Maze Class Variables

| **Identifier** | **Location (Method)** | **Data Type** | **Typical Data (Examples)** | **Data Restrictions** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| self.height | Constructor (\_\_init\_\_) | Integer | 15, 20, 30 | Positive integers | Height of the maze |
| self.width | Constructor (\_\_init\_\_) | Integer | 15, 20, 30 | Positive integers | Width of the maze |
| self.maze | Constructor (\_\_init\_\_) | List of lists | [[0, 0, 1], [1, 0, 1]] | 2D array of integers (0 or 1) | Represents the maze grid |
| i | render | Integer | 0, 1, 2 | Iteration index | Row index in the maze rendering loop |
| j | render | Integer | 0, 1, 2 | Iteration index | Column index in the maze rendering loop |
| color | render | String | “black”, “white” | Predefined settings | Colour for rendering maze cells |

UserAuthentication Class Variables

| **Identifier** | **Location (Method)** | **Data Type** | **Typical Data (Examples)** | **Data Restrictions** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| self.root | Constructor (\_\_init\_\_) | Tkinter widget | Tkinter root window object | Tkinter widget instance | Root widget of the tkinter application |
| self.on\_logic\_success | Constructor (\_\_init\_\_) | Function | Callback function | Function reference | Callback function executed on login success |
| conn | create\_user\_database, create\_database, verious | SQLite3 connection | SQLite3 connection object | SQLite3 connection instance | Connection to SQLite database |
| c | create\_user\_database, create\_database, various | SQLite3 cursor | SQLite3 cursor object | SQLite3 cursor instance | Cursor for database operations |
| hashed\_password | check\_password | Bytes | Encoded string | Hashed string | Hashed password for verification |
| user\_password | check\_password | string | plain text | plain text string | user-provided password for verification |
| widget | clear\_root | tkinter widget | individual tkinter widgets | tkinter widget instances | widgets in the root tkinter window |
| username | login | string | “user”, “admin123” | plain text strings | Username for login |
| password | login | string | “pass”, “admin123” | plain text strings | Password for login |
| result | login | Tuple or None | (hashed\_password,), None | Tuple with hashed password or None | Result of database query for user login |
| self.username\_entry | show\_login\_form | Tkinter entry | Tkinter entry widget | Tkinter entry widget instance | Entry widget for username |
| self.password\_entry | show\_login\_form | tkinter entry | tkinter entry widget | tkinter entry widget instance | entry widget for password |
| self.show\_password\_button | show\_login\_form | Tkinter button | tkinter button widget | tkinter button widget instance | button to toggle password visibility |
| self.new\_username\_entry | show\_signup\_form | tkinter entry | tkinter entry widget | tkinter entry widget instance | To capture the new username input during signup |
| self.new\_password\_entry | show\_signup\_form | tkinter entry | tkinter entry widget | tkinter entry widget instance | entry widget for new user password in signup form |
| self.show\_signup\_password\_button | show\_signup\_form | Tkinter button | Tkinter button widget | Tkinter button widget instance | Button to toggle password visibility in signup form |
| self.signup\_window | show\_signup\_form | Tkinter toplevel | tkinter toplevel widget | tkinter toplevel widget instance | toplevel window for the signup form |
| self.admin\_window | show\_admin\_window | tkinter toplevel | tkinter toplevel widget | tkinter toplevel widget instance | toplevel window for the admin panel |
| scrollable\_frame | show\_admin\_window | tkinter frame | tkinter frame widget | tkinter frame widget instance | frame to constain scrollable content in admin window |
| canvas | show\_admin\_window | tkinter canvas | tkinter canvas widget | tkinter canvas widget instance | canvas for scrollable area in admin window |
| scrollbar | show\_admin\_window | tkinter scrollbar | tkinter scrollbar widget | tkinter scrollbar widget instance | scrollbar for the scrollable canvas |
| inner\_frame | show\_admin\_window | tkinter frame | tkinter frame widget | tkinter frame widget instance | Inner frame placed inside canvas for content |
| users | show\_admin\_window | list of tuples | [(‘user1’, ), (‘user2’, )] | List of tuples with usernames | User data retrieved from the database |
| user | show\_admin\_window | Tuple | (‘username’,) | Tuple with a single string | Individual user tuple from the users list |
| username | show\_admin\_window | String | “user1”, “user2” | Plain text strings | Username extracted from user tuple |
| user\_frame | show\_admin\_window | tkinter frame | tkinter frame widget | tkinter frame widget instance | frame for displaying each user in admin window |
| user\_label | show\_admin\_window | Tkinter label | Tkinter label widget | Tkinter label widget instance | Label displaying the username |
| delete\_button | show\_admin\_window | Tkinter button | Tkinter button widget | Tkinter button widget instance | Button to delete the corresponding user |
| count | delete\_all\_users | Integer | 0, 10, 20 | Non-negative integers | Number of users in the database (Excluding the admin) |
| response | delete\_all\_users | Boolean | True, False | Boolean values | User’s response to the confirmation dialog. |
| new\_username | signup | string | “newuser” | Plain text strings | New username for account creation |
| new\_password | signup | String | “newpass” | Plain text strings | New password for account creation |

