AI Based Simulation Of Navigation For Path Finding And Obstacle Avoidance

Submitted in partial fulfillment of the requirements of

Third Year

in

B.E(Artificial Intelligence and Data Science)

By

Shubhankar Gite	**19**
Vedant Ranade	**58**
Trushank R Vashikar	**71**
Shreyas Yesade	**76**

Supervisor

Dr. Preeti Jain

Department of Artificial Intelligence and Data Science

DATTA MEGHE COLLEGE OF ENGINEERING, AIROLI, NAVI MUMBAI - 400 708

University of Mumbai

(AY 2023-24)

CERTIFICATE

This is to certify that the Mini Project 1A entitled "Title**"** is a bonafide work of **Shubhankar Gite (19) Vedant Ranade (58) Trushank Vashikar (71) Shreyas Yesade (76)** Submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Artificial Intelligence and Data Science".**

(Prof. Preeti Jain)
Supervisor

(Dr. S M Patil)
Head of Department

Dr. D J Pete.)
Principal

Mini Project Approval

This Mini Project entitled " **AI Based Simulation Of Navigation For Path Finding And Obstacle Avoidance**" by **Shubhankar Gite (19) Vedant Ranade (58) Trushank Vashikar (71) Shreyas Yesade (76)** is approved for the degree of **Bachelor of Engineering** in **Artificial Intelligence and Data Science.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place:

DECLARATION

I declare that this project represents my ideas in my own words without plagiarism and wherever others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my project work. I promise to maintain minimum 75% attendance, as per the University of Mumbai norms. I understand that any violation of the above will be cause for disciplinary action by the Institute.

Yours Faithfully

1._____

2._____

3._____

4._____

(Date & Signature of Students)

Contents

Abstract

Artificial Intelligence can be used for the navigation and traversal in a bound space here both the source and goal nodes are predefined. This pathfinding also includes the avoidance of obstacles and any kind of obstruction . Visualizing and imagining this navigation and obstacle avoidance could be incomprehensible for real life or even a virtual scenario . Thus this project aims to use artificial intelligence to its utmost potential to derive algorithm(s) for the navigation and obstacle avoidance as well as simulating the given path in a virtual environment .

Acknowledgement

In the report at hand, profound gratitude is extended to Dr. Preeti Jain, our project guide, for her unwavering support, expert guidance, and invaluable resource contributions. It is worth noting that her mentorship played a pivotal role in shaping the success of this project.

In addition, recognition and appreciation are extended to Dr. Sanjay M. Patil, Head of the Department, and Dr. D.J Pete, Principal of the institution.

The entire Department of Artificial Intelligence and Data Science is duly acknowledged for their steadfast support, provision of excellent facilities, and unwavering encouragement throughout the duration of this project.

List of figures :-

List of abbreviations :-

1. AI ;- Artificial Intelligence
2. ML :- Machine Learning
3. CARLA :- Car Learning to Act
4. GUI :- Graphical User Interface
5. GPS :- Global Pointing System
6. TCP :- Transfer Control Protocol
7. IP :- Internet Protocol
8. AIRSIM :- Aerial Informatics and Robotics Simulation
9. SMTPLIB :- Simple Mail Transfer Protocol Library
10. 3D :- Three Dimensional
11. LiDAR :- :Light Detection and Ranging
12. SMS :- Short Message/Messaging Service
13. API :- Application Programming Interface
14. DFS :- Depth First Search
15. BFS :- Breadth First Search

List of Tables :-

1. Introduction

1.1. Introduction

This report presents a comprehensive overview of the research and the outcomes , titled "Artificial Intelligence in Pathfinding and Obstacle Avoidance." The project's inception was marked by an extensive exploration of navigation and obstacle avoidance algorithms, serving as the foundational framework upon which this endeavor was built. The initial phase of this undertaking involved meticulous research and comparative analysis, which encompassed a thorough examination of various algorithms to enrich our comprehension of the field.

As the project advanced, it became apparent that, despite our proficiency in algorithmic aspects, there existed a limitation in our knowledge regarding simulation techniques. This recognition prompted the commencement of a parallel exploration dedicated to simulation methods, with the goal of amalgamating our algorithmic expertise with practical simulation techniques. This strategic approach aimed to create a more comprehensive and holistic solution to the intricacies of pathfinding and obstacle avoidance. This report serves as a documentation of the development process, insights gleaned, and critical findings acquired during this journey, with an emphasis on the growth and adaptability demonstrated throughout the project's duration

1.2. Problem Statement and Objectives

1.3.1. Problem Statement :-

Implementing AI navigation algorithms for path finding and avoidance of obstacles and simulating the navigation and obstruction avoidance using real time simulation engines.

1.3.2. Objectives:-

1. AI Navigation:

- To implement and develop AI n navigation algorithms tailored for pathfinding and obstacle avoidance in real-world environments.
- To ensure that the AI-driven navigation system is capable of adapting to obstacles and

other environmental conditions, thereby enhancing safety and efficiency.

2. Comparative Analysis of Various Algorithms:

- To conduct a comprehensive comparative analysis of various pathfinding and obstacle avoidance algorithms, evaluating their strengths, weaknesses, and suitability for different scenarios.
- To elucidate the advantages and limitations of each algorithm, providing valuable insights for informed decision-making in algorithm selection.

3. Real-Time Simulation:

- To design and construct a real-time simulation environment that mirrors real-world conditions
- To facilitate the practical testing and visualization of the AI navigation algorithms in this simulation, enabling assessment of their performance .

4. User Interface Development:

- To design an user-friendly interface that allows users to interact with the simulation, set navigation goals, and monitor the AI-driven navigation system's progress in real-time.
- To implement sending notification to the end users when the goal state has been reached

5. Machine Learning Integration:

- To explore the integration of machine learning techniques for the AI navigation system, enabling it to adapt and improve its performance based on past navigation experiences.
- To develop machine learning models that can enhance the system's decision-making processes and further optimize navigation and obstacle avoidance.

