Autonomous robotics Final Project: Runtime documentation

Introduction

This guide helps you run through the Autonomous Robotics project, which involves terrain generation, autonomous exploration using an agent, and path finding using the Rapidly-exploring Random Tree (RRT) algorithm. Follow these steps to run the simulation and achieve the outputs.

Prerequisites

- Python environment setup with necessary libraries (`numpy`, `matplotlib`, `scipy.ndimage import gaussian_filter`, etc.).
- The project consists of four main files: `Agent.py`, `TerrainGen.py`, `RRT.py`, and `main.py`.
- Ensure all these files are located in the same directory.

Step-by-Step Instructions

Step 1: Terrain Generation

1. **Start with Terrain Generation**: The `TerrainGen.py` file is responsible for generating the terrain.
   - The `TerrainGenerator` class is initialized with parameters like seed, max and min elevation, and the number of trees.
   - For consistency, we are keeping seed = 125 for out experimentation and results. You can also change the max and min elevation in `TerrainGenerator` class parameters.
   - The `generate_terrain_with_trees()` method creates a terrain matrix that represents the generated terrain with elevations and obstacles (trees).

Step 2: Agent Initialization

2. **Initialize the Agent**: In the `Agent.py` file, an `Agent` class is defined.
   - Create an instance of the `Agent` class, specifying the start position, step size, and other parameters.
   - The Agent has init parameters like the starting point of the agent that can be edited, and also the details regarding the step size for the RRT and also the Iteration limit for the RRT and the Agents Discovery radius during exploration phase.
   - The agent is responsible for exploring the terrain and interacting with the environment.

Step 3: Path finding with RRT

3. **Implemented RRT for Path finding**: The `RRT.py` file contains the `RRT_Exp` class.
   - Initialize this class with the terrain matrix, start and goal points, and the agent.
   - Use methods like `expand_tree()` and `find_path()` to generate and find a path from the start to the goal.
   - There are multiple other functions that help in the rrt implementation.

Step 4: Running the Main Simulation

4. **Execute the Main Simulation**: The `main.py` file integrates all components.
   - Run this file to start the simulation.

- It initializes the terrain and the agent, sets up the simulation environment, and handles GUI interactions.
- Initially, a plot will appear with the entire terrain as below.

