | None | None |
| 6 | Mouse hovering on a node which has not been visited | None | None |
| 7 | Single click or double click on a particular node which has been visited | Mouse click | Show the correct corresponding detail information of that particular node |

Consider the sprint task #14– Develop the GUI portion of block view provide options for user to choose run the algorithm step by step or execute the algorithm for a fixed number of times/step

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 2 | Scenario | Input(s) | Expected output |
| 1 | User clicks the button which for running one step in block view | Mouse click the button “run one step”( top right corner) | The GUI shows the correct current agents in corresponding space and the color of the open space which is visited by agent will change, and there is a trace for every agent will be showed |
| 2 | User clicks the button which running multiple steps in block view, but the user does not input any step in the button. | Mouse click the button “run steps” without inputting( top right corner) | None |
| 3 | User clicks the button which running multiple steps in block view | Input particular steps and mouse click the button “run steps” ( top right corner) | The GUI shows the correct current agents in corresponding space and the color of the open space which is visited by agent will change, and there is a trace for every agent will be showed |
| 4 | User clicks other button which is not for running one step or steps in block view | Mouse click | None |
| 5 | User runs multiple step in block view | Input a non integer steps and mouse click the button “run steps” ( top right corner) | The GUI shows error message |

Consider the sprint task #15– Develop the GUI portion for showing the execution of the algorithm on block view

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 2 | Scenario | Input(s) | Expected output |
| 1 | User runs one step in block view | Mouse click the button “run one step”( top right corner) | The GUI shows the correct current agents in corresponding space and the color of the open space which is visited by agent will change, and there is a trace for every agent will be showed |
| 2 | User runs multiple steps in block view | Input particular steps and mouse click the button “run steps” ( top right corner) | The GUI shows the correct current agents in corresponding space and the color of the open space which is visited by agent will change, and there is a trace for every agent will be showed |
| 3 | User clicks the button which running multiple steps in block view, but the user does not input any step in the button. | Mouse click the button “run steps” without inputting( top right corner) | None |

Consider the sprint task #16– Develop the GUI portion for showing the execution of the algorithm on graphical view

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 2 | Scenario | Input(s) | Expected output |
| 1 | User runs one step in graphical view | Mouse click the button “run one step”( top right corner) | The GUI shows the correct current agents in corresponding space and the color of the open space which is visited by agent will change, and there is a trace for every agent will be showed |
| 2 | User runs multiple steps in graphical view | Input particular steps and mouse click the button “run steps” ( top right corner) | The GUI shows the correct current agents in corresponding space and the color of the open space which is visited by agent will change, and there is a trace for every agent will be showed |
| 3 | User clicks other button which is not for running one step or steps in graphical view | Mouse click | None |
| 4 | User clicks the button which running multiple steps in graphical view, but the user does not input any step in the button. | Mouse click the button “run steps” without inputting( top right corner) | None |
| 5 | User runs multiple step in graphical view | Input a non integer steps and mouse click the button “run steps” ( top right corner) | The GUI shows error message |

Consider the sprint task #20 – Develop DB access code for storing and retrieving run information

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 20 | Scenario | Inputs | Expect Output |
| 1 | Save a run information | JSON data of a run | Server return message that save successful |
| 2 | Retrieve a history run using start date and end date | Start date and end date | Matched history run is retrieved from the table |
| 3 | Retrieve a history run using start date and end date(but start date is behind end date) | Start date and end date | Show alert message that the start date should before the end date |
| 3 | Retrieve a history run just using start date | Start date | Server return error message that must include end date |
| 4 | Retrieve a history run just using start date | End date | Server return error message that must include start date |
| 5 | Retrieve a history run using description | Description | Matched history run is retrieved from the table |
| 6 | Retrieve a history run using correct start date and end date with corresponding description | Start date, end date and description | Matched history run is retrieved from the table |
| 7 | Retrieve a history run using correct start date and end date with non-corresponding description | Start date, end date and description | None |

Consider the sprint task #21 – Develop GUI to show all run information

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 21 | Scenario | Inputs | Expect Output |
| 1 | Display all history runs | None | All history runs are displayed on the screen |
| 2 | Display all history runs; there is no history run in DB | None | Nothing will be displayed on the screen |

Consider the sprint task #22 – Develop GUI to filter history runs

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 22 | Scenario | Inputs | Expect Output |
| 1 | Select the start date and end date | Mouse Click | The time selector is shown |
| 2 | Filter the history runs after selecting start and end date | Mouse click on filter button (Selected start date is Feb 27 2017, end date is Feb 27 2017) | All history runs are displayed on the screen |
| 3 | Filter the history runs before selecting start and end date | Mouse click on filter button | Message showing that user must select start and end date before clicking the filter button |
| 4 | Filter the history runs after selecting start and end date; the start date is after the end date | Mouse click on filter button (Selected start date is Mar 5 2017, end date is Feb 27 2017) | Message showing that the start date must be before the end date |
| 5 | Filter the history runs by only selecting the start date | Mouse click on filter button (Selected start date is Feb 27 2017) | Message showing that “please enter both start date and end date” |
| 6 | Filter the history runs by only selecting the end date | Mouse click on filter button (Selected end date is Feb 29 2017) | Message showing that “please enter both start date and end date” |
| 7 | Filter the history runs only by inputting description | Description | All matched history runs are displayed on the screen |
| 8 | Filter the history runs by selecting correct the start date and end date with corresponding description | Mouse click on filter button (Selected start date is Feb 27 2017, end date is Feb 28 2017) and input description | All matched history runs are displayed on the screen |
| 9 | Filter the history runs by selecting correct the start date and end date with non-corresponding description | Mouse click on filter button (Selected start date is Feb 27 2017, end date is Mar 4 2017) and input description | None |

