Group O Project Capstone Scope

Air-To-Ground Search

Team Members: Jimmy Hoerschgen, Matthew Lopez, Ryan Kluesner, Ethan Heckart, Jack Wyeth

Project Goals

1. State your primary objective
    - What is the main problem you're solving?
        1. We are designing a program to simulate air-to-ground search over an arbitrarily sized grid containing obstacles.
    - What is the end result you want to achieve?
        1. Design a/n algorithm(s) capable of scanning a large number of cells in a grid in a minimal number of movements while avoiding obstacles.
2. List specific goals
    - Make them measurable and achievable
    - Use action verbs (develop, create, implement, etc.)
    - Keep them focused on key deliverables
        1. Develop Algorithms able to scan at least 80% of cells in a grid.
        2. [*Quantify minimum number of moves*]
            a. (Search 80% of spaces while visiting <50% of spaces?)
            b. (Search 80% of spaces while visiting n/6 spaces, where n is the number of spaces not occupied by an obstacle?)
        3. The algorithm can be combined with other planes in order to search the grid more efficiently.
            a. This could either mean routinely searching >80% of spaces or achieving 80% coverage in roughly half as many moves
        4. Implement automatic search area generation including randomly placed obstacles, or allow the user to manually define the search area.
        5. The program is capable of visualizing the search space including the plane's location, searched cells, unsearched cells, and obstacles.

Project Boundaries

1. Identify what's included
    - Which features will you implement?
        1. Interactive GUI
            a. For selecting different algorithms, number of planes
        2. Multiple algorithms for pathfinding
        3. Method for generating maps (including obstacles)
        4. Iterative testing
        5. A method for the plane to interact with the user
        6. While the airplane is determining its next position the plane will avoid spaces where it can get stuck.
            a. Avoid box canyons.
        7. The airplane shall attempt to start and stop on the same square.
        8. While the airplane is moving it shall avoid obstacles.
        9. When the airplane moves the GUI shall update to reflect its current position.
    - What functionality is essential?
        1. Multiple algorithms for pathfinding
        2. Visual
        3. Testing
    - What data will you handle?
        1. The time it takes for the algorithm to find the optimal flight path
        2. The coordinates as the plane maneuvers through the map

2. Specify what's excluded
    - Which features are out of scope?
    - What won't your system handle?
    - What are the limitations?
        1. The bonus feature of moving obstacles is something the team would want to implement, but at this starting point for the project it is not a feature we are expecting to implement.
        2. A database
        3. The airplane will not be controlled by user input.
        4. The project will not involve a real, physical airplane.

Required Resources

1. List technical resources
    - Program languages used:
        1. Python
        2. C++
    - Development tools
        1. Anaconda
        2. Mypy
        3. Black
        4. Cppcheck
        5. Clang-format
    - Testing frameworks
        1. Gtest
        2. unittest
    - Version control systems
        1. Git
        2. Github
2. Identify data requirements
    - Input data formats
        1. A method for loading pre-made maps from a configuration file. (TOML, JSON, XML, …)
    - Test data needs
        1. Test the movements of the airplane
        2. Test specific behaviors of the airplane
            a. E.g. avoiding canyons where it can get stuck
        3. Testing the GUI
    - Storage requirements
        1. A means by which to store the pathfinding data and then select the most optimal path amongst the dataset
3. Specify computing needs
    - Hardware requirements
        1. Windows 7 or better, 4+ GB RAM, NVidia 4090 GPU
    - Software licenses
        1. 3-Clause BSD License
    - Cloud services (if needed)
        1. Hopefully none

Aircraft Routing Project Scope Example

Project Goals:

- Develop a route optimization system
- Implement cost calculation algorithms
- Create input/output processing system

Project Boundaries:

- Limited to static location data
- Fixed cost factors only
- Single route calculation at a time

Required Resources:

- Development environment
- Algorithm testing framework
- Location and cost data sets
- Computing resources for calculations

Important Notes

1. Keep goals SMART:
    - Specific
    - Measurable
    - Achievable
    - Relevant
    - Time-bound
2. Make boundaries clear:
    - Be explicit about limits
    - State what's excluded
    - Define constraints clearly
3. Be realistic about resources:
    - List all essential tools
    - Include testing needs
    - Consider development environment