**Snake Game**

**Zachary Yuan and Jonathan Li**

# PROJECT PROBLEM STATEMENT

The problem that we had was that we wanted to create a snake game that ran in javascript efficiently for our own uses. We also wanted it to be unique so we plan on adding in unique features allowing the game to be more interesting for the player.

# CONCISE DESIGN OVERVIEW

We plan on using type script and nodejs to run the project in vscode. Furthermore, we plan on using github as a way to organize all of our information efficiently. We also used next js in order to be more modern in our design. We have 5 main files for the game itself, the typescript files of: GameEngine, RoomGenerator, RoomGraph, Snake.ts, and Types.ts

# SCOPE STATEMENT

Snake game where the snake can move around, eat food, and grow larger. The ability to reset the game.

# TIMELINE

Draw a timeline, including milestones to serve as the basis for a work breakdown structure (WBS) and appropriate Gantt or PERT charts. For this class, you can refer to the Work Breakdown Excel Sheet. This can be a table with anticipated tasks listed for each school day of the project. You will submit completed chart(or Record of Tasks) along with your final deliverables.

# TEST PLAN

We will test the solution by running the game and seeing if aspects of it work. If the movement works, if hitting walls triggers the loss flag, if eating food extends snake length, if the visuals are working together and are clear to the user what each thing in the game is.

# RISKS AND CONTINGENCY PLANS

Project failing and nothing working – unlikely

Some bugs occurring – Likely

Our plans to mitigate these is to develop the game faster and accurately to account for any bugs.