WITCHER RPG

Jan 2023 - Present

My Role

Game Designer
Project Manager

Summary

This is an MMORPG game built on Minecraft, featuring an ARPG-style combat system. It incorporates elements such as puzzles, cooperation, and character development, aiming to enhance player social interactions and provide a richer online gaming experience.

TeamWork

This is a medium-sized team project that has been ongoing for over a year, with the team growing from 2 members to nearly 15. In this project, I have been responsible for various roles, ranging from programming and game design to project management.

Team

Lizhuoyuan Wan Zhaoyang Gong Xuanyu Wang Qianyu Zhang
Hongyu Yang Guanping Peng Zekun Yang Yiyang Mao
Zhou Dong Xijun Ma Wang Lang Jialong Chen Junxuan Li Xiwei Huang

Inspiration 💡

The inspiration for this game stems from the social connections fostered through gaming. To me, a game serves as a medium that brings people closer together. From creating simple Flash-based multiplayer games to local network co-op games, and now online multiplayer experiences, the role of social interaction in game development has become increasingly significant. In my view, games are not just a topic of conversation but a unique bond that allows me and my friends to adopt distinct identities in the game world. They help us escape the burdens of daily life, offering pure joy and enthusiasm through social interaction. This is precisely the reason behind creating this project: to use games as a social tool, providing a rich and engaging social experience in an alternate world.

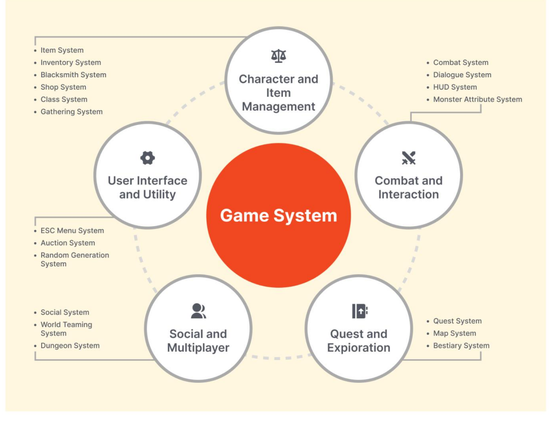
System

System design serves as the foundation for other aspects of game design, making it my top priority.
The goal: to enhance players' social and combat experiences by designing systems that meet the requirements of both MMORPG and ARPG games.

Game System

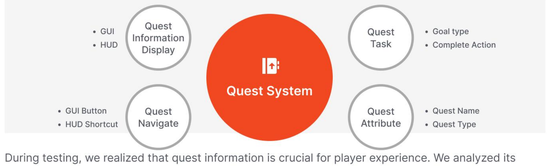
I categorized the systems into five primary aspects: character-related, interface-related, combat interaction, social, and exploration. Starting from these categories, I systematically expanded each aspect, developing interconnected features and mechanics to create the foundation of the initial version of the game. Each aspect was designed to ensure a cohesive experience, balancing player interaction, gameplay dynamics, and immersion.

Some systems were developed as part of our focus on social interaction, while others emerged as we identified needs during the development process.

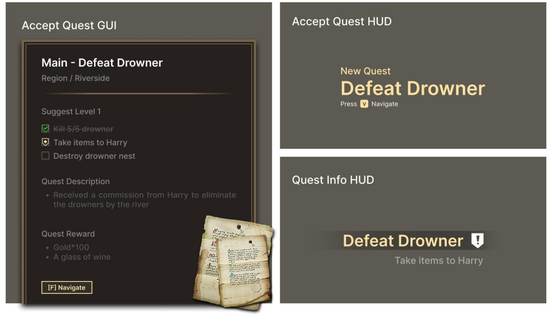


Quest System

The quest system consists of two key components: information display and the data content of the quests.

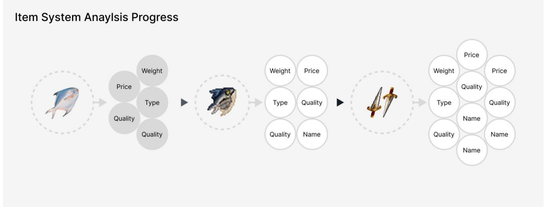


During testing, we realized that quest information is crucial for player experience. We analyzed its display across multiple dimensions: in the inventory interface and within the HUD for quest acceptance and progress updates.