MazeApplication Class Variables

| **Identifier** | **Location (Method)** | **Data Type** | **Typical Data (Examples)** | **Data Restrictions** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| self.root | Constructor (\_\_init\_\_) | tkinter widget | tkinter root window object | tkinter widget instance | Root widget of the tkinter application |
| self.user\_auth | Constructor (\_\_init\_\_) | UserAuthentication instance | Instnace of the UserAuthentication | Instance of UserAuthentication class | Handles user authentication and login logic |
| self.maze\_type | Constructor (\_\_init\_\_) | String | “Perfect”, “Non-Perfect” | defined settings | Stores the type of maze to be generated |
| maze\_id | delete\_maze | Integer | 1, 2, 3 | Positive integers | Identifier for a specific maze in the database |
| count | delete\_all\_mazes | Integer | 0, 10, 20 | Non-negative integers | Number of mazes in the database |
| response | delete\_all\_mazes | boolean | True, False | Boolean values | User confirmation response |
| maze\_str | save\_current\_maze | String | Comma and space separated string of integers | String representation of mazy array | Serialised string format of the maze for database storage |
| a, b | a\_star\_search (in heuristic) | Tuple | (0, 0), (5, 5) | Tuple of two integers | Coordinates used in the heuristic function of the A\* search |
| pos | a\_star\_search, get\_neighbors | Tuple | (0, 0), (1, 1) | Tuple of two integers | Position of a cell in the maze |
| neighbors | a\_star\_search, get\_neighbors | List of tuples | [(0, 1), (1, 0)] | List of tuples of two integers | List of walkable neighboring cells in the maze |
| x, y, dx, dy | a\_star\_search, get\_neighbors | List of tuples | [(0, 1), (1, 0)] | List of tuples of two integers | List of walkable neighboring cells in the maze |
| frontier | a\_star\_search | PriorityQueue instance | Instance of PriorityQueue | Instance of PriorityQueue class | Priority queue to determine the next cell to visit in A\* search |
| current | a\_star\_search | Tuple | (0, 0), (1, 1) | Tuple of two integers | The current cell being processed in the A\* search |
| next | a\_star\_search | Tuple | (0, 0), (1, 1) | Tuple of two integers | The next cell to visit in the A\* search |
| new\_cost | a\_star\_search | Integer | 1, 2, 3 | Non-negative integers | Cost of the new path to a neighboring cell in the A\* search |
| priority | a\_star\_search | Integer | 4, 5, 6 | Non-negative numbers | Priority value for the PriorityQueue in A\* search |
| came\_from | a\_star\_search | Dictionary | {(0, 0): None, (1, 1): (0, 0)} | Dictionary mapping tuples to tuples or None | Tracks the cell from which each cell was reached in A\* search |
| cost\_so\_far | a\_star\_search | Dictionary | {(0, 0): 0, (1, 1): 1} | Dictionary mapping tuples to integers | Tracks the cost to reach each cell in A\* search |
| path | a\_star\_search, find\_path, show\_path | List of tuples | [(0, 0), (1, 1), (2, 2)] | List of tuples of two integers | Represents the found path in the maze |
| height, width | Various | Integer | 15, 20, 30 | Positive integers | Dimensions of the maze (height and width) |
| start, end | find\_path, a\_star\_search | Tuple | (0, 0), (height - 1, width - 1) | Tuple of two integers | Start and end points for pathfinding the maze |
| canvas\_width, canvas\_height | set\_window\_size, display\_maze | Integer | 100, 200, 300 | Positive integers | Width and height of the canvas in pixels |
| widget | clear\_root | Tkinter widget | Individual tkinter widgets | Tkinter widget instances | Widgets in the root tkinter window |
| height\_label | make\_maze\_menu | Tkinter label | Tkinter label widget | Tkinter label widget instance | Label for displaying the text “Maze Height” |
| self.height\_slider | make\_maze\_menu | Tkinter scale | Tkinter scale widget | Tkinter scale widget instance | Slider for selecting the height of the maze |
| width\_label | make\_maze\_menu | Tkinter label | Tkinter label widget | Tkinter label widget instance | Label for displaying the text “Maze Width” |
| self.width\_slider | make\_maze\_menu | Tkinter scale | Tkinter scale widget | Tkinter scale widget instance | Slider for selecting the width of the maze |
