# 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

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

## 3 Supervisor/Stakeholder Meeting Attendance

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

## 4 Lecture Attendance

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

Only software lectures are considered.

# 5 TA Document Discussion Attendance

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

# 6 Commits

| Student | Commits | Percent |
|---|---|---|
| Total | Num | 100% |
| Kai Arseneau | 38 | % |
| 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 | 21,301 | 100% |
| Kai Arseneau | 5,835 | 27.39% |
| Travis Wing | 5,171 | 24.28% |
| Cameron Dunn | 5,190 | 24.37% |
| Grace McKenna | 5,105 | 23.96% |

| Student | Deletions | Percent |
|---|---|---|
| Total | 9763 | 100% |
| Kai Arseneau | 2,725 | 27.87% |
| Travis Wing | 2,351 | 24.04% |
| Cameron Dunn | 2,367 | 24.21% |
| Grace McKenna | 2,335 | 23.88% |

# 7  Issue Tracker

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

# 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.