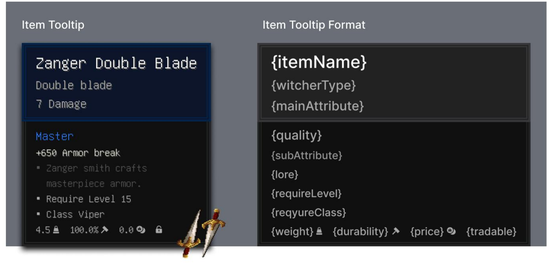


Item System

When designing the item system, I drew inspiration from real-world objects, extrapolating their elements into game items to include as many relevant attributes as possible from the outset.

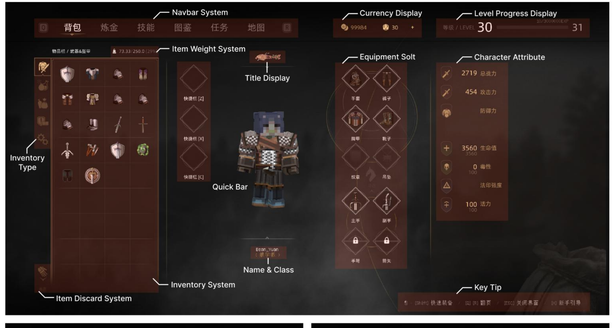


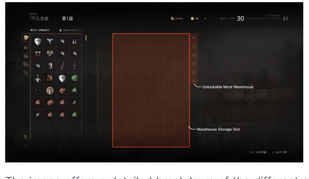
After considering item attributes, I designed a system capable of adapting descriptions for various types of items to accommodate future expansions and new item categories.

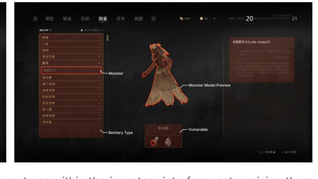


System GUI

The image shows a screenshot of the inventory interface, displaying various systems from the player's point of view.







The image offers a detailed breakdown of the different systems within the inventory interface, categorizing them into functional and informational compaonents.

Combat

Combat system: the most direct interaction point for players, making stability and responsiveness crucial.
The goal: to ensure a smooth and engaging gameplay experience, where the combat system enhances immersion and supports the overall enjoyment of the game.

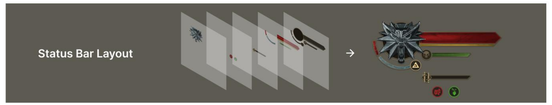
Combat System

I classified the game systems from two perspectives. The first diagram provides a complete classification from a designer's perspective, grouping systems by their functions, such as combat, exploration, or character management.

Although our art quality may not match that of larger companies, after incorporating multiple dimensions like visual effects, sound design, and other elements, the combat experience now feels more complete and engaging.

Combat HUD

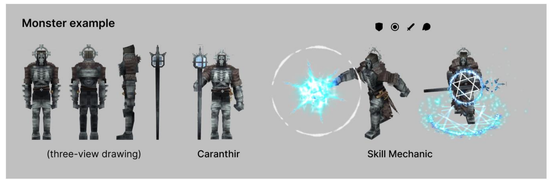
During the gradual development process, we realized that the initially designed HUD was merely a basic imitation, lacking clarity and functionality in many areas. We then decided to reassess our approach, starting from the actual needs of each system and summarizing their requirements for HUD displays. Ultimately, we recognized that the HUD is the most frequently interacted interface during both combat and regular gameplay.



The HUD primarily includes displays related to combat and essential information, such as consumable hotkeys, player status bars, combat button prompts, buff status, boss health bars, quest information, and the mini-map. Above is the final outcome of our refined HUD design.

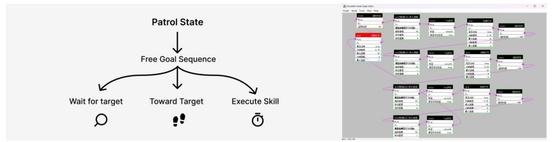
Monster Design

For the boss battle design, I initially referenced mechanics from mature MMORPGs and applied them to our boss encounters. However, during beta testing, we found these designs didn't meet our intended scenarios. It was clear that we hadn't fully analyzed our expectations. In the latest iteration, we revisited our goals and added supportive mechanisms.



