TEAM I - CS 4091

Boeing Project #3: Air-to-Ground Search

Requirements Analysis

Will Weidler
Bailey Schoenike
Brennen Crawford
Donovan Bale
Lucas Wiley

1. Project Foundation

Project Overview

1. Project Title
    a. Optimized Air-to-Ground Grid Search: Heuristic Strategies for Efficient Coverage
2. Members and Roles
    a. Will Weidler
    b. Brennen Crawford
    c. Bailey Schoenike
    d. Donovan Bale
    e. Lucas Wiley
3. Project's primary goal and objectives
    a. What is the main problem you're solving?
        i. How can we optimize a search pattern within a time constraint?
    b. What is the end result you want to achieve?
        i. Develop path planning system / backend based on generated inputs
        ii. Develop multiple competing heuristics and/or algorithms
            1. Analyze which works better and why
        iii. Implement coverage algorithms for path planning system
        iv. Create obstacle avoidance handler for path planning system
        v. Implement frontend framework that facilitates control of all essential systems
4. Project scope
    a. Which features will you implement?
        i. The project will have the feature to generate a 2D grid world only
            1. The planning system will use this world to navigate around
        ii. The project will have fixed movement patterns and static obstacles
    b. What functionality is essential?
        i. The project will include a working path planning system with research to back up the system chosen
    c. What data will you handle?
        i. The project will parse the data in the randomly generated world to correctly label obstacles and find a clear path through it
    d. Which features are out of scope?
        i. This project will not focus on 3D mapping environments
    e. What won't your system handle?
        i. This project will not need to support multiple users

2. Understanding Your Users and Stakeholders

User Analysis

1. Who are your primary users?
    i. Aircraft pilots
    ii. Defense contractors
2. What problems are you solving for them?
    i. Surveying terrain in unfamiliar territories
        1. For search-and-rescue
        2. For geological surveys
        3. For infrastructure damage assessment
3. What are their technical capabilities?
    i. Piloting the aircraft
    ii. Understanding how far ahead the sensor can detect
    iii. Providing information for troops on the ground to prepare for combat/surveying
4. What are their key needs and pain points?
    i. Needs
        1. Pilots will want to survey the land with minimal input for the sensor
            a. Could prove distracting to operator
        2. Feedback from the sensor
    ii. Pain points
        1. Confusing display of the terrain on map
        2. Slow performance from the search due to many possible paths

Stakeholder Analysis

1. List all project stakeholders (professors, clients, end-users)
    i. Boeing executives
    ii. Boeing investors
2. Document their expectations and requirements
    i. They expect geographical information to be easily accessible and readable
    ii. Boeing executives would want project code integrable into future projects
    iii. Their requirements would be a reasonable project timeline
3. Note any conflicts between different stakeholder needs
    i. Boeing executives would want geographic information presented in a clear and concise manner, but this would most likely require additional time on the project, going against Boeing investors expectations.

3. Functional Requirements

Feature ID: F001

Name: Path Planning Algorithm Selection

Description: The system should provide a selection of path planning algorithms for the user to choose from, including A*, Dijkstra's, and potentially others.

Priority: High

User Story: "As a developer, I want to be able to select different path planning algorithms so that I can compare their performance and identify the most suitable one for my specific needs."

Acceptance Criteria:

The system displays a list of available path planning algorithms.

The user can select a specific algorithm from the list.

The selected algorithm is used for path planning in the generated world.

Feature ID: F002

Name: Visualize Path Planning Results

Description: The system should visually display the calculated path on the generated grid world, highlighting the path taken by the planning algorithm.

Priority: High

User Story: "As a developer, I want to be able to visualize the path generated by the chosen algorithm so that I can easily understand its performance and identify any potential issues."

Acceptance Criteria:

The generated path is clearly displayed on the grid world.

The path is visually distinct from the obstacles.

The system provides information about the path length and time taken to calculate the path.

Feature ID: F003

Name: Obstacle Avoidance Handler

Description: The system should implement an obstacle avoidance handler that ensures the planned path avoids all obstacles in the grid world.

Priority: High

User Story: "As a developer, I want to ensure that the path planning system avoids all obstacles in the generated world so that the planned path is safe and feasible."

Acceptance Criteria:

The calculated path does not intersect with any obstacles.

The system handles situations where the path is blocked by obstacles and finds alternative routes.

Feature ID: F004

Name: Performance Metrics Display

Description: The system should display key performance metrics for each path planning algorithm, such as path length, time taken to calculate the path, and potentially other relevant metrics.

Priority: Medium

User Story: "As a developer, I want to see performance metrics for each algorithm so that I can compare their efficiency and effectiveness."

Acceptance Criteria:

The system displays path length and time taken for each algorithm.

The system provides the option to display additional metrics, such as the number of nodes explored or the average path cost.

Feature ID: F005

Name: User-Friendly Interface

Description: The system should have a user-friendly interface that allows for easy navigation and control of the path planning system.

Priority: Medium

User Story: "As a user, I want to be able to easily interact with the system and control the path planning process."

Acceptance Criteria:

The interface is intuitive and easy to use.

The user can easily generate new worlds, select algorithms, and view results.

The system provides clear feedback to the user during the path planning process.