| maze\_type | make\_maze\_menu | Tkinter StringVar | Tkinter StringVar object | Tkinter StringVar insance | Variable to hold the selected maze type |
| randomize\_button | make\_maze\_menu | tkinter button | tkinter button widget | tkinter button widget instance | Button to randomize maze dimensions |
| event | update\_size | Tkinter event | Tkinter even object | tkinter event instance | Event triggered by a UI action, like slider movement |
| maze\_id | delete\_maze | Integer | 1, 2, 3 | Positive integers | Identifier for a specific maze to be deleted from the database |
| maze\_id | regenerate\_saved\_maze | Integer | 1, 2, 3 | Positive integers | Identifier for a specific maze to be regenerated |
| maze\_str | regenerate\_saved\_maze | String | string of integers | serialized maze string | serialized string format of the maze retrieved from the database for regeneration |
| maze\_array | regenerate\_saved\_maze | List of lists | 2D integer array | 2D array of integers | Converted array format of the serialized maze string for display or manipulation |
| delete\_all\_button | my\_mazes | Tkinter button | Tkinter button widget | Tkinter button widget instance | Button to delete all saved mazes |
| scrollable\_frame | my\_mazes | Tkinter frame | Tkinter frame widget | Tkinter frame widget instance | Frame for a scrollable area to list saved mazes |
| canvas | my\_mazes | Tkinter canvas | Tkinter canvas widget | Tkinter canvas widget instance | Canvas associated with the scrollbar in the saved mazes listing |
| scrollbar | my\_mazes | tkinter scrollbar | tkinter scrollbar widget | tkinter scrollbar widget instance | Scrollbar for navigating through the list of saved mazes |
| inner\_frame | my\_mazes | tkinter frame | tkinter frame widget | tkinter frame widget instance | Inner frame inside the canvas to display the list of saved mazes |
| saved\_mazes | my\_mazes | List of tuples | [(1, 20, 20, ‘2023-01-01’), (2, 15, 15, ‘2023-01-02’)] | List of tuples with maze data | List of saved mazes retrieved from the database |
| maze | my\_mazes | Integer, Tuple | 1, (20, 20, ‘Maze data’, ‘2023-01-01’) | Positive integer, tuple with maze data | Used in the loop to process each saved maze information |
| maze\_frame | my\_mazes | Tkinter frame | Tkinter frame widget | Tkinter frame widget instance | Frame for each saved maze in the listing |
| maze\_label | my\_mazes | tkinter label | tkinter label widget | tkinter label widget instance | Label to display the information of each saved maze |
| regenerate\_button | my\_mazes | Tkinter button | tkinter button widget | tkinter button widget instance | Button to regenerate a specific saved maze |
| delete\_button | my\_mazes | Tkinter button | Tkinter button widget | Tkinter button widget instance | Button to delete a specific saved maze |
| button\_frame | add\_go\_back\_button\_generate\_maze | tkinter frame | tkinter frame widget | tkinter frame widget instance | Frame for holding a “Go Back” button in the maze generation menu |
| canvas | display\_maze | Tkinter canvas | Tkinter canvas widget | Tkinter canvas widget instance | Canvas where the maze is rendered |
| save\_button | display\_maze | tkinter button | tkinter button widget | tkinter button widget instance | Button to save the current state of the maze |
| quit\_button | display\_maze | Tkinter button | Tkinter button widget | Tkinter button widget instance | Button to quit the application |
| regenerate\_button | display\_maze | tkinter button | tkinter button widget | tkinter button widget instance | button to regenerate the maze based on current settings |
| find\_path\_button | display\_maze | tkinter button | tkinter button widget | tkinter button widget instance | Button to find and display a path through the maze |
| valid\_maze | generate\_maze | boolean | True, False | Boolean values | Flag indicating whether a valid maze has been generated |
| maze\_generation\_attempts | generate\_maze | Integer | 0, 1, 2 | Non-negative numbers | Counter for the number of attempts made to generate a valid maze |
| start | generate\_maze | tuple | (0, 0) | Tuple of two integers | Starting point for maze generation algorithms |
| end | generate\_maze | tuple | (49, 49) | Tuple of two integers | End point for maze generation algorithms |