1.4.Organization of the Report

As shown in Section 1, this chapter covers the introduction, motivation, and objectives of the project. Chapter 2 the literature survey, discussing existing systems, their implementations, and their limitations. Chapter 3 analyzes the proposed methodology, where different simulation engines are discussed. In Chapter 4, the overall results of this project are examined. We conclude with Chapter 5, which includes the conclusion and outlines the future scope of this project

2. Literature Survey

2.1. Existing system:-

2.1.1. OpenAI Gym

OpenAI Gym is a important device inside the global of synthetic intelligence and reinforcement studying. It's an open-source platform that offers researchers with a standardized way to create and test reinforcement getting to know algorithms.[7] One of its standout functions is the extensive variety of simulated environments it gives. Researchers can work on something from easy duties to complex video games, making it easy to evaluate their algorithms always and song their progress. OpenAI Gym's consumer-pleasant functions, detailed documentation, and great library of environments have extensively speed up research and improvement within the AI area. It's a precious aid for creating clever agents which could cope with actual-world problems. So, when doing a survey of existing structures, it is important to apprehend OpenAI Gym's impact on AI studies and the precious centers it offers for experimentation and innovation.[6]

2.1.2. CARLA:

CARLA, referred to as "Car Learning to Act," performs an important position inside the realm of self-riding automobile research. Similar to OpenAI Gym, CARLA gives a specialized platform designed to help scientists and engineers in their paintings on self-sufficient riding.[1] Its capabilities as exceedingly practical virtual surroundings, like a sophisticated video game, where researchers can check and improve self-using automobile programs. CARLA excels at simulating complicated town riding conditions, offering a rich set of facilities for checking out and enhancing self-riding generation. These centers consist of unique urban environments, sensible automobile management systems, equipment for path making plans, and abilities for education of self-driving structures through getting to know and enjoy. When conducting a literature survey to explore current systems, it is important to highlight CARLA's pivotal role in growing the self-using generation, which has the capability to make our roads safer and transportation greener.[3]

2.2. **Limitations**:-

2.2.1 OpenAI Gym:

OpenAI Gym, a broadly used platform for reinforcement learning, has numerous barriers really worth noting. One number one difficulty is its single-agent cognizance. While Gym is first rate for unmarried-agent situations, it's far less ready for multi-agent environments. Additionally, among the environments furnished are deterministic, missing the stochasticity and randomness found in actual-international situations. This can restrict the platform's applicability to troubles concerning uncertainty. Furthermore, Gym might also pose scalability demanding situations, especially while dealing with complex and huge-scale fashions. It won't be optimized for high-overall performance computing on allotted systems, which could restrict its utility for positive high-complexity duties. Despite those barriers, OpenAI Gym remains a valuable device for unmarried-agent reinforcement learning research and experimentation.[6]

2.2.2 CARLA:

CARLA, the Car Learning to Act platform, whilst a precious useful resource for self-reliant using studies, does have specific obstacles. First, its simulation realism, even as advanced, might not completely replicate the complexities of real-world. Factors like unpredictable weather conditions and dynamic pedestrian behavior can be difficult to version correctly. Moreover, running CARLA, especially with high-fidelity settings, can be computationally demanding, probably necessitating powerful hardware and restricting accessibility for a few research teams. Developing custom environments in CARLA can also be complex and time-ingesting, requiring a learning curve for builders. Achieving actual-time overall performance in CARLA, in particular whilst implementing superior manage algorithms or the usage of high-resolution sensor statistics, may be a task and can require optimization. These barriers, at the same time as gift, do no longer decrease CARLA's price as a studies tool, imparting a managed and reproducible environment for experimenting with self-reliant driving algorithms in diverse eventualities.[3]

2.3. **Miniproject contribution**

AI-based navigation and obstacle avoidance projects have notably improved Enhanced Autonomous Systems by enhancing the navigation capabilities of autonomous vehicles and robots. These projects also contribute to Simulation for Training, offering a safe and cost-effective method to train AI systems by exposing them to various scenarios. Realistic Gaming benefits from AI-driven simulations, creating more immersive and responsive gameplay experiences. Additionally, thid project play a crucial role in Safety and Risk Mitigation, particularly in aerospace, by modeling system performance to identify and address potential issues before they manifest in real-world applications, ultimately enhancing safety and reducing risks.

3. Proposed Systems

3.1. Introduction to proposed system

3.1.1. Overview

In the ever-evolving landscape of autonomous technology, our proposed system represents a significant advancement in the field of drone navigation. The primary goal is to facilitate the effortless and secure movement of drones from one user-specified location to another. By simply inputting the starting and ending coordinates, operators can rely on the system to autonomously manage takeoff, path planning, and obstacle avoidance, ensuring the drone reaches its intended destination safely. This innovation not only underscores the expanding role of drones in various industries but also underscores the importance of enhancing their capabilities for applications like agriculture, infrastructure inspection, emergency response, and environmental monitoring.

At the heart of this system lies state-of-the-art technology, including machine learning, computer vision, and real-time data analysis, enabling drones to make informed decisions during flight. By leveraging data from a myriad of sensors and cameras, the system can identify and circumvent obstacles, whether they are stationary or in motion. Furthermore, our simulation-based approach enables rigorous testing and refinement, mitigating risks associated with real-world drone operations. This system, with its ability to streamline drone operations and ensure safer and more precise missions, promises to be a transformative tool across a wide range of industries, heralding a future where drones become indispensable partners in tasks demanding aerial precision, efficiency, and security.

3.1.2. Search of a suitable algorithm

In the process of developing our autonomous drone navigation system, we conducted a search to identify a suitable algorithm for path planning and obstacle avoidance. Our approach involved considering a range of basic algorithms to assess their compatibility with the software and to lay the foundation for potential future improvements. The algorithms we evaluated were:

3.1.2.1. DFS

DFS is one of the fundamental graph traversal algorithms. In the context of your autonomous drone navigation system, DFS may have been considered for path planning and exploration. DFS explores as far as possible along a branch before backtracking, which can be useful in scenarios where a drone needs to explore a given area thoroughly. However, DFS may not always be the best choice for finding the shortest path, which is crucial for efficient drone navigation.

3.1.2.2. BFS

BFS is another classic graph traversal algorithm. It explores neighbors of a node before moving to their children. In the context of your system, BFS might have been considered for path planning. BFS guarantees that the shortest path is found first but can be computationally expensive in large grids. It's more suitable for scenarios where finding the shortest path is a priority over exploration.