System Behaviors

1. Describe how your system responds to user actions
   a. The system selects and enacts a path planning algorithm based on the input of the user, and shows the planned path through the gridworld from this algorithm.
2. Define input requirements and expected outputs
   a. Inputs will be in the form of a .csv file detailing the gridworld, which can be generated by a pre-existing python script. The output will be a generated path to the goal destination, displayed on screen. The user will also have to pick a path planning algorithm to use.
3. Specify error handling procedures
   a. If the system fails to find a path for the given algorithm, a message saying that no such path can be found will be displayed. This is possible if the goal destination is surrounded by obstacles.

4. Technical Requirements

Development Specifications

1. Programming languages and frameworks
   a. C++ and Python for the back end
   b. Blazor for the front end
2. Database requirements
   a. Not needed
3. API integrations needed
   a. Blazor Server has integrations built-in for interacting with backend
4. Development tools and environments
   a. Visual Studio Code
   b. Docker

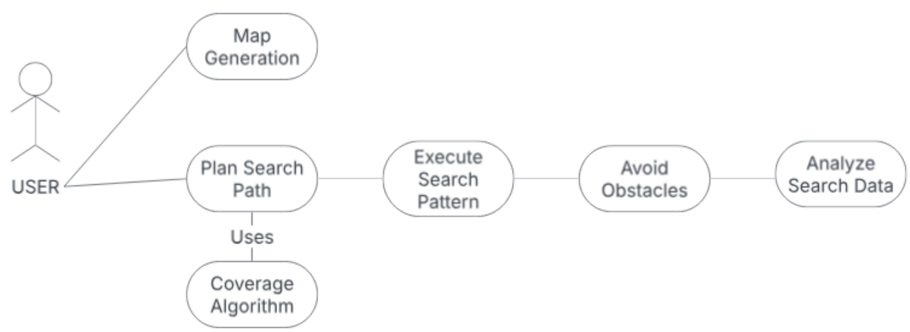
Quality Requirements

1. Performance expectations (response times, load capacity)
   a. Time needed for processing
      i. Map generation
      ii. Path finding
   b. Hosted locally on user's system

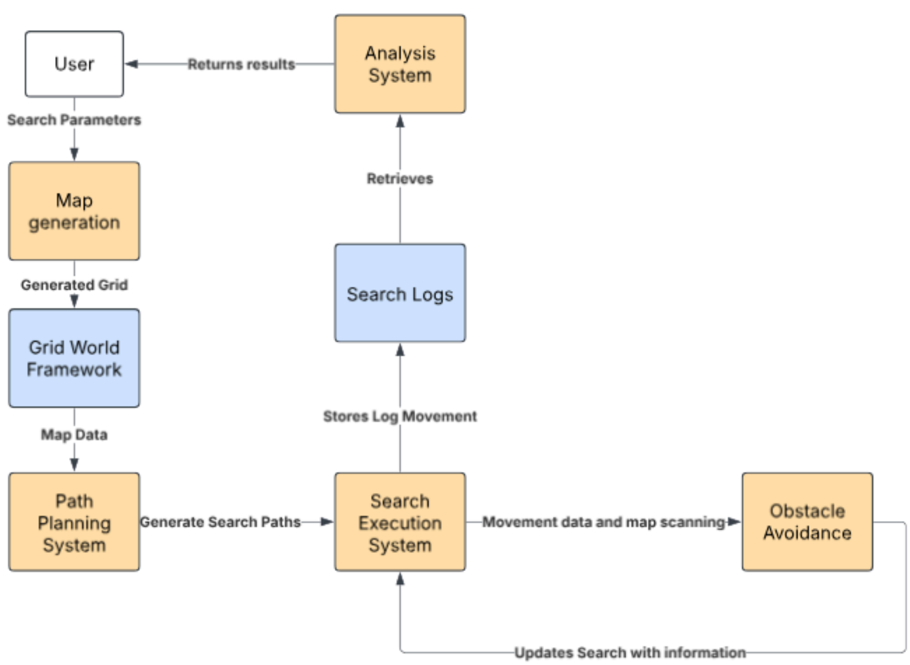
2. Security requirements
    a. Security requirements are not necessary for this project
        i. All maps will be generated locally or imported by user
3. Reliability standards
    a. Must generate the same path given the same input
4. Compatibility requirements
    a. Blazor will provide compatibility with web browsers

5. Documentation and Diagrams

Use Case Diagram



Data Flow Diagram



Entity-Relationship Diagram: Not Applicable - No database

6. Project Planning Timeline

1. Sprint 1: Weeks 1 - 4
    a. Project overview and task assignment
        i. Scope Definition
        ii. Requirements Analysis
        iii. Project Plan
2. Sprint 2: Weeks 5 - 6
    a. Starting frontend
        i. Blazor Server setup
        ii. Development container created
    b. Ensuring frontend works as expected
        i. Testing edge cases
        ii. Outside user testing
3. Sprint 3: Weeks 7 - 8
    a. Integrating frontend with backend
        i. Prototype 1 finished
4. Sprint 4: Weeks 9 - 10
    a. Functioning base algorithm - A*
        i. Development container created
        ii. Prototype 2 finished
            1. Focused on backend research and development
5. Sprint 5: Weeks 11 - 12
    a. Implementing second algorithm - Djikstra's
        i. Ensuring functionality with frontend and backend
6. Sprint 6: Weeks 13 - 14
    a. Final refinement
        i. Bug fixing and last needed implementation

7. Validation Strategy

Review Process

1. Self-review the documentation
    a. During sprint reviews
2. Peer review by team members
    a. During sprint review
3. Professor/advisor review
    a. During office hours
4. Stakeholder review
    a. Charles Tullock

5. Final adjustments based on feedback
   a. Before final presentation