

Game Project: Mom Fury

Initial Plan

Brainstorm:

- RPG: but the character are ai chatbots
- Flappy bird but with ben's face
- Subway surfers but with ben's face
- **Top down shooter**
- Tower Defense
- Slot machine

Core Concept

In *Mom Fury*, players take on the role of a no-nonsense Asian mom defending her chaotic household from waves of misbehaving kids, clutter monsters, and rebellious chores. The game combines fast-paced top-down shooter action with humor and cultural flair, turning everyday parenting frustrations into over-the-top combat. Armed with iconic household "weapons" from the flying slipper and broom of justice to the deadly rolling pin, players must restore peace before guests arrive. The game's hook lies in transforming the relatable chaos of family life into a hilarious, action-packed battle for order. Players navigate through rooms, dodging spilled milk and flying toys while collecting "Respect Points" and power-ups like "Calm Tea Mode." With its mix of satire, intensity, and heartfelt energy, *Mom Fury* delivers a uniquely fun and culturally grounded user experience that celebrates and exaggerates the unstoppable power of moms everywhere.

Describe the core gameplay system of your game.

Mom Fury is a 2D top-down shooter built around fast-paced, chaotic combat and humorous household mayhem. The player controls an Asian mom navigating through rooms of her home as she fights off waves of mischievous kids, flying toys, and personified chores. The core experience combines twin-stick shooter mechanics with a comedic, culturally inspired theme, transforming everyday parenting frustrations into a high-energy battle for order.

The game is designed around a tight gameplay loop of clearing waves, collecting rewards, upgrading abilities, and advancing through progressively harder levels of domestic chaos.

The player's control system is central to the gameplay. Movement is handled through 8-directional input using a `CharacterBody2D`, while aiming and attacking use the mouse or right analog stick. The player has three primary attacks: the *Slipper Throw*, a ranged projectile that boomerangs back; the *Broom Sweep*, a short-range melee attack with knockback; and the *Mom Yell*, an area-of-effect stun powered by a "Patience Meter." These attacks operate under a player state machine, managing transitions between idle, attacking, dashing, and cooldown states. The system emphasizes timing and resource management—players must balance aggression with patience to maximize combos and survive increasingly frantic encounters.

Enemies are governed by individual behavior state machines consisting of states such as Idle, Chase, Attack, and Recover. Using Godot's `NavigationAgent2D`, each enemy can navigate around obstacles and pursue the player intelligently. Different enemy archetypes create variety and pacing: *Child Runners* dash toward the player in quick bursts, *Toy Drones* attack from range, and *Dust Bunnies* split into smaller minions when defeated. As levels progress, wave intensity and enemy stats scale dynamically, creating a sense of escalating challenge. Combat uses `Area2D` collision detection and signal-based communication for registering hits, triggering visual effects, and applying damage or stun effects.

A power-up and upgrade system adds depth and replayability. Enemies drop temporary power-ups such as *Calm Tea* (health recovery), *Tiger Balm* (speed boost), and *Double Slipper* (dual shot). Respect Points, the game's main currency, are earned through defeating enemies and maintaining combo chains. Between levels, players can spend these points to permanently upgrade attacks, reduce cooldowns, or unlock new abilities. This upgrade loop ensures long-term progression and player satisfaction, even after repeated playthroughs.

Each level represents a part of the household, such as the kitchen, living room, or backyard—serving as a distinct combat arena. After completing a series of waves, players face a boss fight or a time-limited cleanup challenge before advancing to the next area. Levels are structured as escalating "days," such as Morning Rush, Afternoon Chaos, and Dinner Disaster, each introducing new enemy behaviors and environmental hazards.

The overall player experience is designed to blend intensity with humor. The chaotic combat, expressive animations, and exaggerated sound effects create a sense of both

stress and comedy, echoing the energy of real-life family life. From a gameplay programming standpoint, *Mom Fury* focuses on robust input handling, modular AI systems, and responsive combat feedback loops that make every attack, dodge, and combo feel rewarding. The goal is to deliver a smooth, polished gameplay experience that balances technical precision with comedic charm—capturing both the chaos and the heart of being an unstoppable mom.

Explain your implementation plan to develop your core gameplay system.

We will use Godot. For version control we will use Github.

The core gameplay system for *Mom Fury* will be implemented in Godot.. Each system will be implemented using a combination of GDScript, Godot nodes (`CharacterBody2D`, `Area2D`, `NavigationAgent2D`, etc.), and the engine's built-in signal system to ensure efficient communication between components.

