Team Contributions: POC
SFWRENG 4G06

Team 9, dice_devs
John Popovici
Nigel Moses
Naishan Guo
Hemraj Bhatt
Isaac Giles

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.

Numbers for all sections as of start of 2024-11-24.

1 Demo Plans

The team will demonstrate the following key components of the system during the POC demonstration.

1. **Game Setup and Customization:**

   - Demonstrate how users can set up a new local area network multi-player game.
   - Showcase customization of gameplay attributes, such as adjusting the number of dice and player health.
   - Through this, the modularity of the system will be displayed.

2. **Gameplay Mechanics:**

   - Conduct a walkthrough of a round of gameplay, highlighting how players reroll the dice and accumulate points.
   - Explain the scoring rules and demonstrate how player points are added.
   - This will showcase the basic game flow.

3. **Game State and Progression:**

   - Showcase how the game state is saved, i.e. how the game tracks and displays each player's current dice and hand points.
   - Demonstrate the endgame conditions, illustrating what happens when all rounds are played and points are tallied.
   - This will show how the system preserves game data.

4. **Multiplayer Mode:**

   - Demonstrate how players take turns simultaneously and show the tracking of player dice for both players.
   - This will highlight how data is synchronized between both players and how game integrity is preserved.

5. **Error Handling and Edge Cases:**

   - Showcase implemented safeguards, such as preventing invalid moves and handling unexpected inputs through demonstrating typical game actions.
   - Demonstrate the system's response in case of such scenarios.
   - This will showcase the rigidity and stability of the system.

2 Meeting and Lecture Attendance

Student	Team Meetings	Supervisor Meetings	Lectures	TA Meetings
Total	10	2	11	3
John P.	9	2	11	2
Nigel M.	10	2	8	3
Naishan G.	10	1	10	3
Isaac G.	9	1	7	3
Hemraj B.	5	1	6	3

There was one supervisor meeting at the beginning of the term before the whole team was together, and most of the communication was done through the team liaison, John, through email and thus there was no need for many additional meetings.
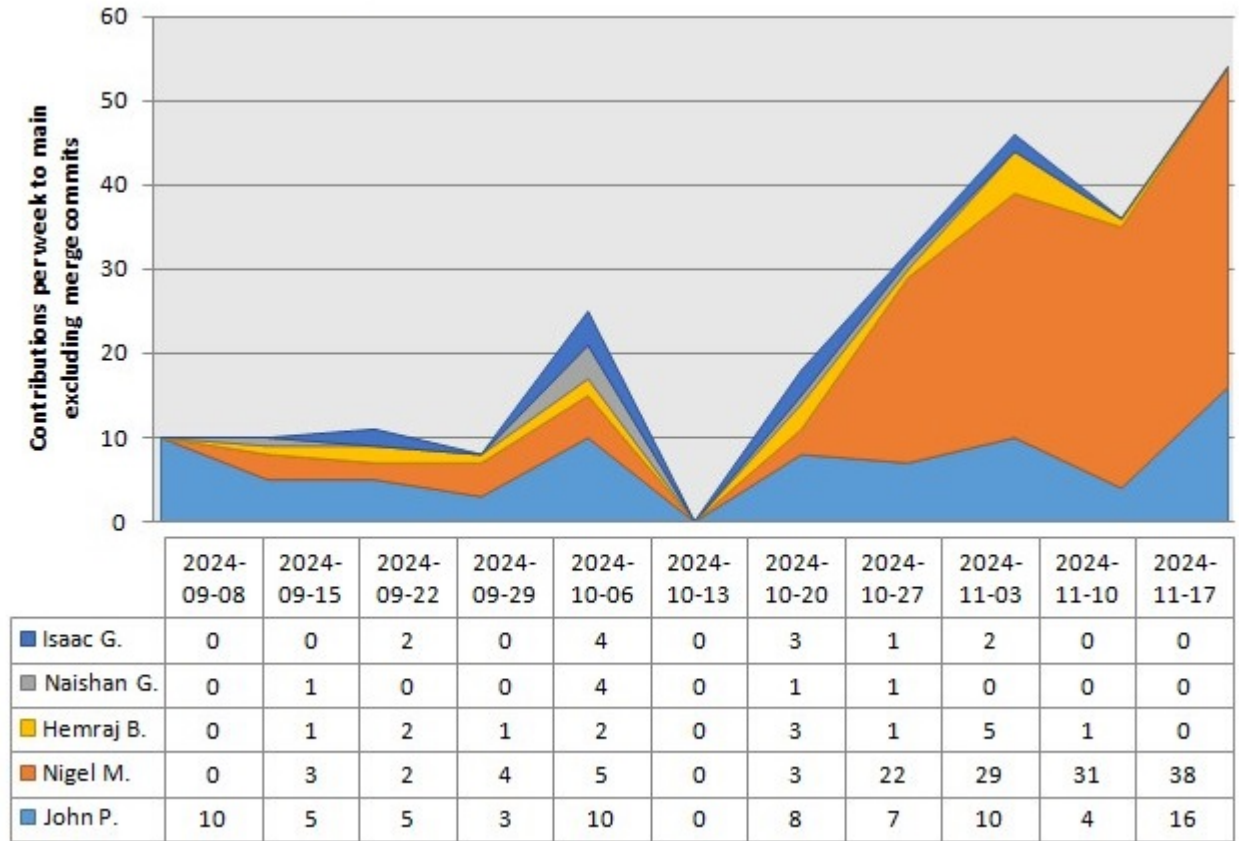
We always had one team member at every lecture and a majority at every TA discussion.

