3D Maze Generator and Solver

# Description

The software will generate a three dimensional maze using a modified version of Prim’s algorithm. It will then display the maze to the user in layers that the user may scroll through. cells in the maze that have an upwards passage will have an up arrow in the and the same from downwards passages but with a down arrow. Cells witch have an upwards and downwards passage will be marked with a bidirectional arrow. After the maze has been generated and displayed, it will be able to be solved with the Recursive Backtracker algorithm or the A\* Pathfinding algorithm. The solved maze will then be displayed to the user along with statistics about how efficient the algorithm was.

# The Interface

When the program is launched, there will be nothing on the left side of the window. On the right side of the window are two input sections labeled “Generation” and “Solving” and an information panel labeled “Results”. In the “Generation” section the user will be able to specify a width, height and depth for the maze to be generated. They may also select whether they wish to view an animation of the generation process and set the speed of the animation. When the user hits generate the resulting maze will be displayed on the left. The legend for cell colours is as follows:

|  |  |
| --- | --- |
| **Colour** | **Meaning** |
| White | In Maze |
| Gray | Not In Maze |
| Black | Frontier |
| Yellow | Selected |
| Dark Gray | Visited |
| Red | In Solution Path |
| Green | Start of Maze |
| Magenta | End of Maze |

The user may scroll through the layers of the maze using the scroll wheel and alter the start and end positions by clicking on a cell. Left clicking sets start point and right clicking sets end point. After a maze is generated the user may select a solving algorithm from the dropdown menu in the “Solving” section of the interface. They may also specify whether they wish to view an animation of the solving process and set the speed of the animation. When the user hits the solve button the selected algorithm will solve the maze and display its found path to the user. It will also display statistics about it efficiency in the “Results” section of the interface.

# Classes

|  |  |  |  |
| --- | --- | --- | --- |
| GUI | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| mazeGenerator | MazeGenerator | The object used for maze generation | |
| mazeSolver | MazeSolver | The object used for maze solving | |
| maze | Maze | The current maze being displayed to the user | |
| layer | int | The current layer of the maze being displayed | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| GUI | GUI | None | The constructor. Initializes the generator and solver. |
| generateActionPreformed | void | ActionEvent evt | Is called when the user presses the “Generate!” button. Uses the mazeGenerator object to create a new maze. |
| solveActionPreformed | void | ActionEvent evt | Is called when the user presses the “Solve!” button. Uses the mazeSolver object to solve the current maze. |
| animateGenStateChanged | void | ChangeEvent evt | Updates the maze generator on whether or not to animate the generation depending on the state of the checkbox. |
| animateSolveStateChanged | void | ChangeEvent evt | Updates the maze solver on whether or not to animate the solving depending on the state of the checkbox. |
| speedGenStateChanged | void | ChangeEvent evt | Tells the maze generator how fast to animate the generation. |
| speedSolveStateChanged | void | ChangeEvent evt | Tells the maze solver how fast to animate the solving. |
| canvasMouseClicked | void | MouseEvent evt | Updates the maze’s start and end position to the cell the user clicked on based on which mouse button was used. |
| CanvasMouseWheelMoved | void | MouseWheelEvent evt | Updates the current layer of the maze being displayed based on which direction the mouse wheel was moved. |

|  |  |  |  |
| --- | --- | --- | --- |
| Maze | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| grid | MazeCell[][][] | A three dimensional array to store the cells of the maze. | |
| start | MazeCell | The cell set as the start of the maze. | |
| end | MazeCell | The cell set as the end of the maze. | |
| complete | boolean | Boolean that is true when the maze is done being generated. | |
| canvas | JPanel | The component to render the maze onto. | |
| layer | int | The layer of the maze to generate, in other words, the z-coordinate. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| Maze | Maze | int xSize  int ySize  int zSize  JPanel canvas | The constructor. Initializes the grid and fills it with new maze cells and sets the default start and end points. |
| render | void | None | Renders the maze onto the canvas using a buffered image. |
| reset | void | None | Resets all the maze cells back to their original state. Used to re-solve a maze. |
| getOutNeighbors | ArrayList<MazeCell> | MazeCell cell | Returns a list of all direct neighbors of the given cell that are not yet in the maze. Used in maze gen. |
| getInNeighbors | ArrayList<MazeCell> | MazeCell cell | Returns a list of all direct neighbors of the given cell that are in the maze already. Used in maze gen. |
| getUnvisitedNeighbors | ArrayList<MazeCell> | MazeCell cell | Returns a list of all traversable neighbors of a given cell that have not yet been visited. Used in maze solving. |
| getDistanceFromEnd | double | MazeCell cell | Returns the distance between the given cell and the end using Pythagorean theorem. Used in maze solving. |
| getDistanceBetween | double | MazeCell cell1  MazeCell cell2 | Returns the distance between the first given cell and the second using Pythagorean theorem. Used in maze solving. |
| getCellAt | MazeCell | int x  int y | Gets the maze cell at the given screen coordinates. |
| getStart | MazeCell | None | Getter for start. |
| setStart | void | MazeCell start | Setter for start. |
| getEnd | MazeCell | void | Getter for end. |
| setEnd | void | MazeCell end | Setter for end. |
| setlayer | void | int layer | Setter for layer. |

