**Artificial Intelligence Mini Project Report**

|  |  |
| --- | --- |
| **Mini Project Title** | Shortest Path Finder Project Using BFS |
| **Problem Statement:** | The project aims to develop a shortest path finder using the Breadth-First Search (BFS) algorithm. The application allows users to interactively create a grid, set starting and ending points, and add obstacles. The BFS algorithm then determines the shortest path between the specified starting and ending points while avoiding obstacles. |
| **Programming Language Used:** | PYTHON |
| **Software Used:** | JUPYTER NOTEBOOK |
| **Name of Dataset:** | There is no external dataset used in this project. The grid and obstacles are created interactively by the user during runtime. |
| **Link to the Dataset:** | Not applicable. |
| **AI Methodology** | **Turtle:** To do the overall visualisation of the entire process  **Tkinter:** To show info if there is not path available between start and end points  **random:** to place the obstacles randomly at any point of the grid  **numpy:** to create 2d arrays to be used as adjancency matrix , also used in making some other necessary arrays.  **Implementation Overview:**   * The project comprises two main classes: **BFS** and **Body**.   + The **BFS** class initializes the grid, defines the adjacency matrix, and implements the BFS algorithm to find the shortest path.   + The **Body** class handles the graphical interface, user interactions, and visualization of the grid.   **Usage Instructions:**   1. Press 's' to set the starting point. 2. Press 'e' to set the ending point. 3. Press 'h' to place obstacles on the grid. 4. Press 'r' to place random obstacles. 5. Press the space bar to find and display the shortest path. 6. Press 'o' to clear the grid. |
| **Name of Algorithm:** | Breadth First Search(BFS) |
| **Output/ Screenshots(software)**   1. **Accuracy result** 2. **Visualization Plots** 3. **Input-Output Screenshots** | Green box indicating starting point  . Blue box indicating ending point . Red boxes indicating Hurdles  Red  Black line indicates the Shortest Path to reach its destination. |
| **Interpretation from Result/ Conclusion:** | A window is opened in front of the user once this program is runned. A set of quick instructions is provided at the top of the window. A grid is present on the window , where the user can put the starting and ending point by choosing the respective key on the keyboard and clicking on any block of the grid. User can also place hurdles(obstacles) one by one in the grid, by clicking in the particular position in the grid. Each press of 'r' in the keyboard places 100 hurdles(obstacles) at random positions in the grid. After that the user can hit the "spacebar" to see the program finding out the shortest path between start and end point. Press 'o' to clear the grid. |
| **Name of Students and sign:** | 1. Vivek Balasaheb Auti 2. Harshada Vijay Jagtap |
| **Date of Submission:** | 01/12/23 |

Prof. A. H. Hingmire

Subject In-charge