Development Plan
ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

[Put your introductory blurb here. Often the blurb is a brief roadmap of what is contained in the report. —SS]

[Additional information on the development plan can be found in the lecture slides. —SS]

1 Confidential Information?

[State whether your project has confidential information from industry, or not. If there is confidential information, point to the agreement you have in place. —SS]

[For most teams this section will just state that there is no confidential information to protect. —SS]

2 IP to Protect

[State whether there is IP to protect. If there is, point to the agreement. All students who are working on a project that requires an IP agreement are also required to sign the "Intellectual Property Guide Acknowledgement." —SS]

3 Copyright License

[What copyright license is your team adopting. Point to the license in your repo. —SS]

4 Team Meeting Plan

[How often will you meet? where? —SS]

[If the meeting is a physical location (not virtual), out of an abundance of caution for safety reasons you shouldn't put the location online —SS]

[How often will you meet with your industry advisor? when? where? —SS]

[Will meetings be virtual? At least some meetings should likely be in-person. —SS]

[How will the meetings be structured? There should be a chair for all meetings. There should be an agenda for all meetings. —SS]

5 Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]

6 Team Member Roles

7 Workflow Plan

- How will you be using git, including branches, pull request, etc.?

- How will you be managing issues, including template issues, issue classification, etc.?

- Use of CI/CD

8 Project Decomposition and Scheduling

8.1 Project Management

The Kanban board template through Github Projects will be used to organize and monitor existing issues. The GitHub project can be found through the TPG project's Github repository. This board will contain the opened issues that can be moved across different stages of the development lifecycle such as In Progress, In Review and Closed.

8.2 Project Decomposition

The team will be using Github Milestones to group issues based on their common goal to be used as feedback regarding future improvements. This can be used to keep track of all the important deadlines in the course outline and their deadlines. As far as the project goes, it can be decomposed through several milestones based on the major phases of the development, for example Introduction and Research, and Development of CI/CD pipeline. The team must fully understand each large task or milestone to properly break them into smaller, manageable tasks and to set reasonable deadlines. For example, a large task can be decompose into:

- Planning and Research - gathering of information and identifying the scope

- Implementation - designing, execution of actions and coding

- Verification - testing and validating of the implementation

- Review and Feedback - peer, stakeholder and supervisor review

- Final Approval and Publication - publishing of approved implementation

By having this approach of task decomposition, the team can be flexible with assigning roles to better manage the workloads that the project requires. This method can be applied to every aspect of the project to ensure that large tasks are consistently broken down into smaller tasks.

8.3 Task Assignment

Each team member will have different strengths and weaknesses and interests that will be considered when assigning tasks. Issue size, which can be assigned within the Github Project, will be leveraged to ensure equal workload for each member. At the start of each milestone, the team must have a meeting to plan the tasks for upcoming weeks. Each task must be evaluated one-by-one to fairly estimate the effort required per task. These tasks must have a set buffer time in mind in the case of delays or unexpected challenges.

9 Proof of Concept Demonstration Plan

For the project's proof of concept demonstration plan, some significant risks may occur while in development and hinder the project's success. Here are the most significant risks:

- **GitHub / GitLab integration:** As of now, the TPG repository is currently on GitLab; however, GitHub is required to be utilized for this project. This TPG repository is still currently being contributed by both Dr. Kelly and graduate students. Failure to be able to integrate both changes could deal with some serious merge conflict issues.

- **Code Robustness:** Due to the large codebase and current unfamiliarity of the code, difficulties in testing may occur and the team may be unable to gather large code coverage within our testing suite. This will delay the goal of establishing a CI/CD environment for TPG and cause major hurdles in the remainder of the project.

- **Time-Consuming Refactoring:** A part of this project includes refactoring the code to follow standard software engineering principles, as this code is currently being modified throughout the project, refactoring may be a significant time-consuming task, preventing the completion of the other project goals.

- **Programming Language Learning Curve:** As most of the team for this project have not had experience with the C++ programming language, it will require some additional learning. This may slow down the progress of our project and prevent the proof of concept demonstration plan from being achieved.

- **Unable to create compatibility with the MuJoCo environment:** The project goal in the end is to allow the TPG code to be simulated

within the MujoCo environment. If we are somehow unable to do this, we may need to reroute the focus of our project.

If these significant risks can be prevented or overcome for this project, then the proof of concept demonstration shall include the following to demonstrate the overcoming of the risk:

1. The overcoming of TPG integrated into a MuJoCo simulation environment by a basic demonstration of CartPole.

2. Walkthrough the CI/CD pipeline of the TPG project.

10 Expected Technology

[What programming language or languages do you expect to use? What external libraries? What frameworks? What technologies. Are there major components of the implementation that you expect you will implement, despite the existence of libraries that provide the required functionality. For projects with machine learning, will you use pre-trained models, or be training your own model? —SS]

[The implementation decisions can, and likely will, change over the course of the project. The initial documentation should be written in an abstract way; it should be agnostic of the implementation choices, unless the implementation choices are project constraints. However, recording our initial thoughts on implementation helps understand the challenge level and feasibility of a project. It may also help with early identification of areas where project members will need to augment their training. —SS]

Topics to discuss include the following:

- Specific programming language

- Specific libraries

- Pre-trained models

- Specific linter tool (if appropriate)

- Specific unit testing framework

- Investigation of code coverage measuring tools

- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done

- Specific performance measuring tools (like Valgrind), if appropriate

- Tools you will likely be using?

[git, GitHub and GitHub projects should be part of your technology. —SS]

11 Coding Standard

[What coding standard will you adopt? —SS]

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. Why is it important to create a development plan prior to starting the project?

2. In your opinion, what are the advantages and disadvantages of using CI/CD?

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

[What are your team's external goals for this project? These are not the goals related to the functionality or quality fo the project. These are the goals on what the team wishes to achieve with the project. Potential goals are to win a prize at the Capstone EXPO, or to have something to talk about in interviews, or to get an A+, etc. —SS]

Attendance

Expectations

[What are your team's expectations regarding meeting attendance (being on time, leaving early, missing meetings, etc.)? —SS]

Acceptable Excuse

[What constitutes an acceptable excuse for missing a meeting or a deadline? What types of excuses will not be considered acceptable? —SS]

In Case of Emergency

[What process will team members follow if they have an emergency and cannot attend a team meeting or complete their individual work promised for a team deliverable? —SS]

Accountability and Teamwork

Quality

[What are your team's expectations regarding the quality of team members' preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team's expectations regarding team members' ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]