Team Contributions: Rev 0
CVT Simulator

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

This document summarizes the contributions of each team member for the Rev 0 Demo. The time period of interest is the time between the POC demo and the Rev 0 demo.

1 Demo Plans

[What will you be demonstrating —SS]

2 Team Meeting Attendance

Student	**Meetings**
Total	1
Grace McKenna	1
Travis Wing	1
Cameron Dunn	1
Kai Arseneau	1

3 Supervisor/Stakeholder Meeting Attendance

Student	**Meetings**
Total	0
Grace McKenna	0
Travis Wing	0
Cameron Dunn	0
Kai Arseneau	0

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

4 Lecture Attendance

Student	**Lectures**
Total	1
Grace McKenna	1
Travis Wing	0
Cameron Dunn	0
Kai Arseneau	0

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

5 TA Document Discussion Attendance

Student	**Lectures**
Total	1
Grace McKenna	1
Travis Wing	1
Cameron Dunn	1
Kai Arseneau	1

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

6 Commits

Student	Commits	Percent
Total	40	100%
Grace McKenna	8	20%
Travis Wing	10	25%
Cameron Dunn	8	20%
Kai Arseneau	14	35%

[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)
Grace McKenna	1	Num
Travis Wing	5	Num
Cameron Dunn	0	Num
Kai Arseneau	17	Num

[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.