3.1.2.3. Dijkstra :

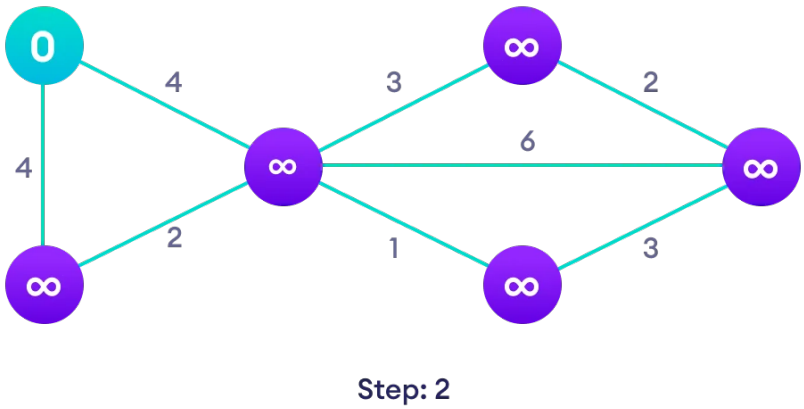

Figure3.1.2.3. Example of Dijkstra

Dijkstra's algorithm is a well-known algorithm for finding the shortest path in a weighted graph. It is an excellent choice for scenarios where drone navigation involves varying distances between grid points, such as varying wind speeds or terrain. Dijkstra's algorithm guarantees finding the shortest path, but it may not be the most efficient option for large grids or complex environments

3.1.2.4. A * Algorithm

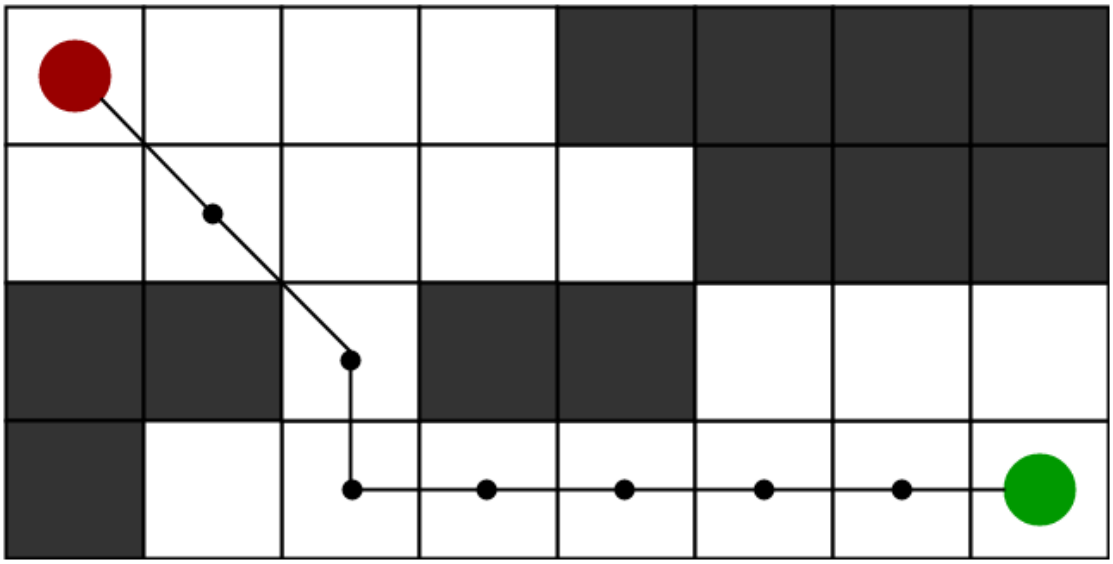

Figure 3.1.2.4. Example of A star algorithm

The A* algorithm, or A-star, is a widely-used heuristic search algorithm for finding the shortest path. A* combines the benefits of both BFS and Dijkstra's algorithm by considering not only the cost to reach a point but also an estimation of the remaining cost to reach the goal. This makes A* highly efficient and often the algorithm of choice for path planning in autonomous systems like your drone navigation system. It tends to find the shortest path while also being computationally efficient.

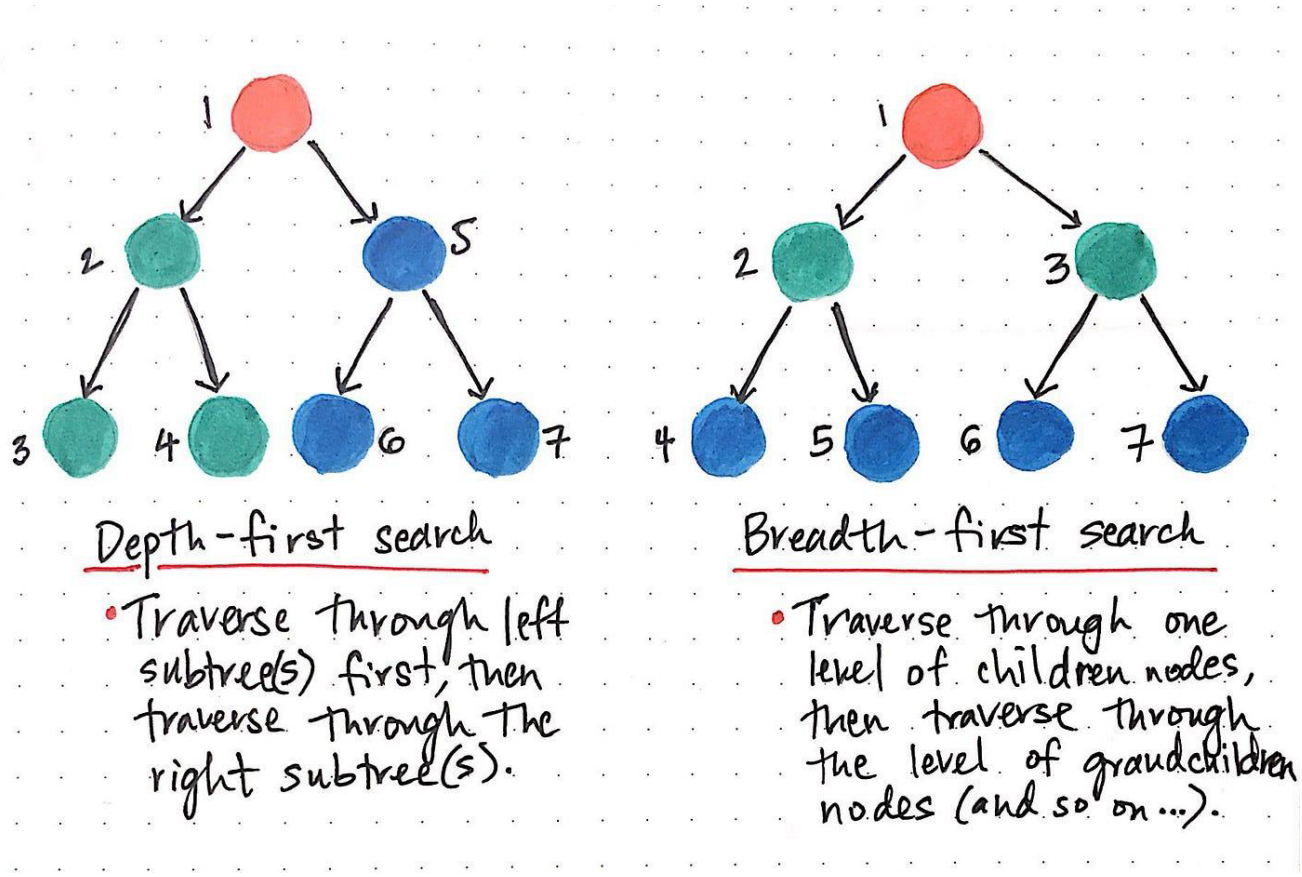

Fig 3.1.3. Comparison of BFs and DFS

3.1.3. Compare n Contrast

DFS	BFS	A*	Dijkstra
DFS explores as far down a branch as possible before backtracking.	BFS explores all neighbours of a node before moving on to their neighbors.	A* is also a weighted graph algorithm that combines information about the cost to reach a node from the start and a heuristic estimate of the cost to reach the goal.	Dijkstra's algorithm is a weighted graph algorithm that explores nodes in order of their distance from the source node.
DFS is not guaranteed to find the shortest path; it can get stuck in infinite loops in the presence of cycles.	It guarantees finding the shortest path in unweighted graphs.	A* can find the shortest path in graphs with non-negative edge weights when using an admissible heuristic.	It guarantees finding the shortest path in graphs with non-negative edge weights.