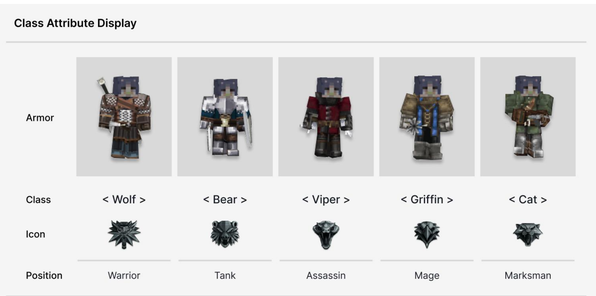
Mob State Machine

During testing, we found that monsters lacked clear objectives, as their state machine was still based on the original Minecraft system. To create a more strategic battle, we introduced our own behavior state machine. During skill cooldowns, monsters previously moved aimlessly. We added a "maneruver" state, where monsters maintain a strategic distance—closing in if the player moves away and pulling back if the player gets closer. This change brought greater depth and strategy to combat.



Class Character Design

Class character design is inspired by the roles of healer, tank, and damage dealer. I aimed for players to rely on coordination among different roles during battles. The initial setup includes roles such as tank, assassin, healer, archer, and off-tank. These roles are differentiated based on the three core archetypes, ensuring players must cooperate as a team to clear dungeon instances.

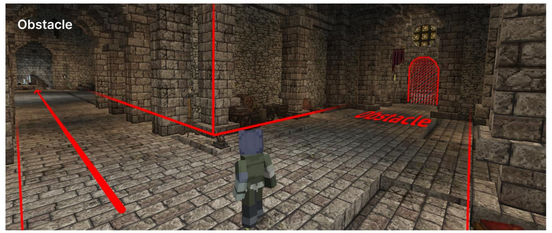


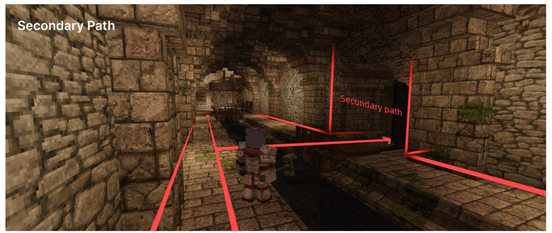
Level

The goal of level design is to guide players in exploring the game world and to create meaningful challenges by thoughtfully planning the flow of each stage.

Core Theory

In the early stages of level design, I was puzzled about what makes a level truly well-designed.

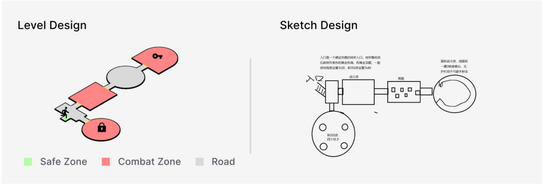




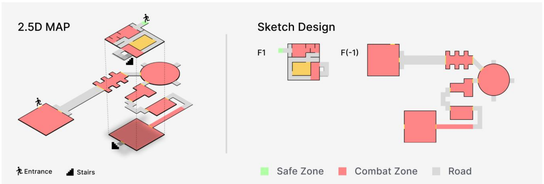
Later, I researched level design theories online and began to understand that some seemingly simple and common design elements are actually indispensable. Features like secondary paths and obstacles can create a more dynamic level layout, enhancing the overall experience.

Level Design

In the map below, I used obstacle design elements. Players can see the boss room from the beginning but will find themselves unable to enter. This prompts them to take another path, where they obtain a key at the end of the route. The obstacle helps to make the level sequence straightforward and clear.



In this level, I utilized numerous occluder elements to add a sense of mystery and maintain visual control, ensuring that players receive information as intended during the design phase. Additionally, I incorporated elements like a box-pushing mini-game and 2D jumping traps to break up the long gameplay and avoid monotony, adding a touch of variety and fun. Throughout the level, I introduced gradual changes in visibility, from narrow to expansive views, to visually signal the importance of upcoming battles.