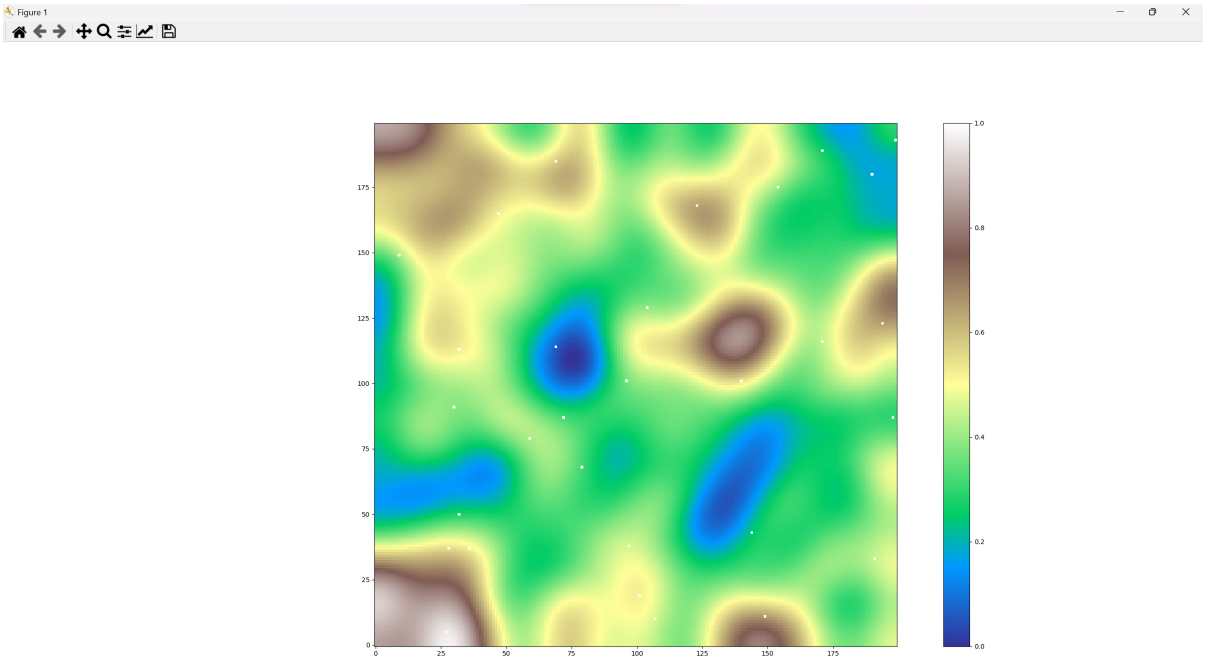

Figure 1: Terrain Generated

- Next if you press any of the arrow keys it will go into the exploration mode with the point Agent at the starting location as red dot with its discovery radius as initialized.
- Pressing Up key drives the car up but the updation is key based so the velocity to the agent is provided but unless you hold the key or keep pressing it the updated position will not appear as below.
- The boundaries doesn not stop the agent but dont add onto the exploration matrix to avoid cost matix bounds and still allow for bette ragent control to avoid losing the agent in exploration.
- Watch the agent as it explores the terrain.
- Once you have explored as needed, press “x” key on the keyboard to end the exploration phase.

Step 5: Interaction and Visualization

5. **Interact with the Simulation after exploration**:
   - Once the exploration has ended and you click Ok on the message box, you can then select 2 point using mouse pointer on the explored area which will be indicated on the Plot as yellow dots.
   - These can only be selected on the explored region and not the unexplored regions and once these are selected, it triggers the RRT functionality.
   - Select start and end points, and observe the agent’s pathfinding in action (depending on the path and locations the pathfinding may take a while and also reruns as the iterations and step size may need tweaking which is a sort of limitation we need to address later).

