# Steps to Take Before Game Implementation

\*\*Project Name:\*\* Kitchen Chaos

\*\*Game Engine:\*\* Unity

\*\*Target Platform:\*\* Mobile (Android/iOS)

## 1. Define the Game Concept

* • Game Genre: Define what kind of game it is (e.g., cooking simulation, time management).
* • Target Audience: Specify your main player demographic (e.g., casual mobile gamers, ages 10–35).
* • Core Gameplay Loop: Describe the main actions players repeat (e.g., cook dishes, serve customers, upgrade kitchen).
* • Monetization: Decide on how you'll earn (e.g., in-app purchases, rewarded ads, cosmetic upgrades).

## 2. Write a Game Design Document (GDD)

* • Story, Characters, and World: Define the theme and narrative (e.g., chef traveling and opening new kitchens).
* • Level Structure: Describe different kitchens and how they unlock.
* • Game Mechanics: Rules for cooking, time limits, orders, etc.
* • UI/UX Layout: General sketches or plans for buttons, menus, HUD.
* • Upgrade Systems: Define coins, items, or helpers that can be improved.

## 3. Sketch the Visual Style

* • Reference Images: Gather inspiration from similar games or Pinterest.
* • Art Style: Cartoonish, realistic, pixel art, etc.
* • Color Palette: Choose visually appealing and consistent tones.
* • 2D or 3D: Confirm Unity project setup.

## 4. Plan the Technical Architecture

* • Code Structure: Plan scripts such as GameManager, UIManager, ShopManager.
* • Core Systems: Input, Audio, Order Queue, Timer, Inventory.
* • Optimization Plan: Plan mobile-friendly structure (e.g., object pooling).

## 5. Prepare an Asset List

* • Visual Assets: Counters, ingredients, chefs, customers, buttons.
* • Audio Assets: Background music, cooking sounds, click feedback.
* • Animation Needs: Characters, cooking interactions, UI transitions.

## 6. Prototype Key Features

* • Build a Minimal Version: Implement just one recipe or one level.
* • Focus on Fun: Adjust interactions and pacing before content creation.
* • User Testing: Let friends try and give feedback early.

## 7. Version Control and Backup

* • Use Git: Create a GitHub or GitLab repo for your project.
* • Commit Often: Track every stage of development.
* • Backup Assets: Store a cloud backup of non-code content.

## 8. Platform Planning

* • Screen Sizes: Design UI for various mobile resolutions.
* • Input Types: Touch-only, avoid mouse-over or hover.
* • Performance: Plan for low-end devices (e.g., texture atlases, batching).