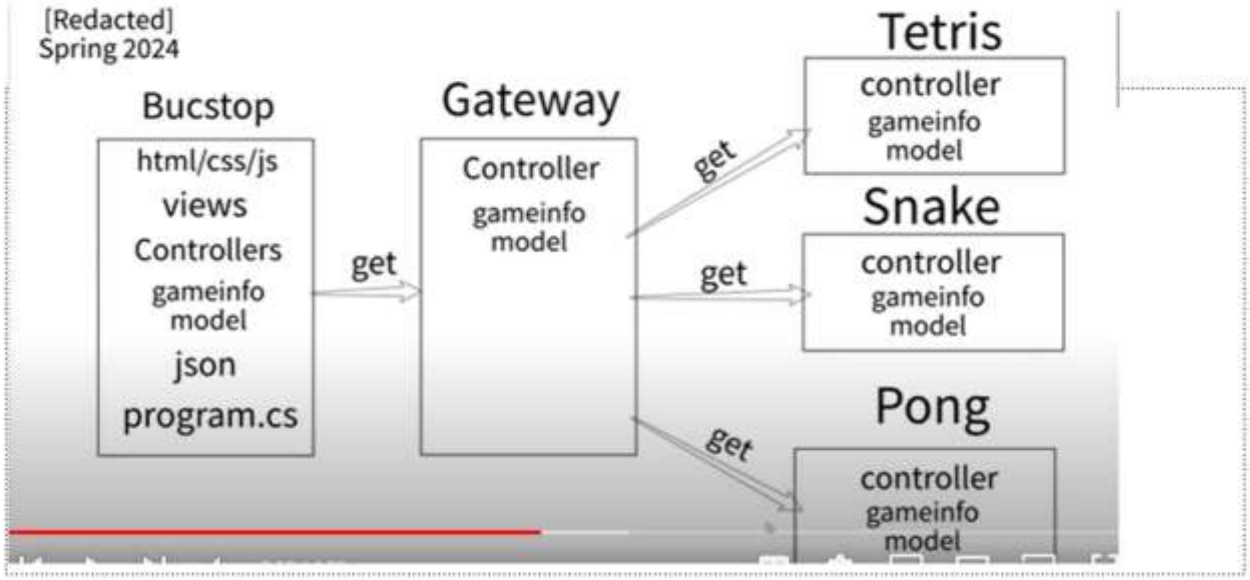
High Level Design



User -> Bucstop -> Gateway -> [Game Service]

Flow

□ User initiates an action on Bucstop (e.g., load Tetris).

□ Bucstop controller formats a request and sends it to Gateway.

□ Gateway routes the request to the correct game module.

□ Game module processes the request and returns relevant gameinfo.

□ Gateway forwards this data back to Bucstop.

□ Bucstop renders data on the frontend.

1. Bucstop (Client Interface)

- **Technologies:** HTML, CSS, JavaScript
- **Purpose:** Main user interface for interacting with the platform
- **Subcomponents:**
  - views: Frontend templates
  - Controllers: Handle user requests
  - gameinfo model: Data model representing game state/info
  - json: Handles data formatting
  - program.cs: Likely the entry point or backend logic (suggests .NET/C#)

2. Gateway (API Router)

- **Role:** Acts as the intermediary between Bucstop and the game modules
- **Responsibilities:**
  - Receives GET requests from Bucstop
  - Forwards those requests to the appropriate game module (Tetris, Snake, Pong)
  - Aggregates and returns game data
- **Components:**
  - Controller: Orchestrates communication
  - gameinfo model: Stores or forwards game-specific data

3. Game Modules (Tetris, Snake, Pong)

- **Each includes:**
  - controller: Handles logic and API interactions
  - gameinfo model: Maintains specific game state or metadata
- **Purpose:** Isolated microservices that provide game-specific data