It uses less memory compared to BFS.	It can be memory-intensive, especially in large graphs.	It can be memory-intensive, but it's more memory-efficient than BFS or DFS.	It uses more memory compared to BFS.

Table 3.1.4. Compare and Contrast

A* stands out as a better option for AI applications compared to DFS, BFS, and Dijkstra's algorithm because it combines the ability to find the shortest path with the flexibility to incorporate heuristic estimates. This makes it highly efficient and effective in AI scenarios where pathfinding, planning, and decision-making require consideration of both optimality and computational efficiency.

3.2. **Architecture**

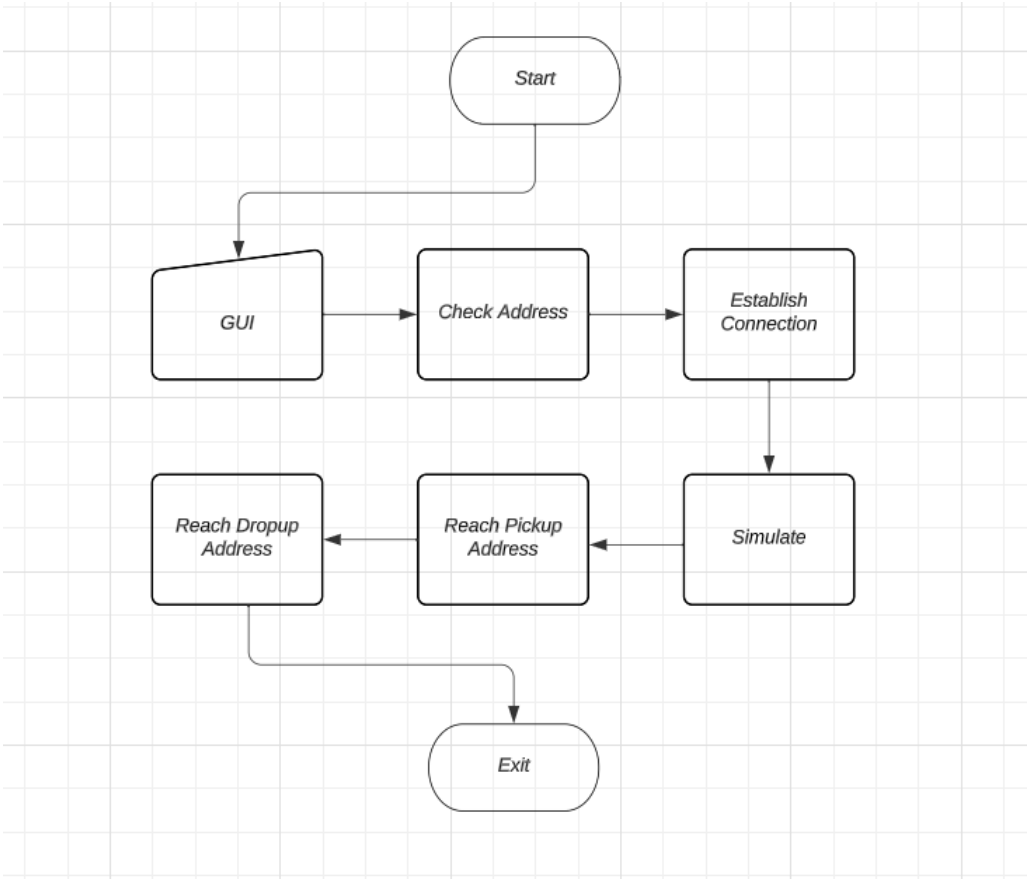

Figure 3.2.1 Architecture of Navigation Simulation

Figure 4.2.1 shows the pipeline working of the project , The user on execution of application will first get a GUI which has multiple options connoting Home, about us, Delivery , Contact us . when user wants to send Delivery then the program first authenticates the address of both

pickup and drop to ensure that it is in valid area of simulation then the connection is established via TCP/IP protocols with the C# script loaded in the simulator. Then the Simulation starts and both the addresses are reached at the same time notified to the use .

3.3 Algorithm and Process Design

3.3.1. Algorithm Overview :

The fundamental success of this project hinges on the intelligent path planning and obstacle avoidance capabilities of the A* (A-star) algorithm. A* is a widely-used, heuristic search algorithm known for its efficiency in finding the shortest path between two points on a grid while considering obstacles. In the context of our autonomous drone navigation system, the algorithm takes center stage in computing the drone's flight path.

3.3.2. Path Planning with A* :

The A* algorithm operates in two key steps: expansion and exploration. It begins with the starting point and iteratively explores adjacent nodes, known as "neighbors," expanding the search tree. At each step, it assigns a cost to each potential path and selects the one with the lowest cost. The cost calculation typically includes factors such as distance traveled and a heuristic estimation of the remaining distance to the goal, ensuring the algorithm tends to select paths that are likely to lead to the goal efficiently.

3.3.3. Obstacle Avoidance :

Incorporating obstacle avoidance is a critical aspect of the process design. As the drone navigates the simulated landscape, it constantly senses its surroundings and recognizes obstacles using sensors integrated with the AirSim platform. When an obstacle is detected, the A* algorithm adapts by considering alternative paths that bypass the obstacle. This dynamic path adjustment ensures that the drone avoids collisions while continuously progressing toward the destination.

