# Mobile Automata

**Overview -** A class of automata similar to cellular automata but which have a single "active" cell instead of updating all cells in parallel. In a mobile automaton, the evolution rules apply only to the active cell, and also specify how the active cell moves from one generation to the next. (Extracted from Wolfram)

**Implementation –**

Created an Object Model for demonstrating valid Mobile Automata.

**Classes Involved**

1. **MAutomataDriver** – Contains the main method to start the simulation. This class extends from MAApp and has all the UI Components MAApp provides i.e., Menu Manager, Jframe etc.

It initializes the Jframe and adds User Interaction Panels to the Frame.

1. **MARegionSet** – Contains a Map to hold MARegion objects with generation Count as key. Used TreeMap as fetching is easy and keys are ordered. I preferred Map over ArrayList as I allowed a Rewind feature to get the Regions based on keys and making the simulation go back in time (Generation). Also this is the Main Simulation panel where Graphics of the simulation are displayed. The run() method contains all the trigger functions to start/pause/rewind/stop/check the simulation status and to call Paint Component to update the drawing using repaint().
2. **MARegion –** Contains RuleNames, 2D Array of MACell, Dimensions of the region. For handling behavior we have helper methods like createNextRegion() which is responsible for creating next cell states based on the current region cell states. Uses nextCellStates() and nextActivePos() methods extensively to create a new Region and return it back to MARegionSet which adds it to the Map.
3. **MACell -** Contains MACellState, MARegion, CellXPos, CellYPos ( to keep track of which region and position it belongs to). Also contains helper methods to determine Neighbors Count which helps MARule class. It implements IMARule which insists MACell to implement getNextCellState() and getNextCellPos() methods whose implementation is provided by derived class MARule. So MACell is an abstract class and you can create only an MARule object.
4. **MARule –** Extends from MACell. The methods getNextCellState() and getNextCellPos() must be implemented and these methods takes help of conditions to navigate the control to the required RuleNames and the rule logic helps in deciding the cell State and next active cell position.

**ENUMS Involved –**

To standardize the Cell States and RuleNames, for easy maintainability I have created 2 neumes namely

1. **RuleNames** – Which holds all the available Rule Names (***LOCKME***, ***DEADALIVE***, ***BRIANSBRAIN***, ***TOPDOWNTREE***, ***GOLDWINNER***) (You can find this enum under MARule Class)
2. **MACellState** – Which holds all the available Cell States (***ALIVE***, ***DEAD***, ***DYING***). You can find this enum under MACell Class)

**Interface Involved –**

1. **IMARule –** This is to force the MACell class or its derived classes to implement getNextCellState() method to determine the future MACellState.

**Note : Classes MAApp and MenuManager are used primarily for UI and most of the implementation is provided by defualt. MAAutomataDriver extends MAApp to get the UI dispatcher thread and instantiate Jframe and associated Jpanels.**

**Program Flow – ( In Code )**

MenuManager

Init(), getMenuBar()

MAApp

initGUI()

MAutomata Driver

main()

Jframe Created showUI()

MARule

getNextCellState()/ getNextCellPos()

User Interaction Starts

initGUI()

User Clicks on Create after choosing the config (Rules etc.)

*MACell*

getNextCellState() / getNextCellPos()

MARegionSet

addRegionToMap()

New Cell State Returned

User Clicks on Start

MARegion

createNextRegion() 🡪 nextCellStates() / nextActivePos()

run()

**Note :** For LOCKME next active cell pos is returned. This is added to track the active cell position which is the need for Mobile Automata.

**User Interaction –**

**Start**

Simulation is started

**Rewind**

Simulation is rewinded from the point where it was paused.

Start the Application

Create the MARegionSet by choosing the available config and click on **Create**

**Pause**

Simulation is paused

**Stop**

Simulation panel is disabled and user can create a new MARegionSet

User can restart from the point of pause

After Starting user can

Pause the simulation

**Note :**

* When Rewind reaches initial State, the Simulation will be paused.
* When Simulation is completed, either by completion criteria or Generation Limit reached, User should click on **Stop** if he wishes to create a new regionSet. (The config will show the previous values selected by the user)

**Use Cases :**

**UseCase 1. DeadAlive**

**ID :** 1

**Level:** Low

**Description :** In this rule, 2 cells will act as active (Alive) agents and help in making the entire colony (Region) Alive.

**Actors :** Alive (Green) cells in the middle.

**Stakeholders & Interests :** All the cells in the region (Initially starting with 2 cells)

**PreConditions:**

1. Rule to be selected by the user is DeadAlive
2. Create the region using Create Button of the panel.

**Triggers :**

1. User clicks on START button to perform the simulation.
2. User clicks on PAUSE button tp pause the simulation.
3. User clicks on PAUSE and the clicks on REWIND button to go back to previous generations, from where he paused.
4. User clicks on STOP button to remove the created regionSet and if wishes can create a new regionSet.