Consider the sprint task #23 – validate the input data before executing the algorithm

Some of the test cases for this task are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 23 | Scenario | Inputs | Expect Output |
| 1 | Choose the free form algorithm and user click the run button | Environment configuration file whose content is in correct format with the number of agents is no more than ceil(n/2) in each region | the algorithm executes correctly |
| 2 | Choose the free form algorithm and user click the run button | Environment configuration file whose content is in correct format with the number of agents is more than ceil(n/2) in each region | Prompt error message |
| 3 | Choose the free form algorithm and user click the run button | Environment configuration file whose content is in incorrect format | Prompt data format is wrong |
| 4 | Choose the constrained3 algorithm and user click the run button | Environment configuration file whose content is in correct format with the number of agents is no more than ceil(n/3) in each region | the algorithm executes correctly |
| 5 | Choose the constrained3 algorithm and user click the run button | Environment configuration file whose content is in correct format with the number of agents is more than ceil(n/3) in each region | Prompt error message |
| 6 | Choose the constrained3 algorithm and user click the run button | Environment configuration file whose content is in incorrect format | Prompt data format is wrong |
| 7 | Choose the constrained4 algorithm and user click the run button | Environment configuration file whose content is in correct format with the number of agents is no more than ceil(n/4) in each region | the algorithm executes correctly |
| 8 | Choose the constrained4 algorithm and user click the run button | Environment configuration file whose content is in correct format with the number of agents is more than ceil(n/4) in each region | Prompt error message |
| 9 | Choose the constrained4 algorithm and user click the run button | Environment configuration file whose content is in incorrect format | Prompt data format is wrong |

Consider the sprint task #24 – Implement the constrained-3 algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 24 | Scenario | Inputs | Expect Output |
| 1 | Set the number of agents in a region less than ⌈n/3⌉ where n is the number of open spaces in that region | Mouse click on the open spaces | the algorithm executes correctly |
| 2 | Set the number of agents in a region equal ⌈n/3⌉ where n is the number of open spaces in that region | Mouse click on the open spaces | the algorithm executes correctly |
| 3 | Set the number of agents in a region exceed ⌈n/3⌉ where n is the number of open spaces in that region | Mouse click on the open spaces | Prompt error message |
| 4 | Run this algorithm for one step ,when an agent choose its target from the target list | Click the button which will run one step | The agent chooses the node that has the longest path from the current position of the agent， the selected target node is NOT deleted from the target list， the node the agents visited is removed from the target list |
| 5 | Run this algorithm for multiple steps that doesn’t exceed the maximum steps, when an agent choose its target from the target list | Click the button which will run multiple steps | The agent choose the node that has the longest path from the current position of the agent, the selected target node is NOT deleted from the target list， the nodes the agents visited are removed from the target list |
| 6 | Run this algorithm for multiple steps that exceeds the maximum steps, when an agent choose its target from the target list | Click the button which will run multiple steps | Prompt the steps exceed the maximum steps |

Consider the sprint task #25 – Implement the constrained-4 algorithm

|  |  |  |  |
| --- | --- | --- | --- |
| Test case #sprint 25 | Scenario | Inputs | Expect Output |
| 1 | Set the number of agents in a region less than ⌈n/4⌉ where n is the number of open spaces in that region | Mouse click on the open spaces | the algorithm executes correctly |
| 2 | Set the number of agents in a region equal ⌈n/4⌉ where n is the number of open spaces in that region | Mouse click on the open spaces | the algorithm executes correctly |
| 3 | Set the number of agents in a region exceed ⌈n/4⌉ where n is the number of open spaces in that region | Mouse click on the open spaces | Prompt error message |
| 4 | Run this algorithm for one step ,when an agent choose its target from the target list | Click the button which will run one step | The agent chooses the node that has the longest path from the current position of the agent， the selected target node is NOT deleted from the target list， the node the agents visited is removed from the target list |
| 5 | Run this algorithm for multiple steps that doesn’t exceed the maximum steps, when an agent choose its target from the target list | Click the button which will run multiple steps | The agent choose the node that has the longest path from the current position of the agent, the selected target node is NOT deleted from the target list， the nodes the agents visited are removed from the target list |
| 6 | Run this algorithm for multiple steps that exceeds the maximum steps, when an agent choose its target from the target list | Click the button which will run multiple steps | Prompt the steps exceed the maximum steps |