3.3.4. First attempt of visualization

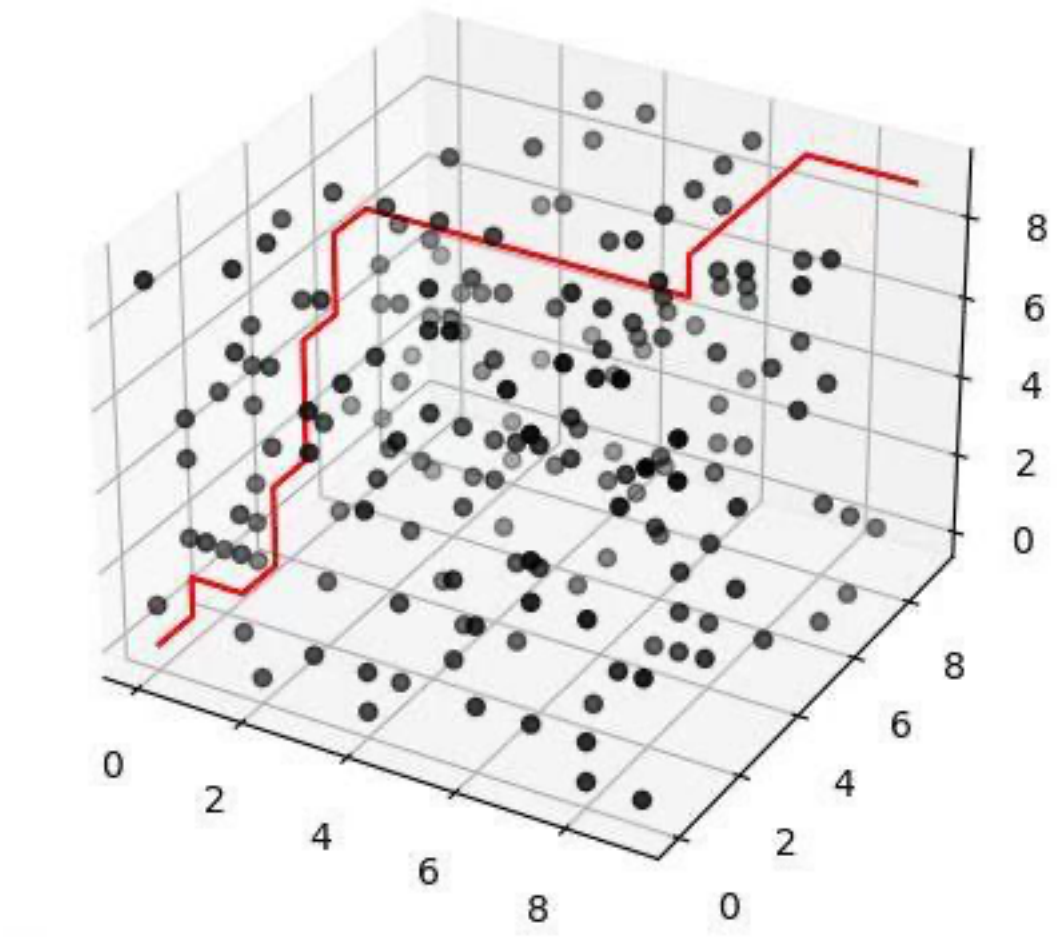

Fig 3.3.4. Visualization of A star

By the comparative study , we concluded by choosing A* algorithm as the shortest path finding algorithm, And as of now for testing we have generated a 10x10x10 grid. Where the start point is (0,0,0) and the end point ins (10,10,10) . And the obstructions will placed randomly .Thus the following show the implementation of A star and the shortest path and obstacles avoidance .

Process Design

- **Start and Destination:** The process begins with the user specifying a start point and a destination using GPS coordinates or other means.
- **A* Path Planning:** The A* algorithm computes an initial path from the start to the destination, taking into account the landscape's features.

- **Obstacle Detection:** As the drone moves through the simulation, sensors actively detect obstacles and their positions, updating in real-time.
- **Dynamic Path Adjustment:** Upon detecting an obstacle in the drone's path, the A* algorithm recalculates the path to navigate around the obstruction.
- **Continuous Monitoring:** The process is ongoing, with the drone continually sensing its environment and adjusting its path as needed, ensuring real-time obstacle avoidance.
- **Completion and Reporting:** Once the destination is reached, the system generates a report, detailing the path taken, obstacles avoided, and any other relevant information.

  By incorporating the A* algorithm for path planning and real-time obstacle avoidance, this project's process design ensures safe and efficient autonomous drone navigation in complex landscapes, providing a solid foundation for various applications and further research in the field of autonomous systems.

3.4. Details of Software.

i) Python is used to by the AI algorithm
ii) C-Sharp is used as it is only language supported in Unity
iii) Various libraries of Python were also used such as:
   a) smtplib for email feature
   b) Matplotlib for Visualization of Algorithm during analysis
   c) socket library for connection between python and c# in different applications
iv) Unreal
v) Airsim

3.5. Conclusion and Futurework

3.5.1. Conclusion

In conclusion, the proposed system represents a way forward in the domain of autonomous navigation and obstacle avoidance. It offers a user-friendly solution for safe and efficient navigationa; operations, underscoring the growing role of drones in diverse applications. With its ability to autonomously path planning, and obstacle avoidance, this system streamlines various applications like drones , cars , toys , household robots etc .

The core of this system relies on cutting-edge technologies such as artificial intelligence , machine learning, computer vision, and real-time data analysis. By harnessing data from a multitude of sensors and cameras etc , it empowers the agent to make informed decisions during navigation, enhancing their reliability and safety. The simulation-based approach allows rigorous testing and fine-tuning, mitigating real-world operations.

As we look to the future, this system promises to play a transformative role in various sectors, including agriculture, infrastructure inspection, emergency response, and environmental monitoring. With continued development and integration into more industries, the potential for autonomous navigational systems is limitless.

3.5.2. Future Work:

i. Hardware Integration: Explore the integration of advanced hardware, such as LiDAR and improved cameras, to enhance obstacle detection and navigation precision.

ii. Machine Learning : Continuously train and fine-tune machine learning models to improve object recognition and path planning based on evolving environmental conditions.

iii. User Interface Enhancement: Improve the user interface to make it more intuitive and user-friendly, catering to a broader range of operators.

iv. Security Measures: Implement robust security features to safeguard the system against potential threats or unauthorized access.

v. Cost Optimization: Work on cost reduction strategies to make such systems more accessible to a wider range of users and industries.

4. Implementation and Result

4.1 System implementation

Unity and Unreal Engine are two popular game development platforms that offer powerful tools for creating interactive 3D environments and games. While they are primarily known for game development, both engines can also be used for various other applications beyond gaming.[8][9] One of the remarkable aspects of both Unity and Unreal Engine is their flexibility in accommodating a wide range of projects, from architectural visualizations to virtual training simulations.[8]

This combination of Unity, Unreal Engine, and Python offers developers a powerful set of tools to create interactive, visually stunning simulations and applications, whether for game development, scientific research, architectural visualization, or immersive training environments. The ability to leverage Python's computational prowess within these engines opens up a world of possibilities for those looking to create sophisticated, real-time experiences.[10]

Python, a versatile and widely-used programming language, plays a significant role in enhancing the capabilities of Unity and Unreal Engine. Its robust libraries and frameworks make it an excellent choice for computation of algorithms and simulation within these engines.

