

Bon Appetit

Product Requirements Document

Johnathon Leon Guerrero jmlg@uw.edu | Katherine Cai klc715@uw.edu |
Hritik Aggarwal hritik@uw.edu | Laksh Gupta lq2000@uw.edu

01/21/22

Summary

This application is a co-op VR multiplayer experience set in a gourmet kitchen. Each chef is in charge of their own station and must coordinate with the other chefs to fulfill orders using realistic dishes. After each service, players can view their performance and track their progress over time.

Project Description

There are a lack of apps that provide a basic cooperative line cooking experience in VR. Although there are many cooking games, they are either too arcade-like focusing on teamwork rather than cooking, or they strive for full realism and lack a multiplayer aspect. While intended mostly for casual players to gain insight into the restaurant industry, this app can also be used by real kitchen staff in order to improve their communication skills and test their adaptability to new kitchen layouts. We aim to strike a balance between teamwork and realism that remains accessible and entertaining for both cooks and non-cooks alike.

Our project, Bon Appetit, is a 3-person co-op VR multiplayer experience built in Unity set in a custom kitchen setting. The 3 player roles will be split into 1 chef de cuisine, or head chef, and 2 station chefs. Orders will be completed by coordinating with teammates and following on screen prompts for following real French recipes. Head chefs will be in charge of receiving tickets and directing station cooks while station cooks will fry, grill, boil, bake, and combine different prepared ingredients as needed. We will be using real recipes with actual ingredient amounts and temperature requirements, but the cooking time will be reduced to retain player enjoyment. Players can earn cumulative and individual performance reports, based on how well they do their individual jobs and communicate with team members.

User Experience

Our user experience will focus on making the VR experience fun and enjoyable for all types of users while also having educational aspects like learning to cook and working in a team. We envision having a divided kitchen environment with a station for each player. The head chef receives tickets and must give directives to the other chefs in order to complete these orders. The chefs then follow recipes accessed through a UI to make components of each dish by combining and cooking different ingredients. Players will also be given a UI that helps them see how much of each ingredient they are using and what temperature their food is currently at through a UI thermometer.

Hardware Platform/Device

The application will be built on the Android SDK with Unity's XR integrations and can be deployed on any VR headset with native android support like the Oculus Quest 2.

Deliverables

- **Minimum Viable Product:** 3-player co-op, featuring the roles 1 chef de cuisine, 1 saucier, and 1 rotisseur. 1 multi-part dish of steak frites (steak, fries, bearnaise sauce). A ticket system for the chef de cuisine. A contextual UI for interacting with food items. A rating system for delivered dishes.
- **Target Product:** Same as the MVP but with 3-5 total recipes, 2-3 different kitchen settings, 4-player support, and progression tracking that combines the rating system from the MVP.
- **Stretch Goals:** Even more recipes and kitchen layouts. 2-player "mini kitchen" support. A versus mode to face off against another team of chefs.

Performance Metrics

For our weekly goals, we will have 2 remote standups per week to discuss individual progress and blockers. This will also allow us to keep track of overall team performance, which will be documented through weekly updates on our website.

We will develop stress test scenarios (high order volume and asset count) in order to ensure that our game remains at a playable frame rate (72-90 fps) and latency (under 100ms) for the multiplayer aspect.

We plan to test our MVP and target product by having other people help playtest our games. Having fresh sets of eyes would help us uncover bugs, ensure that features and UI are intuitive and function properly, and the difficulty of the game is appropriate. By having multiple teams with different people complete levels, we can measure their overall performance using the in-game rating system to decide whether or not to change timers or order volume.

Milestones

Week 3 (1/17 - 1/21)

- Milestone: Try out some basic features, find assets, and ideate further
 - Laksh - investigate multiplayer integration with Photon
 - Hritik - Start work on fries and frying features (scripts and assets)
 - Katherine - import boiling/cooking scripts from tutorial demo, begin researching ways to implement UI for seeing object information
 - Johnathon - import cooking scripts from tutorial demo, import new kitchen assets, create basic behavior scripts for temperature and cookable objects, and begin implementing prefab interactivity (doors, knobs)

Week 4 (1/24 - 1/28)

- Milestone: Get kitchen appliances and basic UI running
 - Laksh - begin implementation of multiplayer integration with Photon
 - Hritik - finish fry and frying features
 - Katherine - Implement a basic UI system for seeing metrics about objects (e.g. temperature, doneness of steak, seasoning amount)
 - Johnathon - finish interactivity with kitchen appliances and cooking behavior scripts. Record sound FX for MVP.

Week 5 (1/31 - 2/4)

- Milestone: Multiplayer alpha, cooking components for steak frites, and recipe UI
 - Laksh - finish alpha for multiplayer integration with Photon and ensure interactable objects work for multiplayer. Add basic player models.
 - Hritik - Begin work on ticketing system for head chefs
 - Katherine - Implement UI for recipes and helpful tips for players (e.g. highlighting an object that needs to be selected)
 - Johnathon - Finish steak and sauce components. Implement seasoning and garnishing for both steak and fries.