|  |  |  |  |
| --- | --- | --- | --- |
| MazeCell | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| x, y, z | int | The coordinates of the cell on the maze grid. | |
| state | int | The state of the cell. Uses the final variable at end of class. | |
| directions | boolean[] | An array of which directions are traversable. Index is true is traversable. Order of indexes is as follows: North, East, South, West, Up, Down. | |
| IN | int | In maze state for cell. | |
| OUT | int | Out of maze state for cell. | |
| FRONTIER | int | Frontier state for cell. | |
| SELECTED | int | Selected state for cell. | |
| VISITED | int | Visited state for cell. | |
| IN\_PATH | int | In solution path state for cell. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| MazeCell | MazeCell | int x, y, z | The constructor. Initializes the coordinate variables. |
| addDirection | void | Direction dir | Adds a new traversable direction to the cell. |

|  |  |  |  |
| --- | --- | --- | --- |
| MazeGenerator (implements Runnable) | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| r | Random | The random number generator for the class. | |
| thread | Thread | The thread the generator is currently running in. | |
| running | boolean | Whether or not the generator is currently running. | |
| maze | Maze | The current maze being generated. | |
| xSize, ySize, zSize | int | The size of the maze to be generated. | |
| frontier | ArrayList<MazeCell> | Stores the current frontier cells. | |
| canvas | JPanel | The canvas for the maze to render onto. | |
| animate | boolean | Whether or not to animate the generation process. | |
| speed | int | The speed of the animation | |
| progressBar | JProgressBar | The progress bar to update with how far along the generation is. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| MazeGenerator | MazeGenerator | int xSize, ySize, zSize  JPanel canvas  boolean animate  int speed  JProgressBar progress | The constructor. Initializes variables. |
| generate | Maze | None | Starts the generation of a new maze in a new thread using the stored parameters and returns it to the user. |
| run (Overriding) | void | None | Generates the maze. Is called when the thread is started in generate(). |
| addToFrontier | void | ArrayList<MazeCell> cells | Adds all the given cells to the frontier array and sets their states. |
| sleep | void | int ms | Pauses generation for given number of milliseconds. |
| setSpeed | void | int speed | Setter for speed. |
| setAnimate | void | boolean animate | Setter for animate. |
| setxSize | void | int xSize | Setter for xSize. |
| setySize | void | int ySize | Setter for ySize. |
| setzSize | void | int zSize | Setter for zSize. |
| isRunning | boolean | None | Getter for running. |

|  |  |  |  |
| --- | --- | --- | --- |
| MazeSolver (implements Runnable) | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| thread | Thread | The thread the solver is currently running in. | |
| running | boolean | Whether or not the solver is currently running. | |
| algorithms | ArrayList<SolvingAlgorithm> | A list of available solving algorithms. Used to populate the dropdown box. | |
| curAlgorithm | SolvingAlgorithm | The currently running solving algorithm. | |
| curMaze | Maze | The maze currently being solved. | |
| animate | boolean | Whether or not to animate the solving process | |
| speed | int | Speed of the animation. | |
| progressBar | JProgressBar | The progress bar to update with how far the solver is from the end of the maze. | |
| results | JTextPane | The text area to put the results of the solving algorithm. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| MazeSolver | MazeSolver | boolean animate  int speed  JProgressBar progressBar  JTextArea results | The constructor. Initializes variables. |
| solve | void | Maze maze  SolvingAlgorithm | Starts the thread to solve the given maze with the given algorithm. |
| run (Overridding) | void | None | Solves the maze using the given algorithms solve method. Then displays results. |
| isRunning | boolean | None | Getter for running. |
| setRunning | void | boolean | Setter for running. |
| setSpeed | void | int | Setter for speed. |
| setAnimate | void | boolean | Setter for animate. |
| getAlgorithms | ArrayList<SolvingAlgorithm> | None | Getter for algorithms. |