4.1.1. Unity Simulation

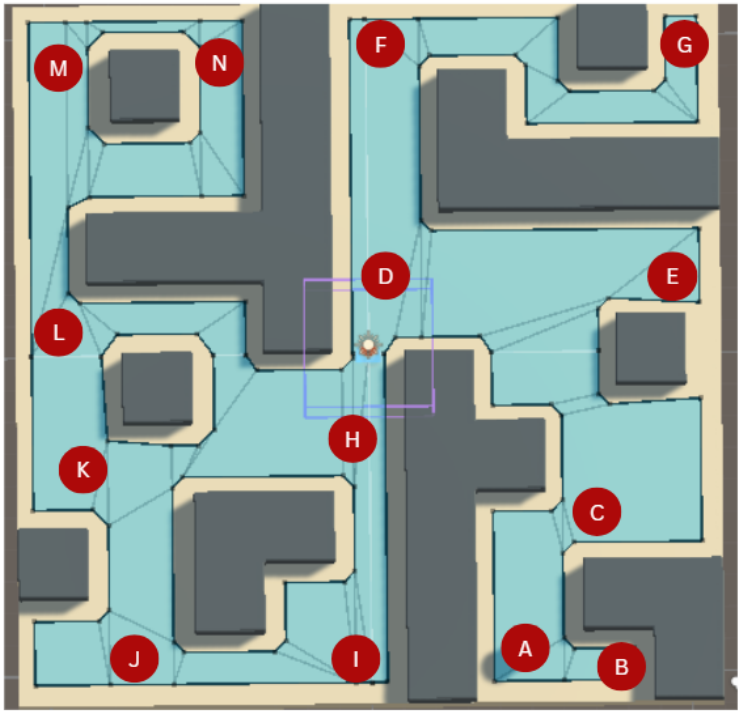

Figure 4.1.1.1 Map for simulation created in unity

From figure 4.4.1.1 we can visualize the static map or level which is created using unity tools and we have identified few points whose coordinates are already known to the humanoid agent who can then easily navigate between these points using A* algorithm

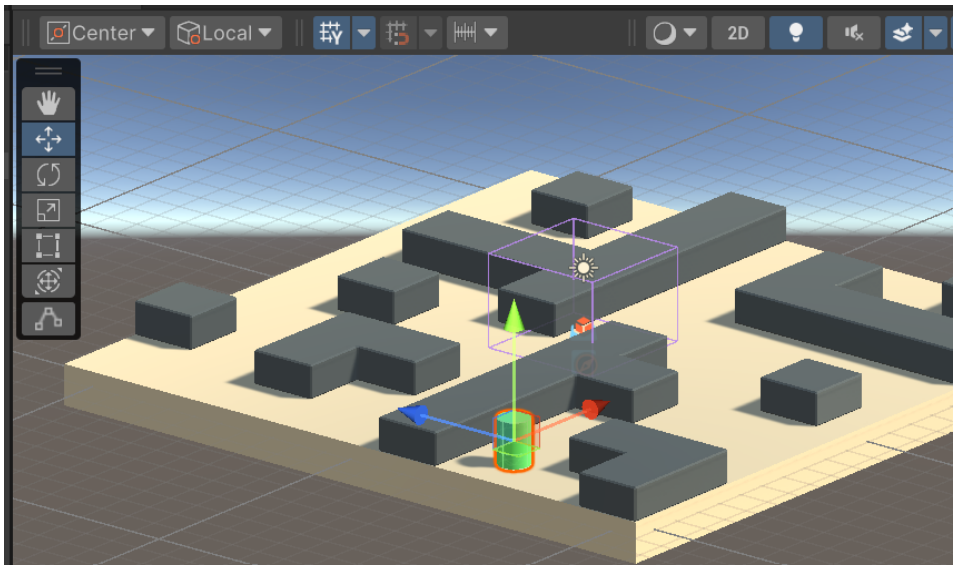

Figure 4.4.1.2 Humanoid agent in unity

From figure 4.4.1.2 we can visualize the static map or level also a cylindrical object or a humanoid who will be assigned task to navigate between various coordinates. This is done by a RayCaste feature of unity which fires 1000 of rays from the main camera which helps to detect walls or ground but it needs more computing power

4.1.2. Unreal Simulation

4.1.2.1. Working Principle :

At the heart of this project lies a complex system that seamlessly combines state-of-the-art technologies. The system utilizes GPS coordinates to determine the desired destination, processing this information through a flight controller. The drone's navigation system employs various sensors, such as LiDAR and cameras, to scan and map the surrounding environment in real-time. Through the fusion of sensor data and GPS coordinates, the drone creates a dynamic path, which is continually adjusted to avoid obstacles. This dynamic path planning is executed by an onboard computer running sophisticated algorithms, ensuring safe and efficient navigation. The drone's actuators, including motors and propellers, work in tandem with the navigation system to control its movements, allowing it to reach the desired destination while avoiding collisions with any obstacles in its path.

4.1.2.2. Fundamentals and Core Components :

The project's core components revolve around the utilization of a simulation-based approach for autonomous drone navigation and obstacle avoidance. This approach ensures safe and efficient testing and development of the system without the need for physical hardware. At its foundation is the AirSim platform, a robust simulator that emulates real-world flying conditions, allowing the drone to interact with a dynamic landscape realistically.

For navigation and obstacle avoidance, the project employs the A* (A-star) algorithm, a popular and effective pathfinding technique. A* is used to determine the optimal path from the drone's current location to a specified destination while taking into account the landscape and any potential obstacles. This algorithm intelligently evaluates possible paths, considering factors like terrain elevation, distance, and any obstructions, to chart a safe course for the drone. The A* algorithm ensures efficient, real-time path planning, enabling the drone to navigate complex landscapes and reach its destination

while avoiding collisions with simulated obstacles.

By combining the power of simulation with the simplicity and effectiveness of the A* algorithm, this project provides a solid foundation for autonomous drone navigation and obstacle avoidance, making it a cost-effective and reliable solution for various applications, from aerial surveys to environmental monitoring.