The first milestone will be implementing smooth 8-directional movement, aiming, and basic attacks. The player's attacks (slipper, broom, and yell) will each be scripted as modular abilities connected to an overarching Player State Machine that handles input transitions and cooldowns. Visual feedback such as recoil, hit particles, and sound effects will be added to ensure responsiveness and polish.

Next, the Enemy System will be developed using scripts. Each enemy type will share a base class for health, damage, and AI state handling (Idle, Chase, Attack, Recover). Navigation will use `NavigationAgent2D` for pathfinding, while attack behaviors and patterns will be configured through exported variables for easy balancing. The Wave Spawner System will handle enemy generation and difficulty scaling based on the player's performance metrics, such as elapsed time and Respect Points earned.

The **Combat System** will be implemented through Godot's physics and signal-based architecture. Attacks will be represented by `Area2D` nodes that emit signals on collision with enemies or objects, triggering damage, knockback, or stun effects. Combo and multiplier logic will be managed within the combat manager, rewarding players for precision and continuous attacks.

Following this, the Upgrade and Power-Up System will be built. Temporary power-ups will be instantiated as collectible objects, while permanent upgrades will be managed through a simple data persistence layer using Godot's `ConfigFile` or JSON saving system. This allows players to progress between levels and customize their playstyle.

Finally, the Level and Progression System will tie everything together. Each room (level) will be a self-contained scene that defines its layout, enemy spawns, and environmental hazards. Transitions between rooms will be handled by a Level Manager that tracks player progress and triggers victory or defeat conditions.

Throughout development, each component will be tested individually and then integrated through iterative playtesting. This ensures that gameplay systems interact smoothly, attacks feel responsive, enemy behavior remains predictable yet challenging, and upgrades meaningfully impact player strategy. By focusing on modular code design and frequent testing, the implementation plan ensures a stable, extensible foundation that supports new content and features later in production.

Why do you want to make this game?

We chose this concept because we all found it humorous, relatable, and deeply engaging. As a team of Asian developers, we wanted to create a game that not only entertains but also celebrates our shared cultural experiences in a lighthearted way. Growing up, many of us witnessed, or were on the receiving end of, the classic "Asian mom energy," defined by tough love, high expectations, and a unique blend of discipline and care. We want to use our game to reimagine a familiar stereotype into something empowering and endearing.

This project gives us an opportunity to highlight aspects of our upbringing that resonate across cultures while maintaining authenticity through small details from weapon choices like slippers and brooms to the emphasis on order, family, and respect. It's both a playful homage and a form of cultural storytelling, where we transform something personal and specific into something universally funny and emotionally resonant. Ultimately, we hope *Mom Fury* captures that balance of chaos and heart that defines the experience of growing up in an Asian household.

Explain your gameplay and the major systems it is composed of in terms of other games and genres.

Non-Core Gameplay Systems and Features

System	Description
Power-Ups and Upgrades	Temporary boosts (Calm Tea, Double Slipper, Tiger Balm) and permanent upgrades purchased with Respect Points between levels.
Combo and Respect System	Players earn Respect Points through combos, damage avoidance, and cleanup speed. High streaks trigger “Proud Mom Mode,” temporarily boosting stats.
Cutscenes and Narrative Moments	Short, humorous cutscenes between levels depict mom’s escalating frustration and determination, ending with comedic vignettes such as kids apologizing or cleaning together. Boss fights are prefaced by mini cutscenes (e.g., “Laundry Beast Rises” or “Homework Warrior Challenge”).
Level Progression	The game progresses through the day’s household timeline—Morning Rush, Afternoon Chaos, Dinner Disaster—with each stage introducing new arenas and environmental hazards like slippery floors or flying utensils.
Endings	Multiple endings based on Respect Points and patience meter balance: “Peaceful Home” (perfect mom run), “Moderate Chaos” (average balance), and “Mom Meltdown” (low patience score).
UI / UX	Minimalistic, colorful interface: circular patience meter, combo counter, and visual pop-ups for Respect Point streaks. Menu uses playful household icons (e.g., slippers for “Play,” broom for “Settings”).
Audio Design	Mix of traditional instruments (gu zheng, wooden percussion) and

upbeat chiptune beats. Dynamic layering increases tempo during boss fights.

Major Scenes Kitchen Battle, Living Room Showdown, Backyard Boss Fight. Each arena includes destructible objects, environmental interactions, and escalating waves.