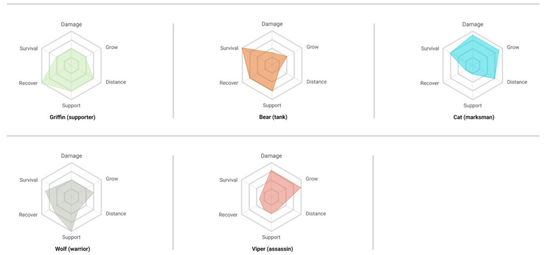


Numerical Design

The aim of the balancing design is to control the game's difficulty through well-tuned values, while integrating system design elements to enhance the overall player experience.

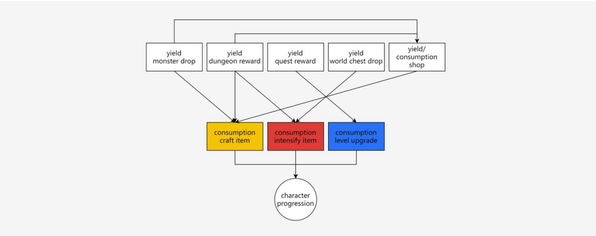
Class

When designing character values, I first set baseline parameters at each level for balance. Then, based on class roles—like a warrior with medium attack and moderate defense—I defined attribute growth trends for all classes. After establishing a clear role for each class, we defined the attribute growth for each one. We then formed different class combinations and calculated their potential combat damage. This helped us determine the highest possible output, ensuring that the monster stats were well-calibrated and met our expectations.



Output and Consumption

In designing the resource production and consumption system, I wanted to ensure that nothing was overlooked during detailed planning. So, I first used a mind map to categorize each system by its resource outputs and consumption. Early on, I defined dungeon battles as the primary means for players to gather resources, with the player's main goal being to improve their character's attributes. This improvement could be through forging equipment, leveling up, or similar actions. After clarifying this flow, I created a process diagram and iteratively refined the production-consumption balance.



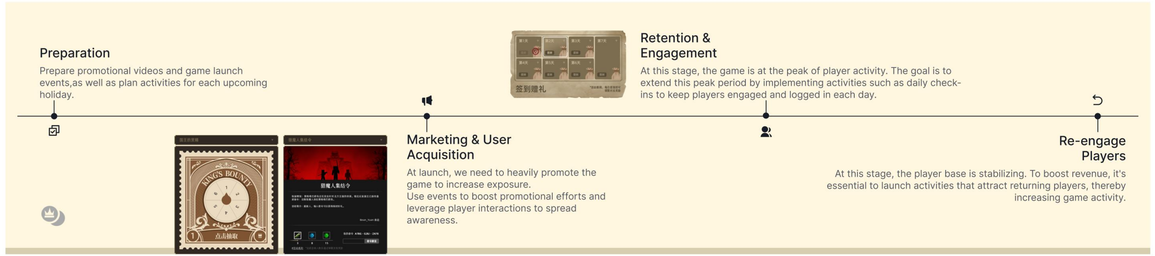
For experience production, I designed it to vary across different chapters. In the first chapter, players gain experience from relatively simple tasks, creating a gradual learning curve. This allows the gameplay experience to progress smoothly from easy to challenging, avoiding overwhelming players from the outset.

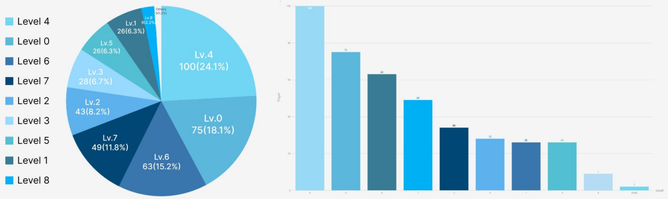
Game Operation

Game operations primarily involve implementing a series of strategies to manage and promote a completed game product in the market, thereby generating revenue. It plays a crucial role in the overall success of the game project.

Game Operation Experiences

My first understanding of game operations came during the C-testing phase of this game. During testing, we frequently interacted with players to gather valuable feedback, realizing that maintaining player engagement and keeping the community active is an essential part of game operations. With this insight, I developed a timeline for the operational plan in preparation for the official launch.