|  |  |  |  |
| --- | --- | --- | --- |
| SolvingAlgorithm (Abstract) | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| running | boolean | Whether or not the algorithm is running. | |
| animate | boolean | Whether or not to animate the algorithm’s process. | |
| speed | int | Speed of the animation | |
| progress | JProgressBar | Progress bar to update with algorithms progress. | |
| numCellsVisited | int | The number of cells the algorithm has visited. | |
| numSteps | int | The number of steps the algorithm taken. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| SolvingAlgorithm | SolvingAlgorithm | boolean animate  int speed  JProgressBar progress | The constructor. Initializes variables. |
| solve | MazeSolution | Maze maze | Abstract method to be implemented in sub-classes. |
| sleep | void | int ms | Pauses the algorithm for given number of milliseconds. |
| setRunning | void | boolean running | Setter for running. |
| setAnimate | void | boolean animate | Setter for animate. |
| setSpeed | void | in speed | Setter for speed. |
| toString (Overridding) | String | None | Abstract method to get the name of the algorithm. |

|  |  |  |  |
| --- | --- | --- | --- |
| RecursiveBacktracker (extends SolvingAlgorithm) | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| r | Random | The random number generator for the class. | |
| path | ArrayList<MazeCell> | The maze solution path. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| RecursiveBacktracker | RecursiveBacktracker | boolean animate  int speed  JProgressBar progress | The constructor. Initializes variables. |
| solve (Overridding) | MazeSolution | Maze maze | Solves the given maze using the recursive backtracker algorithm. |

|  |  |  |  |
| --- | --- | --- | --- |
| AStar (extends SolvingAlgorithm) | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| openSet | PriorityQueue<MazeCell> | Stores the cells currently waiting to be looked at in order of their distance to the end. | |
| score | HashMap<MazeCell, Double> | Stores cells best current path distance from the start. | |
| cameFrom | HashMap<MazeCell, MazeCell> | Stores which cell another cell was “discoved” by. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| AStar | AStar | boolean animate  int speed  JProgressBar progress | The constructor. Initializes variables. |
| solve (Overridding) | MazeSolution | Maze maze | Solves the given maze using the A\* pathfinder algorithm. |
| createPath | ArrayList<MazeCell> | MazeCell cell | Creates the path from given cell to the start using the cameFrom map. |

|  |  |  |  |
| --- | --- | --- | --- |
| AStarComparator (implements Comparator<MazeCell>) | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| maze | Maze | The maze the comparator uses to determine distance to end. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| AStarComparator | AStarComparator | Maze maze | The constructor. Initializes variables. |
| compare (Overridding) | int | MazeCell o1, o2 | Compares the two cells and returns -1 if the first cell is closer to the end, 0 if they are an equal distance and 1 if the second is closer. |

|  |  |  |  |
| --- | --- | --- | --- |
| MazeSolution | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| path | ArrayList<MazeCell> | The solutions path through the maze. | |
| numCellsVisited | int | The number of cells visited to find the solution. | |
| numSteps | int | The number of steps taken to find the solution. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| MazeSolution | MazeSolution | ArrayList<MazeCell> path  int numCellsVisited  int numSteps | The constructor. Initializes variables. |
| toString (Overriding) | String | None | Formats the solution into a string to be displayed to the user. |

|  |  |  |  |
| --- | --- | --- | --- |
| Direction | | | |
| **Variables** | | | |
| **Variable Name** | **Type** | **Description** | |
| dir | int | The number representing the direction. 0-5 represents North, East, South, West, Up and Down respectively | |
| NORTH | int | Constant for North. | |
| EAST | int | Constant for East. | |
| SOUTH | int | Constant for South. | |
| WEST | int | Constant for West. | |
| UP | int | Constant for Up. | |
| DOWN | int | Constant for Down. | |
| **Methods** | | | |
| **Method Name** | **Return Type** | **Parameters** | **Description** |
| Direction | Direction | int dir | The constructor. Initializes variables. |
| getDirectionBetween | Direction | MazeCell cell1, cell2 | Returns the direction from cell1 to cell2. |
| getOpposite | Direction | None | Returns the opposite direction of the current one. |