Week 6 (2/7 - 2/11)

- Milestone: MVP, multiplayer beta, and head chef station
 - Laksh - Polish any found bugs with multiplayer interactivity. Ensure that all stations can pass dishes/items to each other and maintain state.
 - Hritik - Finish ticketing system. Also implement ticket organization and crossing items off tickets.

- Katherine - Implement food rating system and add to UI for when level is completed
- Johnathon - Make final layout for the kitchen that incorporates all 3 stations. Implement a game system for placing orders and balancing game difficulty.

Week 7 (2/14 - 2/18)

- Milestone: Post-MVP, 2 more dishes, and new level(s)
 - Laksh - Work on adding 2 more new French dishes to the menu.
 - Hritik - Work on adding 2 more new French dishes to the menu
 - Katherine - Work on adding 2 more new French dishes to the menu
 - Johnathon - Create 1-2 new kitchens and level designs and work on sound design (background music, recording new sound FX, possibly purchasing new sound FX)

Week 8 (2/21 - 2/25)

- Milestone: 2 more dishes, main menu, 4-player co-op
 - Laksh - Add multiplayer support for 4-player co-op. Help work on sound design if needed.
 - Hritik - Work on adding new dishes to the menu
 - Katherine - Work on main menu and UI for new dish recipes
 - Johnathon - Create 4-player versions of levels

Week 9 (2/28 - 3/4)

- Milestone: Finish progression system, menus, and add any stretch goals
 - Laksh - Update player models if needed. Stress test multiplayer aspects.
 - Hritik - Finalize new dish variations for ordering system.
 - Katherine - Add a post-game menu (e.g. replay, change level, quit) and UI tweaks
 - Johnathon - Implement progression tracking and coordinate UI with Katherine

Week 10 (3/7 - 3/11)

- Milestone: Testing of final product and bug bashing
 - Laksh - Testing and bug bashing focusing on multiplayer systems
 - Hritik - Testing and bug bashing focusing on dishes and cooking features
 - Katherine - Testing and bug bashing focusing on UI and menu integration
 - Johnathon - Testing and bug bashing focusing on difficulty, level design, and environment

Materials and any external help needed

3D Assets: We will need to purchase assets for kitchen appliances, most food items, player models, and any tools that the players might use while in the kitchen (e.g. pans, plates, spatulas).

Software: If there is a food item that we cannot find an asset for, we will have to produce one in a 3D-modeling software like Blender.

Sound FX: We are planning on producing most of our own sound effects, but sounds like background music and ticketing noises will likely have to be purchased or licensed from a free site.

Outside Expertise: Johnathon has former coworkers working in the Seattle area that can be consulted for line-cooking, recipes, and overall direction

Budget

Pandazole - Kitchen Food low poly pack

Link: <https://assetstore.unity.com/packages/3d/props/food/pandazole-kitchen-food-lowpoly-pack-204525#content>

Cost: \$15

New York Strip Steal 3D model

Link: https://www.turbosquid.com/3d-models/new-york-strip-3d-model-1289382?fbclid=IwAR0NxYc_SsCLRpbLwv3rpoAku8mN0ApMCz2u8bLfiz4-DLYA93RtJA2LclU

Cost: \$1.99

Sound Effects and Music

Link: (Nothing specific yet)

Description: Cooking sound effects that cannot be produced at home and suitable background music to be determined after the MVP.

Estimated Cost: \$10 - \$20

Future Recipes

Link: (Nothing specific yet)

Description: Any assets that need to be purchased after the MVP for future dishes (tbd).

Estimated Cost: \$10 - \$20

Initial Cost: \$16.99 + taxes

Estimated Cost Over Entire Project: \$56.99 + taxes

Risks and how they will be addressed

Performance - With the amount of objects and physics within each scene, we may encounter low fps issues with the Quest 2. This is low risk due to the low-poly assets we have already chosen. If these issues do arise, we will reduce the number of objects in the scene (e.g. steak dispenser instead of a drawer full of steaks) and resort to object spawning/deleting as needed to keep performance up.

Plating - Being able to configure plates any way you want might prove to be harder than anticipated. If we can't guarantee fully customized plating options (like a canvas for the player, arranging items in any way, garnishing everything), we will use fixed plating instead (steak, fries, sauce in fixed positions on the plate).

Pan/Pot Interactions - Since unity does not support concave mesh colliders on non-kinematic objects, we will probably have to approximate colliders using primitive shapes. If we cannot feasibly do approximations in a reasonable amount of time, we will instead do a system that fixes the object being cooked to the inside of the pot/pan.

Multiplayer aspect - The photon engine may not be as reliable as we hope and may lead to latency issues. We plan to adapt our application accordingly as to limit multiplayer object interactions. If there are still issues with interaction latency or maintaining game state, we will reduce object interaction between stations and game-state tracking further until acceptable performance is achieved.

Collaborating Simultaneously - Using our current systems, two people won't be able to edit one object simultaneously (even if it's a different property). This will cause a lot of product delays as we wait for each other to finish working on a given scene/object. We plan to update each other about when we are working on which object and thus keep track of what aspect one can work on at a given time. As a backup, we will lock certain items while players are interacting with them.