4.2. Results

4.2.1 Unity Simulation

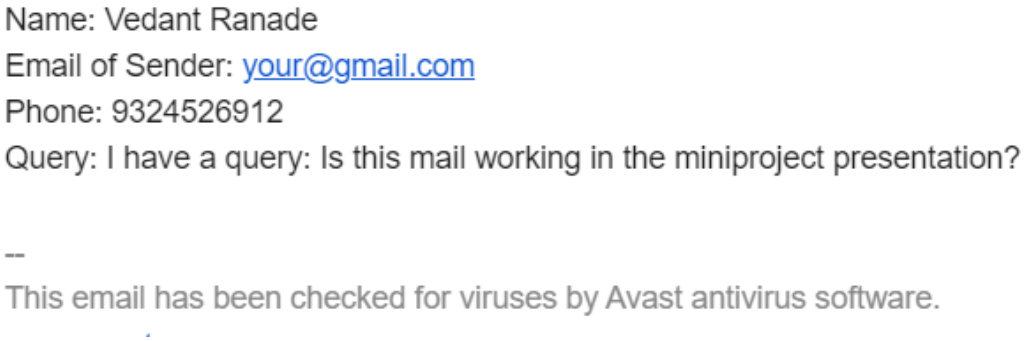

Figure 4.2.1.1 Email received through Contact .

From Figure 4.2.1.1. The mail was sent by the python program using SMTP library which enables the application to deliver email. This feature is used in to Contact the vendor or owner if user has any query .

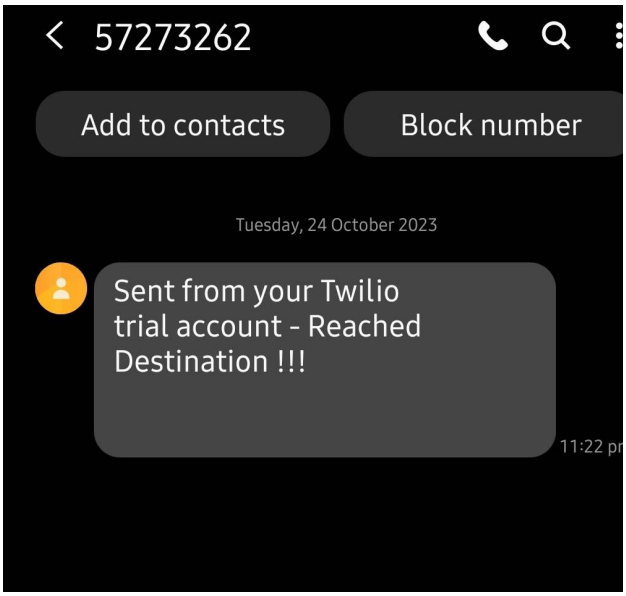

Figure 4.2.1.2 SMS received through API .

From Figure 4.2.1.2 we can see that the SMS was sent by the python program using Twilio API which enables the application to send notifications on users mobile number.

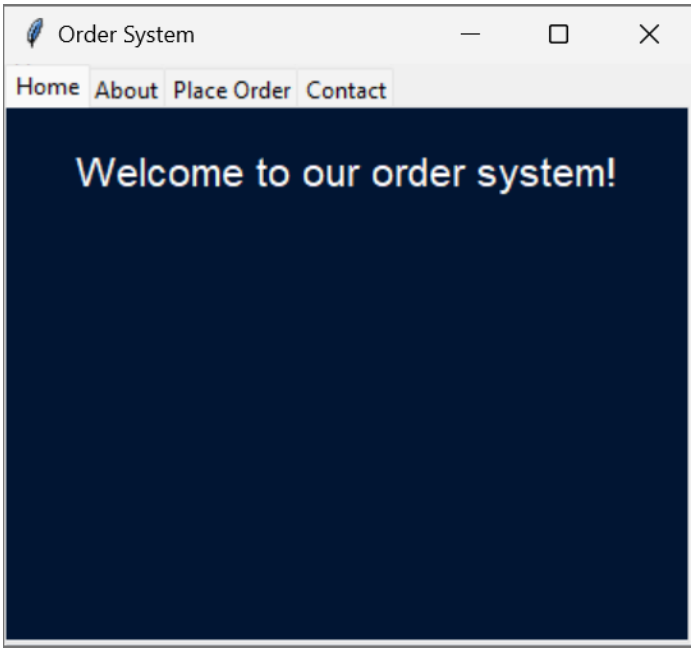

Figure 4.2.1.3. Gui using Tkinter

Figure 4.2.1.3 is a snapshot of our landing window in GUI built on python using Tkinter library which is most popular for building highly interactive graphical user interfaces for application whose backend is in python.

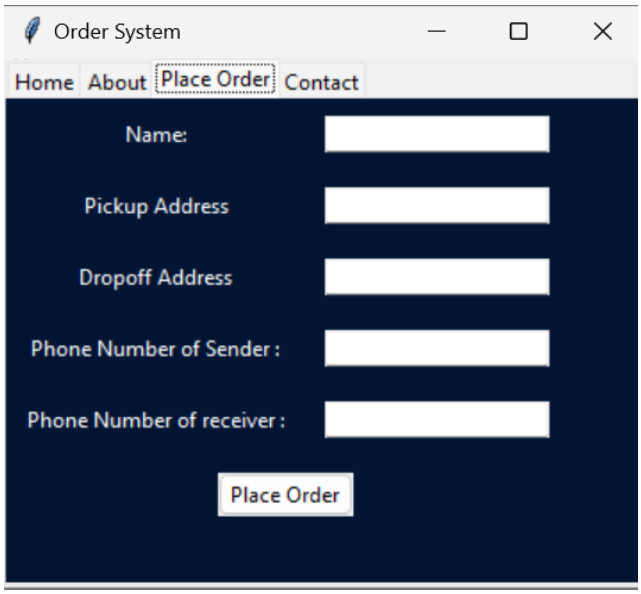

Figure 4.2.1.4. Place order GUI

Figure 4.2.1.4 is a snapshot of our PlaceOrder Tab in GUI built on python using Tkinter library ,This is routed through navigation where the user can enter various details and these details will be verified and then considering these inputs the agent will start executing