Major Classes PlayerMom (state-driven controller), EnemyChild, ToyDrone, DustBunny, BossLaundryBeast, PowerUpItem, and WaveManager for level orchestration.

Assets Hand-drawn sprites for mom, enemies, and projectiles; particle effects for dust, sparks, and “Respect Burst”; background tilesets for different rooms; UI icons for tea, broom, and slipper upgrades.

Core Mechanics Summary Players navigates with WASD/analog stick, aims with mouse/right stick, attacks with 3 abilities, collects Respect Points, upgrades between levels, and survives timed waves to progress. Winning strategy involves managing cooldowns, spatial positioning, and patience meter to chain combos efficiently while minimizing hits.

List your non-core gameplay systems and features.

Your Team

Member: [An Le, anble@ucdavis.edu]

Role, confidence, explanation.

My Role is Game Logic because I want to learn game development as we're building out this project. I'm comfortable with basic coding structures but still gaining experience applying them to gameplay systems, so I'm moderately confident. I'll be responsible for ensuring the game feels responsive, balanced, and technically smooth.

Subrole, confidence, explanation.

My subrole is Visual Cohesion because I have had experience with creating sprites in my

previous projects. I know what it takes to have a cohesive user interface along with the sprite on my previous game project built on typescript. I'm very confident in this role.

Hope to learn or accomplish.

I hope to ship this project as we have planned it very well and I think got a lot of potential to be a very fun game. At the same time I want to learn how to design and implement a fully functional game from scratch.

Anticipated challenges.

Debugging complex interactions between projectiles and physics. I'm still rusty and got a lot to learn.

Member: [Ben Lim, bslim@ucdavis.edu]

My chosen role is user interface and input as I believe that is the aspect of the project that I am most confident in coding. For my subrole, I chose Gameplay Testing as I feel this fits my main role the closest and allows me to be the most efficient with my coding and testing. I hope to improve my programming skills and understanding of coding as a group through this project, and I also hope to produce something fun and memorable. My biggest anticipated challenge would be my relative lack of coding experience, which may make me slower at doing my work than I would like.

Member: [Quynh Tran, pqtran@ucdavis.edu]

Role, confidence, explanation.

My main role is Animation and Visuals role because I enjoy bringing characters and emotions to life. I'm confident in using tools like Godot's AnimationTree and sprite-based effects to make actions

Subrole, confidence, explanation.

My sub role is Narrative Design. I'm moderately confident. I have strong creative writing skills and enjoy character driven storytelling but I haven't had much experience intergating narrative to interactive gameplay.

Hope to learn or accomplish.

I hope to improve at combining visual storytelling and gameplay flow, I also want to gain

hands-on experience in animating for game feel.

Anticipated challenges.

Time management during asset creation and iteration will be a key challenge.

Member: [Kyle Carbonell, kcarbonell@ucdavis.edu]

My role in this project is to create the movement / physics of the characters. I am only somewhat confident in my skills, since I have only been doing game dev for a couple of months now. My sub role for this project is game feel. I am a bit more confident with this role since I think I have a good understanding of what these kinds of games should feel like and how users would perceive our game. Some things I hope to learn are how to create a better game and give users a better feel and understanding in playing our game. I think some challenges for this project would be creating smooth movement.

Scheduling

Provide a timeline for completing your project.

Sprint 1: Basic gameplay systems implemented and story outlined

Sprint 2: Assets and audio designs

Sprint 3: Gameplay systems finalised and story fully implemented

How would your game be if everything went as planned?

It would be a fully fledged story based top down shooter complete with a sophisticated upgrade system, multiple encounters and vivid, cartoony graphics.

What would the results be if you lost significant time (e.g., you lost two weeks due to unforeseen circumstances)?

We would reevaluate as a team what would be a realistic goal, and scale down the scope of our game if needed. This could be in the form of making the story shorter than initially planned, or cutting out gameplay systems that are less necessary to delivering the core premise and value of our game.

If your progress is faster than expected, how would your game change?

Some of our stretch goals include expanding the game's representation by introducing different types of mothers to reflect cultural diversity and broaden the player experience. For example, future iterations may feature characters inspired by the experiences of Caucasian, Latina, and other mothers, each with their own distinct challenges, family dynamics, and play styles (appropriately of course, we'll interview to make sure). This addition would not only enhance gameplay variety but also encourage players to engage with multiple cultural perspectives, highlighting both the universal and unique aspects of parenthood across different communities.