Sushi Game - Development Summary

Sushi Game is a multiplayer prototype created in Unreal Engine 5.6 using C++ for all core systems. The game supports listen-server mode and allows players to prepare sushi recipes collaboratively under time pressure. Players must collect raw ingredients, process them through multiple steps (slice, roll, cook), and deliver the final dish to the correct table to earn points. If an order expires or is delivered incorrectly, players lose points. Victory or defeat is determined by reaching a score threshold.

I structured the game using modular C++ classes for actors such as CookwareActor, IngredientActor, and TableActor, with gameplay logic centralized in OrderManager, SushiGameState, and SushiGameMode. The MatchState system drives all game flow transitions (Lobby, InGame, Victory, Defeat) and updates widgets accordingly. Only the host can start the match or restart it, ensuring correct replication.

The UI system is handled entirely via C++-driven widgets (WBP_Lobby, WBP_PlayerStatus, WBP_Victory, etc.), each showing relevant data like current recipe, progress, orders, and results. A pause menu with ESC/TAB gives players access to return to the lobby or quit. The lobby and gameplay run on separate maps to isolate logic cleanly.

I'm particularly proud of the reliable MatchState replication and intuitive player feedback through UI. Everything was thoroughly tested with multiple clients. If I had more time, I would polish animations, add sound feedback, and optimize UI scaling for multiple resolutions.

This task was enjoyable and allowed me to demonstrate my C++ structure, UI handling, and replication strategy under Unreal Engine multiplayer. I believe the result reflects a complete, functional, and well-organized prototype ready for presentation.