- "is:issue label:meeting"

3 Commits

Student	Commits	Percent
Total	250	100%
John P.	137	54.8%
Nigel M.	78	31.2%
Naishan G.	7	2.8%
Isaac G.	12	4.8%
Hemraj B.	16	6.4%

Figure 1: Contribution over time

	2024-09-08	2024-09-15	2024-09-22	2024-09-29	2024-10-06	2024-10-13	2024-10-20	2024-10-27	2024-11-03	2024-11-10	2024-11-17
Isaac G.	0	0	2	0	4	0	3	1	2	0	0
Naishan G.	0	1	0	0	4	0	1	1	0	0	0
Hemraj B.	0	1	2	1	2	0	3	1	5	1	0
Nigel M.	0	3	2	4	5	0	3	22	29	31	38
John P.	10	5	5	3	10	0	8	7	10	4	16

4   Issue Tracker

Total Authored refers to all internal issues authored including features, bugs, documentation, and team meetings/lectures, which are tracked as issues.
- "is:issue author:John-Popovici"

Assigned closed are those issues that have been closed by the specific student.
- "is:issue is:closed assignee:John-Popovici"

External Authored refers to issues that have been authored in external repositories for the purposes of peer reviews.
- found in https://github.com/simon-0215/UNO-Flip-3D/issues

In the first column, External as a student refers to those external to our group that have authored issues through peer review.
- "is:issue Peer Review in:title"

Student	Total Authored	Assigned Closed	External Authored
Total	105	50	21
External	15	-	-
John P.	53	28	21
Nigel M.	37	23	0
Naishan G.	0	0	0
Isaac G.	0	0	0
Hemraj B.	0	1	0

Some issues are closed and marked as having been worked on by two or more assignees, so the sum may not match up exactly to the total.

5 CICD

Given that our team is developing a Godot video game, the visual and interactive nature of our project makes full CI/CD integration somewhat challenging since there is a need for graphical testing and user input simulations, which standard pipelines don't readily support. Instead, we will focus on integrating Continuous Integration practices that focus on code consistency and collaboration.

Our team will use CI to automate build processes, to ensure that each new feature or fix is integrated into the latest codebase. And we can also implement CI to run unit tests on core game logic/backend systems, to ensure that critical functionalities work as expected with each update. This approach makes the most of the advantages of CI/CD while still most closely adhering to the needs and limitations of our specific project.