
# Networking Document

Banana inc.

To enable multiplayer functionality in our game, we need a server to mediate communication between the players. One option for the server-side architecture is to use a client-server model, where the server acts as a central repository for game state information.

After each turn, the game state data will be stored locally on each player's machine and updated on the server. We can use a database to store game state information on the server and use a programming language with a database API like Python and its SQLite module to manipulate the database.

The server would then broadcast the updated game state information to all clients to ensure synchronized game board representation for fair gameplay.

To control turns, we can utilize a game loop on the server-side that periodically checks for player input and responds accordingly. This would ensure that the game proceeds smoothly and each player has a chance to take their turn.

In terms of game interface, one possible implementation is to use a lobby-like interface where one player creates a game and invites other players to join using unique game room ID numbers. Alternatively, we can use an API such as Socket.IO to enable real-time communication between players, allowing for a more seamless multiplayer experience.

In conclusion, implementing multiplayer functionality in "Can't Stop" would require a client-server model, the use of a database, and programming languages with database APIs. It is essential to ensure synchronized game state representation and a smooth game loop to control turns.