**Post-Conditions :**

1. Success : User is shown “Simulation Completed Successfully” message in status Panel and the console. This may happen when all the cells in the region are ALIVE. (Green)
2. Generation Limit Reached : When Generation Count reaches Generation Limit, a status “Simulation reached maximum generation Limit…” is shown both in the console and status panel.
3. Initial State : When user clicks on PAUSE 🡪 REWIND and goes back to initial state, Simulation is paused and a status “Simulation Paused as user went back to initial state” is shown. Now User can start or stop the simulation.

**Main Scenario :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, to create a new config.

**Alternate Scenario :**

**SCN 1 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation, reaches initial state and simulation is paused again.

**SCN 2 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**SCN 3 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**STEP 7 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 4:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 5:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 6:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 7:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**UseCase 2. BriansBrain**

**ID :** 2

**Level:** High

**Description :** In this rule, 2 cells will act as active (Alive) agents and help in forming chaotic patterns and finally forming spaceships which fly away from the region. Simulation ends when all cells are dead. (i.e., white color )

**Actors :** Alive (Green) cells in the middle. Supporting actors are Dying cells (Blue in color)

**Stakeholders & Interests :** All the cells in the region (Initially starting with 2 cells)

**PreConditions:**

1. Rule to be selected by the user is BriansBrain
2. Create the region using Create Button of the panel.

**Triggers :**

1. User clicks on START button to perform the simulation.
2. User clicks on PAUSE button tp pause the simulation.
3. User clicks on PAUSE and the clicks on REWIND button to go back to previous generations, from where he paused.
4. User clicks on STOP button to remove the created regionSet and if wishes can create a new regionSet.

**Post-Conditions :**

1. Success : User is shown “Simulation Completed Successfully” message in status Panel and the console. This may happen when all the cells in the region are DEAD. (white)
2. Generation Limit Reached : When Generation Count reaches Generation Limit, a status “Simulation reached maximum generation Limit…” is shown both in the console and status panel.
3. Initial State : When user clicks on PAUSE 🡪 REWIND and goes back to initial state, Simulation is paused and a status “Simulation Paused as user went back to initial state” is shown. Now User can start or stop the simulation.

**Main Scenario :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, to create a new config.

**Alternate Scenario :**

**SCN 1 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation, reaches initial state and simulation is paused again.

**SCN 2 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**SCN 3 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**STEP 7 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 4:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 5:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 6:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 7:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**UseCase 3. TOPDOWNTREE**

**ID :** 3

**Level:** High

**Description :** In this rule, 1 cell will act as active (Alive) agent and help in forming a tree as it gets split when simulation is processed. When at least 1 active cell reaches the bottom cell of the region, the simulation is completed, and the color of active cell is changed from Green to Black, and for DEAD cell it is changed from White to gold as a symbol that the tree is grown up and even dead can be happy and prosper like gold.

**Actors :** Alive (Green) cell in the top. Supporting actors are Dying cells (Blue in color)

**Stakeholders & Interests :** All the cells in the region (Initially starting with 1 cell)

**PreConditions:**

1. Rule to be selected by the user is TOPDOWNTREE
2. Create the region using Create Button of the panel.

**Triggers :**

1. User clicks on START button to perform the simulation.
2. User clicks on PAUSE button tp pause the simulation.
3. User clicks on PAUSE and the clicks on REWIND button to go back to previous generations, from where he paused.
4. User clicks on STOP button to remove the created regionSet and if wishes can create a new regionSet.

**Post-Conditions :**

1. Success : User is shown “Simulation Completed Successfully” message in status Panel and the console. This may happen when at least 1 active cell reaches the bottom cell of the region.
2. Generation Limit Reached : When Generation Count reaches Generation Limit, a status “Simulation reached maximum generation Limit…” is shown both in the console and status panel.
3. Initial State : When user clicks on PAUSE 🡪 REWIND and goes back to initial state, Simulation is paused and a status “Simulation Paused as user went back to initial state” is shown. Now User can start or stop the simulation.

**Main Scenario :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, to create a new config.

**Alternate Scenario :**

**SCN 1 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation, reaches initial state and simulation is paused again.

