Team Contributions: POC
CVT Simulator

Team #17, Baja Dynamics
Grace McKenna
Travis Wing
Cameron Dunn
Kai Arseneau

This document summarizes the contributions of each team member up to the POC Demo. The time period of interest is the time between the beginning of the term and the POC demo.

1 Demo Plans

We will have the following components for our POC demo:

Unity FrontEnd

- A Unity frontend that will allow for an input to send to the backend.
- Will display a 3D model that will rotate based on the input.

Python Backend

- A python backend that will receive the input from the frontend and send back a response.
- The python backend will simulate a car without a transmission, simplfying our use case for the POC.

The demo will center around the communication protocol between the Unity frontend and the python backend. Demonstrating that the system can send information back and forth between these two components is the main focus.

2 Team Meeting Attendance

[For each team member how many team meetings have they attended over the time period of interest. This number should be determined from the meeting issues in the team's repo. The first entry in the table should be the total number of team meetings held by the team. —SS]

Student	Meetings
Total	8
Kai Arseneau	7
Travis Wing	8
Cameron Dunn	8
Grace McKenna	8

[If needed, an explanation for the counts can be provided here. —SS]

3 Supervisor/Stakeholder Meeting Attendance

[For each team member how many supervisor/stakeholder team meetings have they attended over the time period of interest. This number should be determined from the supervisor meeting issues in the team's repo. The first entry in the table should be the total number of supervisor and team meetings held by the team. If there is no supervisor, there will usually be meetings with stakeholders (potential users) that can serve a similar purpose. —SS]

Student	Meetings
Total	2
Kai Arseneau	2
Travis Wing	2
Cameron Dunn	2
Grace McKenna	2

[If needed, an explanation for the counts can be provided here. —SS]

4 Lecture Attendance

[For each team member how many lectures have they attended over the time period of interest. This number should be determined from the lecture issues in the team's repo. The first entry in the table should be the total number of lectures since the beginning of the term. —SS]

Student	Lectures
Total	12
Kai Arseneau	9
Travis Wing	9
Cameron Dunn	5
Grace McKenna	11

[If needed, an explanation for the lecture attendance can be provided here. —SS]

Only software lectures are considered.

5 TA Document Discussion Attendance

[For each team member how many of the informal document discussion meetings with the TA were attended over the time period of interest. —SS]

Student	Lectures
Total	3
Kai Arseneau	3
Travis Wing	3
Cameron Dunn	3
Grace McKenna	3

[If needed, an explanation for the attendance can be provided here. —SS]

6 Commits

[For each team member how many commits to the main branch have been made over the time period of interest. The total is the total number of commits for the entire team since the beginning of the term. The percentage is the percentage of the total commits made by each team member. —SS]

Student	Commits	Percent
Total	Num	100%
Kai Arseneau	37	%
Travis Wing	16	%
Cameron Dunn	18	%
Grace McKenna	15	%

For this table, as the repository was mirrored off of Dr. Smith's capTemplate repository, only commits after the mirroring on September 12th, 2024 are considered.

The number of commits alone may not accurately reflect each team member's contributions due to factors like our branching strategy, repository setup, and individual workflows. For instance, Kai Arseneau shows a high number of commits, but this is largely because he initially set up the repository, configuring CI/CD and issue/PR templates directly on the main branch. Many of these commits also include co-authors, leading to overlap. Additionally, the use of squash and merge can make other contributions seem smaller by comparison, even when they represent substantial portions of work. Given that only documentation has been added so far, tracking additions and deletions provides a more accurate view of each member's contributions at this stage.

Student	**Additions**	**Percent**
Total	20,987	100%
Kai Arseneau	5,521	26.31%
Travis Wing	5,171	24.73%
Cameron Dunn	5,190	24.64%
Grace McKenna	5,105	24.32%

Student	**Deletions**	**Percent**
Total	9763	100%
Kai Arseneau	2,710	27.76%
Travis Wing	2,351	24.08%
Cameron Dunn	2,367	24.24%
Grace McKenna	2,335	23.92%

[If needed, an explanation for the counts can be provided here. For instance, if a team member has more commits to unmerged branches, these numbers can be provided here. If multiple people contribute to a commit, git allows for multi-author commits. —SS]

7 Issue Tracker

[For each team member how many issues have they authored (including open and closed issues (O+C)) and how many have they been assigned (only counting closed issues (C only)) over the time period of interest. —SS]

Student	Authored (O+C)	Assigned (C only)
Kai Arseneau	31	7
Travis Wing	11	6
Cameron Dunn	1	5
Grace McKenna	1	5

[If needed, an explanation for the counts can be provided here. —SS]

8 CICD

We will utilize CICD to automatically lint, format and test our code.
For the python backend we will be using the following tools:

flake8 - A Python linter that checks for PEP8 compliance.

black - A Python code formatter that will ensure consistent code style.

unittest - Python's built-in testing framework that will be used for unit testing.

coverage - A testing framework for Python that will be used for code coverage.

For the Unity C# frontend we will be using the following tools:

SonarLint - C# linter that checks for code quality and security vulnerabilities.

StyleCop - C# linter that checks for code style and formatting.

UTF - A testing framework for C# that will be used for unit testing.

UTR - A testing framework for Unity that will be used for unit testing and code coverage.