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 2024-11-06 18:00.

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 game, choosing between single-player and multiplayer (2-player) modes.
    - 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 roll the dice and accumulate points.
    - Explain the scoring rules and demonstrate how player health bars are affected.
    - 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 health and point totals.

- Demonstrate the endgame conditions, illustrating what happens when a player's health reaches zero, including win/loss indicators.
- This will show how the system preserves game data.

4. **Multiplayer Mode:**
    - Demonstrate how turns alternate between players and show the tracking of player health for both players.
    - This will highlight how data is synchronized between both players and how game integrity is preserved.
5. **Single-Player Mode:**
    - A stretch goal that will be demonstrated if developed in time for POC demonstration.
    - Demonstrate how the game plays when a user competes against a computer.
    - This will outline the computer's decision-making process, contrasting it with multiplayer mode.
6. **Error Handling and Edge Cases:**
    - Showcase implemented safeguards, such as preventing invalid moves and handling unexpected inputs.
    - 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	8	1	11	3
John P.	7	1	11	2
Nigel M.	8	1	8	3
Naishan G.	8	0	10	3
Isaac G.	7	0	7	3
Hemraj B.	4	1	6	3

There was only 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 any additional meetings. There is a meeting to be planned before the POC to show the project demo.

We always had one team member at every lecture and a majority at every TA discussion.

3 Commits

Student	Commits	Percent
Total	140	100%
John P.	57	40.71%
Nigel M.	54	38.57%
Naishan G.	7	5%
Isaac G.	10	7.14%
Hemraj B.	12	8.57%

As of 2024-11-05 21:15, there are 136 commits, 35 related to the project source code and all 101 other commits related primarily to documentation.

4 Issue Tracker

Total Authored refers to all issues authored including those of team meetings and lectures, which are tracked as issues. Assignable Authored refers to issues that have to be assigned and resolved by a team member, including TA feedback that has been made into an issue for the purposes of tracking on GitHub. External

Authored refers to issues that have been authored in external repositories for the purposes of peer reviews. In the first column, External as a student refers to those external to our group that have authored issues through peer review.

Student	**Total Authored**	**Assignable Authored**	**Assigned Closed**	**External Authored**
Total	58	35	11	21
External	15	15	0	-
John P.	42	20	11	21
Nigel M.	1	0	0	0
Naishan G.	0	0	0	0
Isaac G.	0	0	0	0
Hemraj B.	0	0	0	0

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 msot of the advantages of CI/CD while still most closely adhering to the needs and limitations of our specific project.