As per the course syllabus, 25% of your course grades are based on semester projects. Below are the project guidelines and deliverables.

Project Guidelines

The goal of the project is to build a motion planner for the Poppy Ergo Jr robotic arm. The motion planner will be presented with blocks of various shapes and sizes, and needs to plan trajectories that move and stack the blocks into various arrangements. Evaluation will include a simulation component (with PyBullet) and hardware component (with a physical Ergo Jr arm).

You can rely on the following **assumptions**:

- You will be provided with a simulation environment, hardware platform, and low-level control API that can command the arm's motor angles with position control. The API code will be available at: https://github.com/garrettkatz/s25rmp
- The initial and target arrangements of the blocks will be provided to your motion planner, so you do not need to handle visual perception. However, the robot does have a camera and you are allowed to use it to improve performance if you want.
- Some blocks may have more complex shapes than rectangular prisms. However, all shapes will be defined within a voxel grid with fixed grid spacing. In other words, each block can be viewed as a union of small axis-aligned cubes with one connected component. That said, the block as a whole may be oriented in a way that does not align with the coordinate axes.
- The simulation environment will include evaluation metrics and test cases you can use to measure your motion planner's performance.
- A single hardware assembly of the arm will be available for the class to share. Some lecture periods and other times will be available for teams to reserve access to the hardware and test their code.

Note the following **requirements**:

- Your planned approach must be approved by the instructor in the proposal phase. You can use inverse kinematics, sampling-based planners, machine learning, a combination, or other methods. However, the instructor will not re-run any training procedure, so if you use machine learning, you must include pre-trained model weights in your submission.
- Your code, including any dependencies and environment configuration, must be easy to run. If the instructor has difficulty running your code you may receive zero on the coding part of the grade. Any more complicated languages/dependencies than Python and the SciPy/PyTorch stacks must be approved by the instructor in the proposal phase.
- This is a group project; you can form your own teams. Each team must have between 1 and 4 members. The entire team will receive a single grade, so by forming a team, you accept the risk that some teammates may contribute more than others for the same grade.
- You must host your code on GitHub and submit a report including the GitHub link.
- The report must use the IEEE conference paper template (used at IROS).

Project Proposal

Due date: Sunday March 30, 11:59pm

Project ideas must be proposed to and approved by the instructor, possibly subject to modifications. Your proposal can be brief (less than one page, bullet points are fine) and only one team member needs to submit. The proposal must contain the following information:

- What are the names and SU emails of all team members in the group?
- Do you want to be in a group but can't find any teammates?
- Are you a small group and open to additional teammates?
- Which motion planning approach, language and libraries will you use? Inverse kinematics, sampling-based, machine learning, something else? Include citations for specific methods and libraries you will use. If your approach is non-standard, provide additional explanation of your proposed architecture.

Proposals do not receive a separate grade. However, failure to submit a proposal may result in a zero grade on your entire project, regardless of your final report quality.

Project Reports

Due date: Tuesday May 6, 11:59pm

The report should be 4 pages and use the IEEE conference LaTeX template. Use the four pages as effectively as possible: Your report should be neither longer nor substantially shorter than 4 pages. It should have the following structure:

- Frontmatter
  - Title
  - Names and SU emails of all team members
  - Brief abstract summarizing the report
- Introduction
  - Motivate your project. Described at a conceptual level, which particular approaches did you use and why?
  - Summarize related literature on the approaches you used including in-text citations.
  - Include a link to a GitHub repository with README explanation and all code needed to reproduce your results.
- Methods
  - Provide more technical detail on your approach. Include a figure diagram and important equations to describe your motion planner. Relevant details may include: What are the objective and constraint functions? How did you optimize them? What is the motion planner's top-level algorithm pseudocode? What code libraries did you use (with in-text citation)? Etc.

- Results
  - **Simulation Results:** Use 30 independent validation trials generated by the simulation environment to measure the performance of your motion planner. Plot the validation metrics returned by the simulation environment in an appropriate format, such as histograms or violin plots.

    Next, if you use machine learning, plot the learning curves. Otherwise, plot metrics returned by the optimization tool you used during the validation trials (solver iterations, function and Jacobian evaluations, etc.).

    Finally, form a hypothesis as to why your method fails in certain cases. Describe those failure cases textually and using screenshots captured from the simulator. Design, collect, and plot an additional, more informative performance metric that tests your hypothesis.

    Your figures should be concise and well-organized; they should not take up multiple pages without any text and unnecessary whitespace.
  - **Hardware Validation:** Run 5 independent validation trials on the hardware platform. Manually record the overall success rate and fraction of per-block goal positions that was achieved at the end of each trial. Present these quantitative results and qualitatively describe the performance.
- Conclusion
  - Summarize your approach and results. Discuss any differences you observed between simulation and hardware, and hypothesize the main sources of the differences.
  - What were the main challenges in your project, and if you continued working on this project, what would be your next steps to improve it?
- References
  - Include a bibliography with consistent and standard formatting such as MLA or APA.
  - Any reasonable discussion of background work should result in at least 5 bibliography entries.
  - All factual claims made in your paper that are not supported by your own experiments should have corresponding references with in-text citations.

Rubric

Your project grade will not be based on the performance of your approach, but it will be graded on the quality of your experimental design, your code, and your report. Quality includes style, citation format, spelling and grammar, clarity, and thoroughness covering all points above. Each team member will receive the same grade. Specific percentage weights are as follows:

- Link and code runnability: 20%
- Completeness of requested report information (Intro motivation/background, details in method section, questions for conclusion section): 30%
- Experimental results: 35%

- Proper bibliography and in-text citations: 5%
- Style (spelling/grammar, text/figure formatting, conference paper template): 5%
- Novelty*: 5%

*The "novelty" criterion means that you did not simply clone another paper's repository and run their code, but you tried implementing some aspects of the motion planning yourself, using self-designed algorithms and/or formulations of the optimization problems involved. Make sure you clearly explain any novel changes you made in the methods section.