Team Contributions: Rev 0
AutoVox

Team #10, Five of a Kind
Omar Abdelhamid
Daniel Maurer
Andrew Bovbel
Olivia Reich
Khalid Farag

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; the contributions prior to the PoC are NOT included.

1 Demo Plans

For our Rev 0 Demo, we aim to demonstrate the full functionality of our product, AutoVox. In addition to the polished functionality shown in the POC demo, the demo will focus on:

- **Refreshed UI:** We will showcase an improved UI that follows traditional principles of human-centred design, is organized logically and efficiently, and aligns with the non-functional requirements listed in the SRS.
- **Layer Editing Interface:** We will demonstrate the workflow of how voxels are edited within a layer of the voxel model, including layer selection, voxel selection, and property assignment.
- **Partitioning of models:** We will show how larger models are broken down into segments that are rendered separately, and swapping between them.

2 Team Meeting Attendance

Student	Meetings
Total	10
Omar	9
Daniel	9
Andrew	8
Olivia	10
Khalid	10

We try to schedule meetings so that all can attend; if one has missed meetings, it is either discussed prior the meeting, or the meeting was not relevant to all members.

3 Supervisor/Stakeholder Meeting Attendance

Supervisor's Name: [Dr. Onaizah]

Student	Meetings
Total	1
Omar	1
Daniel	1
Andrew	0
Olivia	0
Khalid	1

Due to scheduling issues, our team has not been able to meet with Dr. Onaizah as frequently as we would like. During the winter term, she has become quite busy. We have sent requests to meet throughout the month of January and have spoken to her in passing, but as of the submission of this report we have not received a response to confirm a meeting. We aim to address this issue as soon as possible, including potentially reaching out to one of her graduate students to speak to as a proxy.

4 Lecture Attendance

Student	Lectures
Total	2
Omar	0
Daniel	2
Andrew	0
Olivia	0
Khalid	0

We aim to always have one group member attend lectures in order to take notes, who is then responsible for updating the rest of the team.

5 TA Document Discussion Attendance

TA's Name: [Chris Schankula]

Student	Lectures
Total	2
Omar	2
Daniel	2
Andrew	2
Olivia	2
Khalid	2

6 Commits

Student	Commits	Percent
Total	118	100%
Omar	7	5.9%
Daniel	45	38.1%
Andrew	10	8.5%
Olivia	27	22.9%
Khalid	29	24.6%

Some members have lower commits due to commit sizes being larger and less frequent (e.g. only commiting once an entire assigned section is complete),

and some have more due to commits being smaller and more frequent (this is especially true for hot-fix commits).

7 Issue Tracker

Student	Authored (O+C)	Assigned (C only)
Omar	2	18
Daniel	74	32
Andrew	0	2
Olivia	16	16
Khalid	4	3

The large discrepancy in issue authorship is due to the assumed roles within our group; discrepancy in issue assignment/closed is due to group members forgetting to link issues until after the fact. Additionally, a set of issues closed dealt with small fixes from either peer review or TA feedback.

8 CICD

This section explains the Continuous Integration and Continuous Deployment (CI/CD) plan that has been utilized so far in our project.

8.1 Source Control and Branching Strategy

- **Repository Setup**: Our team has used GitHub for version control and collaboration.
- **Branching Strategy**:
  - `main`: The main branch used to store the production-ready code.
  - `<name of team member>-<issue name>`: The temporary branches created for each team member to work on their assigned issues. These branches are merged into the main branch after the task is completed.
- **Merging Policy**: All pull requests have at least two reviews before merging, as outlined in the Development Plan. This helps ensure that the code is of high quality and meets the requirements of the project.

8.2 Testing and Code Analysis Pipeline

Our team uses GitHub Actions for CI/CD to automate testing and code analysis on pull requests. They include the following:

- **Static Code Analysis & Linting**
  - **Frontend**: `ESLint` is used to enforce React coding standards and identify potential issues.
  - **Backend**: `PyLint` is used to handle both code smells for static analysis and enforce the coding standards for the backend.
- **Testing**
  - **Frontend Unit Tests**: Unit tests are written using `Jest` for React components and utility functions, the GitHub Actions run the tests using the command 'npm test'.
  - **Backend Unit Tests**: Unit tests are written using `pytest` for API endpoints and core functionality, the GitHub Actions run the tests using the command 'pytest'.
- **Documentation**: LaTeX documentation files are automatically compiled to PDF on commits to the `main` branch. The GitHub Actions run the compilation using the command 'make all'.

9 Team Charter Trigger Items

The triggers for team performance and accountability are as follows:

- **Meeting Frequency:** Team meetings must be held at least once per week, with all members expected to attend unless otherwise specified.
- **Supervisor Updates:** The supervisor must be updated at least once every two weeks via email, unless questions arise that require more frequent communication.
- **Code Review Requirements:** Each pull request must be reviewed by at least two other team members before being merged into the main branch.
- **Performance Metrics:** Team member contributions are tracked using commits, meetings attended, and issues completed to evaluate performance.
- **Underperformance Criteria:** A team member is considered underperforming if they are not contributing to the project or are not meeting deadlines, which triggers a team meeting to address the issue.

Our team has experienced no violations of these triggers in the team charter.

10 Additional Productivity Metrics

Our team has no other productivity metrics in use.