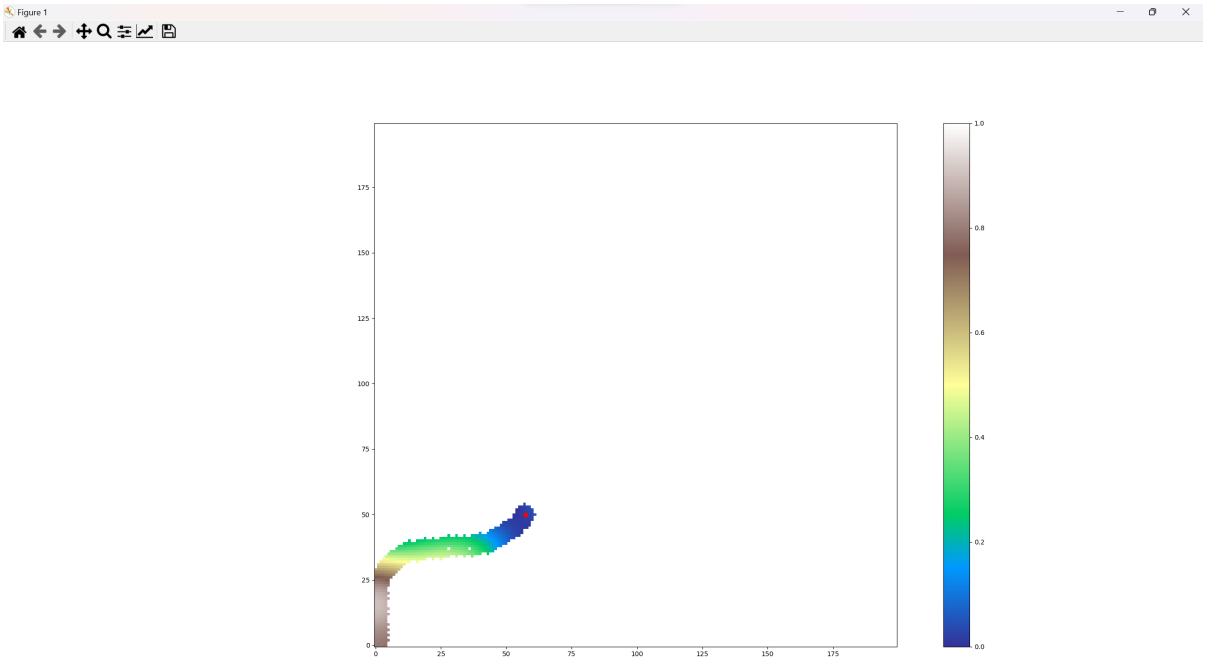

Figure 2: Terrain Exploration

Figure 3: Exploration Ended

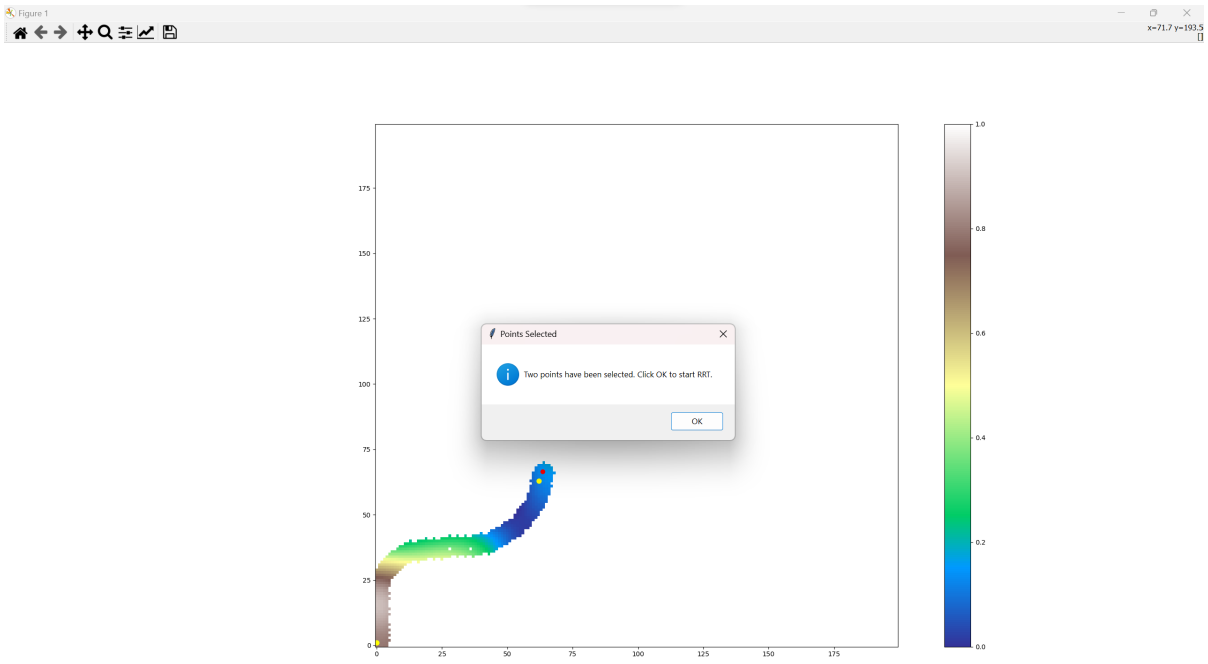

Figure 4: Start and End locations

Step 6: Analyzing the Output

6. **Analyze the Results**:
    - The final output is a visualization of the agent's exploration path and the path planned using RRT.
    - The Original plot gets updated with the explored Tree paths in blue and in the same location as the files it saves a file called `Exploration.gif` which has the animation of the agent traversal from the start to end ina magenta colored path.
    - Analyse how different parameters (like terrain complexity, agent's step size, etc.) affect the path finding i.e. path avoid the brown-white peaks and prioritizes visiting the deep blue depths as marked in color bar.

Step 7: Modifications and Experimentation

7. **Experimentation with Different Scenarios**:
    - Modify parameters in the `TerrainGenerator`, `Agent`, and `RRT_Exp` classes to see influence on the agent's behavior and path finding efficiency.
    - Experiment with different terrain types, agent capabilities, and goal locations.

Conclusion

By following these steps, you can effectively run and interact with the Autonomous Robotics project. Each component plays a crucial role in simulating a realistic exploration and pathfinding scenario. Experiment with different configurations to explore the capabilities of your autonomous agent in varying environments.