**SCN 2 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**SCN 3 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**STEP 7 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 4:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 5:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 6:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 7:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**UseCase 4. GOLDWINNER**

**ID :** 4

**Level:** High

**Description :** In this rule, 1 cell will act as active (Alive) agent (blue) and help in opening a black bag (black cells) which contain gold. When the blue cell reaches top it starts to open the bag as it is unzipped (i.e., the agent splits and goes in 2 directions). Gold cells appear when the bag is opened. Simulation is completed when there are only 2 strips (columns) of black cells remained (one at each end of the region).

**Actors :** Alive (BLUE) cell in the bottom.

**Stakeholders & Interests :** All the cells in the region (Initially starting with 1 cell)

**PreConditions:**

1. Rule to be selected by the user is GOLDWINNER
2. Create the region using Create Button of the panel.

**Triggers :**

1. User clicks on START button to perform the simulation.
2. User clicks on PAUSE button tp pause the simulation.
3. User clicks on PAUSE and the clicks on REWIND button to go back to previous generations, from where he paused.
4. User clicks on STOP button to remove the created regionSet and if wishes can create a new regionSet.

**Post-Conditions :**

1. Success : User is shown “Simulation Completed Successfully” message in status Panel and the console. This may happen when there are only 2 strips (columns) of black cells remained (one at each end of the region).
2. Generation Limit Reached : When Generation Count reaches Generation Limit, a status “Simulation reached maximum generation Limit…” is shown both in the console and status panel.
3. Initial State : When user clicks on PAUSE 🡪 REWIND and goes back to initial state, Simulation is paused and a status “Simulation Paused as user went back to initial state” is shown. Now User can start or stop the simulation.

**Main Scenario :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, to create a new config.

**Alternate Scenario :**

**SCN 1 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation, reaches initial state and simulation is paused again.

**SCN 2 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**SCN 3 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**STEP 7 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 4:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 5:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 6:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 7:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**UseCase 5. LOCKME**

**ID :** 5

**Level:** VeryHigh

**Description :** In this rule, 1 cell will act as active (Alive) agent (BLACK) . Initially it starts at 0th cell. The cell checks for its neighbors (i.e., if cell’s position is (x,y) , it checks for (x+1,y+1 to x-1,y-1) a total of 8 neighbors in that order.) The alive agent moves to a dead(white) cell in the neighbor. The first neighbor checked is dead the alive agent jumps to that position. The mobile automata ends when the active cell cannot find any dead neighbors.

**Actors :** Active cell in the 0th cell position. (black in color)

**Stakeholders & Interests :** Active cell and it’s 8 neighbors at any given point of time. This rule is mainy developed to satisfy the basic mobile automata where only an active cell changes its state and remaining cells do not change their state over time.

**PreConditions:**

1. Rule to be selected by the user is LOCKME.
2. Create the region using Create Button of the panel.

**Triggers :**

1. User clicks on START button to perform the simulation.
2. User clicks on PAUSE button tp pause the simulation.
3. User clicks on PAUSE and the clicks on REWIND button to go back to previous generations, from where he paused.
4. User clicks on STOP button to remove the created regionSet and if wishes can create a new regionSet.

**Post-Conditions :**

1. Success : User is shown "OOPS!! You are locked... Simulation completed Successfully..." message in status Panel and the console. This may happen when the active cell cannot find dead neighbors.
2. Generation Limit Reached : When Generation Count reaches Generation Limit, a status “Simulation reached maximum generation Limit…” is shown both in the console and status panel.
3. Initial State : When user clicks on PAUSE 🡪 REWIND and goes back to initial state, Simulation is paused and a status “Simulation Paused as user went back to initial state” is shown. Now User can start or stop the simulation.

**Main Scenario :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, to create a new config.

**Alternate Scenario :**

**SCN 1 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation, reaches initial state and simulation is paused again.

**SCN 2 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**SCN 3 :**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is Paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5 :** User clicks on PAUSE, Simulation is paused.

**STEP 6 :** User clicks on START, Simulation is started from the point it was paused.

**STEP 7 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 4:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 5:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 6:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2 :** User clicks on START, Simulation is started, and simulation is completed successfully.

**STEP 3 :** User clicks on PAUSE, Simulation is paused.

**STEP 4 :** User clicks on REWIND, Simulation goes back from the generation it was paused to previous generation.

**STEP 5:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**SCN 7:**

**STEP 1 :** User selects the required configuration and clicks on Create.

**STEP 2:** User clicks on STOP, RegionSet is disabled and user can create a new config if he wishes.

**Sample Outputs :**

|  |
| --- |
| **DEADALIVE** |

|  |
| --- |
| **BRIANSBRAIN** |

|  |
| --- |
| **TOPDOWNTREE** |

|  |
| --- |
| **GOLDWINNER** |

|  |
| --- |
| **LOCKME** |

THANK YOU !!