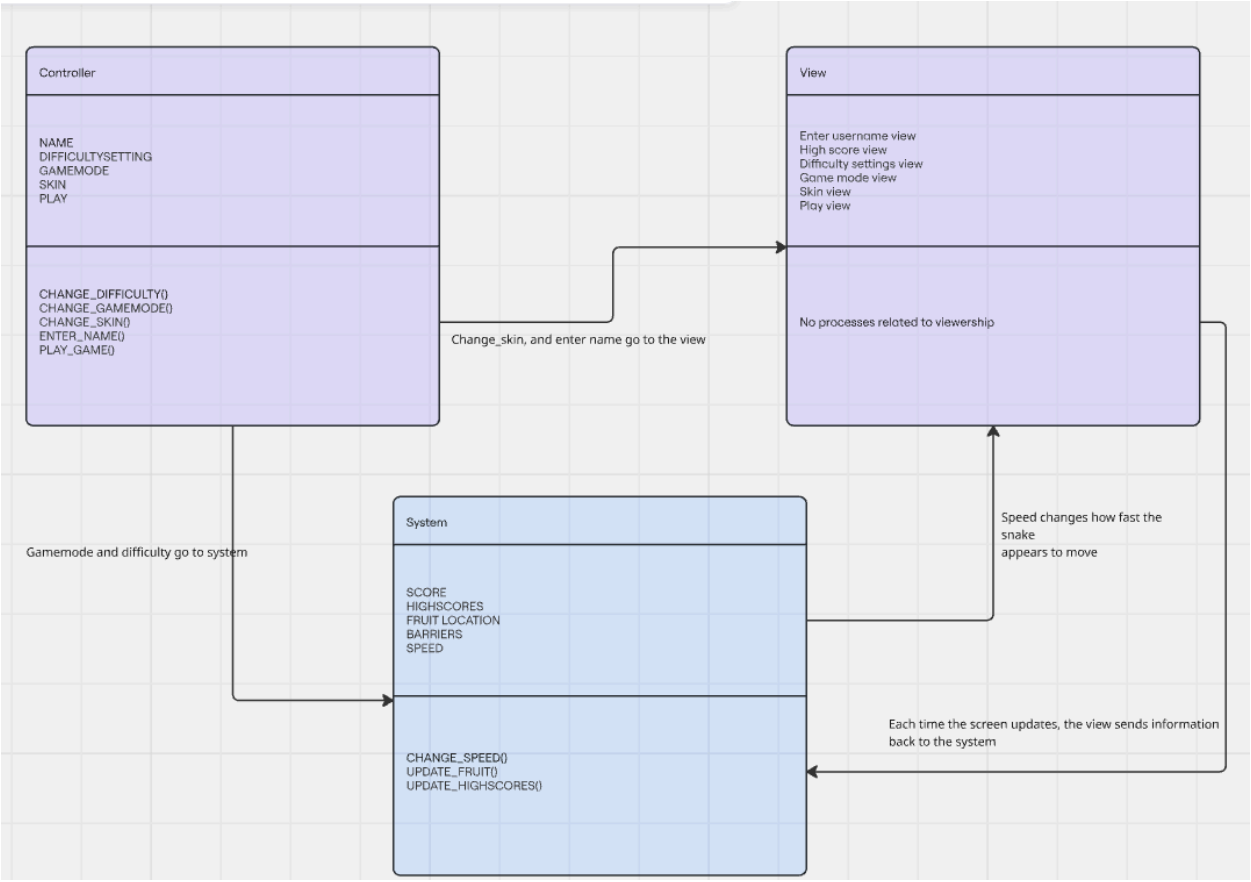
Team: Ceagulls

Architecture Design:

We designed a snake game, with the intention of innovating and modernizing it. Unlike pacman, tetris, and other well-known arcade games, snake has gotten considerably less attention. To do this, we decided to create different game modes and unlockable skins to keep the game fresh, as well as a global leaderboard to add a competitive aspect.

Architecture Design Diagram:



https://miro.com/welcomeonboard/S1NpaTd6RmdXdzc2RXZnSUNEZ00yVmpEVXY1OWJYQ2d5c3VseDRoYmQ2YnNwZ3cyZG9LcU1qL1BpY3JZLzlZS3FoR0lyQXovOGZCZUtQZ3hGM3dBdW5DSy9nVGpKMDR3cTdlOURtVXN3M3M5dDRiekVnY1YrS2dsWlplM2tRbnlNakdSWkpBejJWRjJhRnhhb1UwcS9BPT0hdjE=?share_link_id=486035734646

Design Description:

The architectural pattern consists of three components, the view, the controller and the model.

The view for Super Snake handles the UI for the game through React and Bootstrap, and can display several different screens. The enter name screen is where the user will enter the name for their character. This will trigger the first time the game is opened if the user does not have a name, otherwise the screen won't show up. Interacting with the UI can trigger other events such

as changing the skin or gamemodes, and this process sends information to the controller to handle.

The controller handles the various events from the view, such as simple button presses to change game modes or character skins, display the leaderboard, or start the game. When an input is detected, the controller then notifies the view of any changes if necessary and then notifies the model of any state changes.

The system for Super Snake handles the data for the app. The view accesses the data for what to display. The controller also notifies the system of any state changes, and if needed, sends out a notification to the view to then update the display. This allows the view to be updated when the user switches between screens in our system.