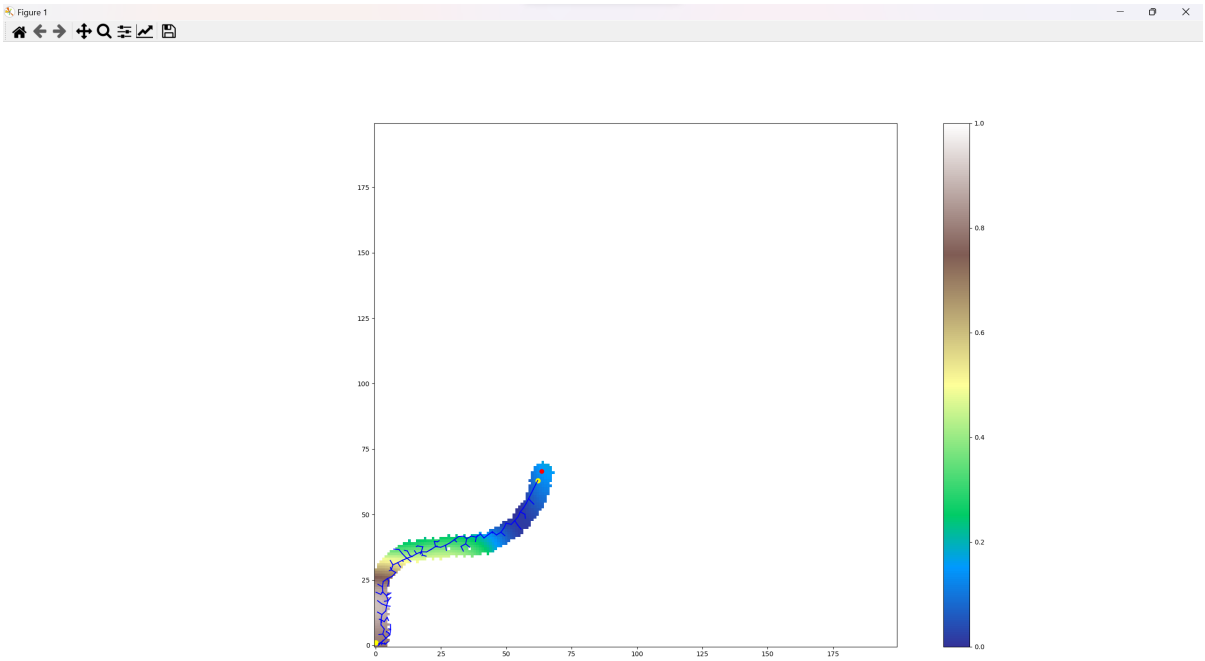

Figure 5: RRT path

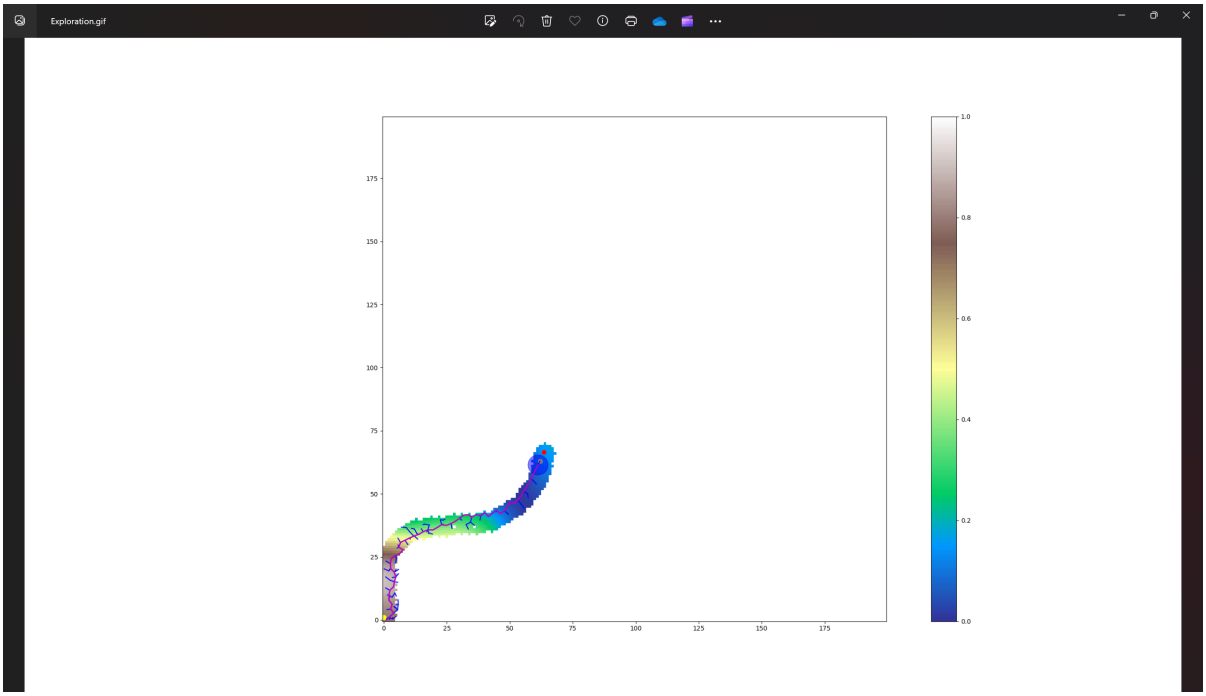

Figure 6: RRT Final path