perplexity

can you fine tune the project?

Of course. Let's fine-tune the **Motion Planning with Graph of Convex Sets (GCS)** project to make it more manageable for your team, given your experience level. The goal is to reduce the initial difficulty from a "7" down to a more approachable "5" or "6" by strategically simplifying the scope while preserving the core learning objectives.

This fine-tuned plan focuses on achieving a solid, demonstrable result within one month without getting bogged down in the most complex aspects of the problem.

Fine-Tuned One-Month Project Plan: A Practical GCS Planner

Week 1: Simplified Foundations & Setup

Objective: Get a visual environment running quickly with minimal theoretical overhead.

- **Task 1: Pain-Free Environment Setup**
  - **Actionable Step:** Instead of wrestling with commercial solver licenses, install `CVXPY` with a bundled open-source solver that supports mixed-integer problems, such as `ECOS_BB` or `CBC`. This avoids administrative hurdles.
- **Task 2: Focus on Intuition, Not Just Dense Theory**
  - **Actionable Step:** Watch introductory video lectures on convex optimization and motion planning *before* diving deep into research papers. Focus on understanding the high-level concepts first: "What is a convex set?" and "Why is convex optimization useful?"
- **Task 3: Create a "Toy" World**
  - **Actionable Step:** Manually define a *very* simple 2D environment. Use `matplotlib` to draw:
    - A start point and a goal point.
    - One or two large, simple, convex polygon obstacles (e.g., squares or triangles).
    - Manually draw 3-4 large, overlapping convex polygons that cover the "safe" path around the obstacles. This will be your graph.
- **Task 4: Basic Visualization**
  - **Actionable Step:** Write a simple Python script that just visualizes your "toy" world: the obstacles, the start/goal points, and the hand-drawn convex regions.

Week 2: Core Solver Implementation (Simplified)

Objective: Get the optimizer working on the simplest possible version of the problem.

- **Task 1: Formulate the "Barebones" MICP**
  - **Actionable Step:** In `CVXPY`, formulate the most basic version of the GCS problem.
    - **Cost Function:** Start by simply minimizing the path length (a straightforward quadratic cost).
    - **Constraints:** Implement only the most essential constraint: the trajectory must remain inside the chosen convex sets.
- **Task 2: Build Incrementally**
  - **Actionable Step:** Don't try to solve the full mixed-integer problem at once.
    1. First, write a program that finds the shortest path between two points *inside a single convex set* (a simple convex problem).
    2. Next, expand this to find a path through a *pre-defined sequence* of two convex sets.
    3. Finally, add the binary integer variables to let the solver *choose* the best sequence of sets. This incremental approach makes debugging much easier.
- **Task 3: Frequent Visualization**
  - **Actionable Step:** After every successful run of the solver, immediately plot the resulting path. Visual feedback is the fastest way to know if your formulation is correct.

Week 3: Controlled Expansion & Analysis

Objective: Add one layer of complexity to demonstrate deeper understanding and then analyze the results.

- **Task 1: Add a Single Dynamic Constraint**
  - **Actionable Step:** Introduce one simple dynamic constraint, such as a maximum velocity limit between points on your trajectory. This is a common and easy-to-implement linear constraint that adds a level of realism.
- **Task 2: Test on Different Scenarios**
  - **Actionable Step:** Create two more simple "toy" worlds. For example, one with a narrow passage and another where the shortest path is blocked. Run your planner on all three and save the resulting plots.
- **Task 3: Analyze Solver Performance (Qualitatively)**
  - **Actionable Step:** In your notes for the final report, document how the solver's runtime changes as you add more convex sets or constraints. You don't need a formal complexity analysis; just observing and reporting that "adding more sets increased the solution time" is a valuable insight.

Week 4: Report, Present, and Conclude

Objective: Deliver a polished and concise report and presentation that focuses on what you achieved.

- **Task 1: Simplify the Final Report**
  - **Actionable Step:** Structure your report to highlight your successes. The focus should be on:
    - A clear explanation of your simplified problem formulation.
    - Visual results from your three test scenarios.
    - The lessons you learned about applying convex optimization to a robotics problem.
- **Task 2: A Demo-Focused Presentation**
  - **Actionable Step:** Prepare a short presentation that heavily relies on visuals. The centerpiece should be a live demo (if possible, in a Jupyter Notebook) or a video showing your planner successfully finding paths in your different "toy" worlds.
- **Task 3: Discuss Limitations and Future Work**
  - **Actionable Step:** Conclude your report and presentation by acknowledging the simplifications you made (e.g., manual decomposition of free space) and suggesting how a future team could build on your work (e.g., by implementing automatic convex decomposition). This shows maturity and a deep understanding of the problem domain.