Global Variables

| **Identifier** | **Location (Method)** | **Data type** | **Typical data (examples)** | **Data restrictions** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| root | Main execution block | Tkinter tk instance | Tkinter root window object | Tkinter tk instance | Main application window the tkinter GUI |
| app | Main execution block | MazeApplication instance | Instance of MazeApplication | Instance of MazeApplication class | Instance of MazeApplication class to run the application |
| height | recursive\_backtracker, prims\_algorithm, carve\_passage\_from | Integer | 15, 20 | Positive integers | Height of the maze used in maze generation algorithms |
| width | recursive\_backtracker, prims\_algorithm, carve\_passage\_from | Integer | 15, 25 | Positive integers | Width of the maze used in maze generation algorithms |
| cx | in carve\_passage\_from of recursive\_backtracker | Integer | 0, 1, 2 | Positive integers | Current cell’s x coordinate in the maze during passage carving |
| cy | in carve\_passage\_from of recursive\_backtracker | Integer | 0, 1, 2 | Positive integers | Current cell’s y coordinate in the maze during passage carving |
| grid | in carve\_passage\_from of recursive\_backtracker | List of lists | [[0, 0, 1], [1, 0, 1]] | 2D array of integers (1 or 0) | Represents the maze grid during passage carving |
| directions | in carve\_passage\_from of recursive\_backtracker | List of tuples | [(cx - 1, cy), (cx + 1, cy), (cx, cy - 1), (cx, cy + 1)] | List of tuples of two integers | directions to carve passages in the maze |
| nx | in carve\_passage\_from of recursive\_backtracker | Integer | 0, 1, 2 | Positive integers | Next cell’s x coordinate in the maze during passage carving |
| ny | in carve\_passage\_from of recursive\_backtracker | Integer | 0, 1, 2 | Positive integers | Next cell’s y coordinate in the maze during passage carving |
| maze | recursive\_backtracker | List of lists | [[0, 0, 1], [1, 0, 1]] | 2D array of integers (1 or 0) | Represents the maze grid structure in the recursive backtracker algorithm |
| start\_x, start\_y | recursive\_backtracker | Integer | 0, 5, 10 | Positive integers | Starting x and y coordinates for maze generation in the recursive backtracker algorithm |
| local\_maze | prims\_algorithm | List of lists | [[0, 0, 1], [1, 0, 1]] | 2D array of integers (0 or 1) | Represents the maze grid structure in prim’s algorithm |
| visited | prims\_algorithm | Set of tuples | {(0, 0), (1, 1)} | Set of tuples of two integers | Tracks visited cells in the maze during prim’s algorithm |
| walls | prims\_algorithm | Set of tuples | {(0, 1), (1, 0)} | Set of tuples of two integers | Tracks the walls in the maze during prim’s algorithm |
| start | prims\_algorithm | Tuple | (0, 0) | Tuple of two integers | Starting cell in the maze for prim’s algorithm |
| wall | prims\_algorithm | Tuple | (0, 1), (1, 0) | Tuple of two integers | Represents a specific wall being considered for removal in prim’s algorithm |
| neighbors | prims\_algorithm | List of tuples | [(0, 0), (1, 1)] | List of tuples of two integers | Neighboring cells of a wall in the maze during Prim’s algorithm |