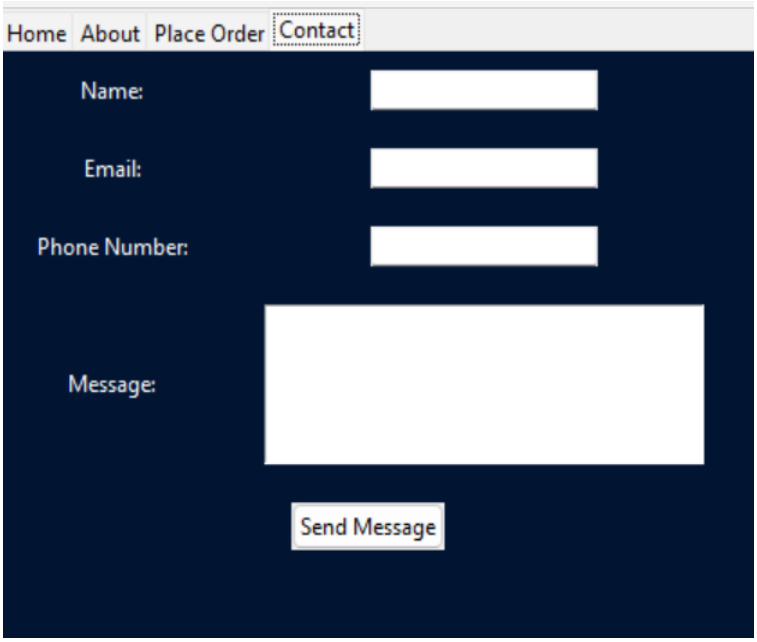

Figure 4.4.1.5 Place order GUI

Figure 4.4.1.5 is a snapshot of the Contact Tab in GUI built on python using Tkinter library ,This is routed through navigation where the user can enter various details if the user wishes to contact then

the mail is sent using SMTP library and it is secure as users credentials of mail are not required in this action

4.2.2 Unreal Simulation

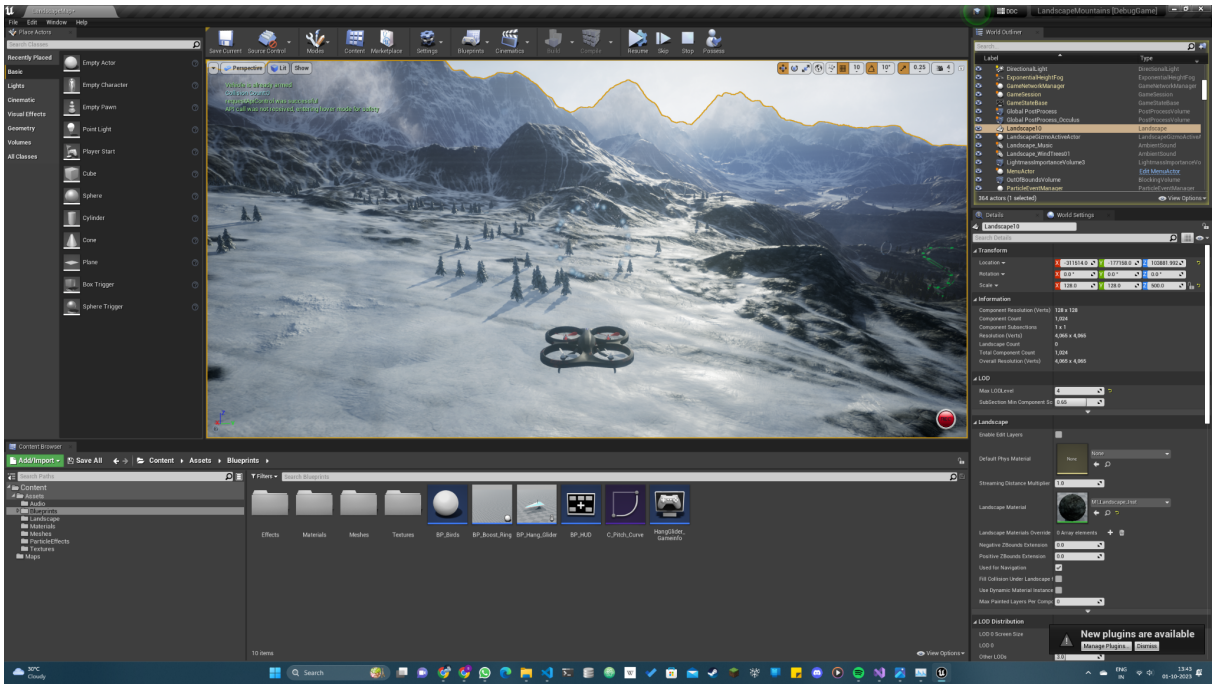

Figure 4.2.2.1 Simulation created in unreal.

Figure 4.2.2.1 is a snapshot of the simulation in unreal engine . Unreal engine has a huge array of options for environments to choose from , here the environment which is simulated is an icy mountainous landscape and the agent is a drone which navigates through the environment , This was chose to shown the navigation by drone , to hard to reach places ,

5. Conclusion and Future Scope

5.1. Conclusion

In this project , the research, comparison, and contrast of various pathfinding algorithms, including Depth First Search, Best First Search, Dijkstra, and A* (A star). Given the heuristic nature of the A* algorithm and it's from one point to one point .

Firstly, visualizing the A* algorithm in its basic form by simulating it within a 10x10x10 grid. This allowed us to observe the highlighted path generated by the algorithm. Building upon this initial exploration, we decided to develop and simulate the A* algorithm using Unity software. While the Unity project performed adequately, limitations arose due to constraints such as the quality of environments and the resource utilization of computer components, including ROM and RAM.

Consequently, explored other similar environment simulation and game engines. Unreal engine offers a lightweight platform with high-quality maps and environments to facilitate a more robust and immersive simulation of the algorithms.

5.2. Future Scope

The future endeavors aim to enhance the project's capabilities by implementing a comprehensive graphical user interface (GUI). This GUI will enable users to interact with the project in a user-friendly manner, providing notifications for various events such as reaching the goal state or alerting users in the event of errors.

Furthermore, the plans to extend the research to explore additional pathfinding algorithms, including the ant colony and bee hive algorithms, alongside advanced variations of the A* algorithm. The objective is to conduct comprehensive performance assessments of these algorithms, taking into account various constraints and conditions. This comparative analysis will help identify the fastest and most effective algorithm for each unique scenario.

In addition, the integration of machine learning (ML) into the project. By harnessing the power of ML, the enhancement of the adaptability and decision-making capabilities of the navigation

system, enabling it to learn and optimize its performance based on real-time data and experiences.

These future prospects represent the commitment to advancing and expanding the project's scope, making it a dynamic and evolving solution for a wide range of applications and challenges.

References:

1. https://carla.org/
2. https://arxiv.org/ftp/arxiv/papers/2110/2110.00910.pdf
3. https://medium.com/velotio-perspectives/exploring-openai-gym-a-platform-for-reinforcement-learning-algorithms-380beef446d
4. https://paperswithcode.com/method/carla
5. https://www.sciencedirect.com/science/article/pii/S1877050921025552
6. https://arxiv.org/pdf/1606.01540.pdf
7. https://cmudeeprl.github.io/403_website/assets/lectures/s21/s21_rec1_gym.pdf
8. https://docs.unity3d.com/Manual/GettingStartedWithPythonInUnity.html
9. https://docs.unrealengine.com/4.27/en-US/Python
10. https://docs.python.org/3/library/socket.html