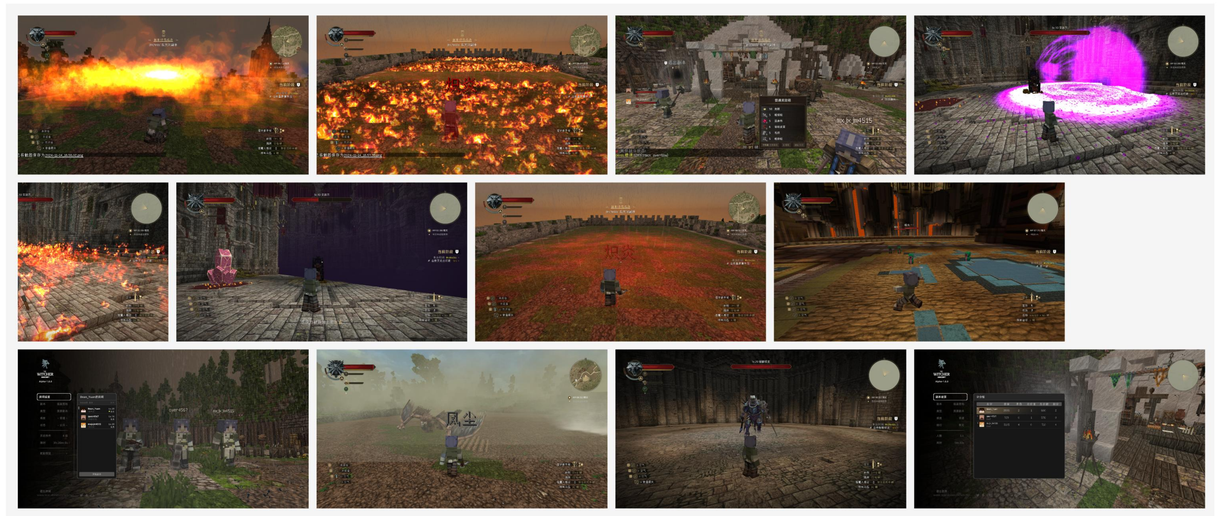
During the C test phase, we noticed no significant player growth despite making changes based on active players' feedback. Realizing this might not reflect the majority's experience, we analyzed player data directly and found most players were stuck at level 4, indicating an issue at that stage. We optimized the game accordingly, and as a result, players began progressing normally, with most advancing beyond level 4.

For the official launch, we prepared promotional events and activities to maintain player engagement. For instance, we introduced a prize wheel where players could earn a spin by sharing the game, and a referral program where players received rewards after referring a certain number of new players. These initiatives utilized players as our primary promotional channel to boost overall online activity.

Conclusion

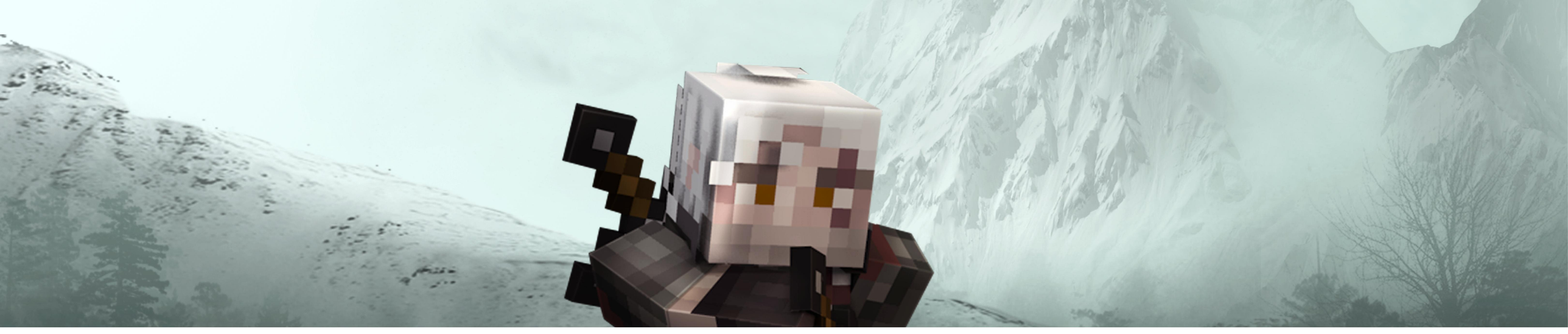
.

This project has become more than just creating a game for me. Throughout its development, I encountered many areas I had never previously explored, and these experiences have been incredibly valuable. It's this process that has motivated me to strive for perfection in every aspect of the work. Beyond game design and the challenges that came with it, I also greatly value the collaboration with my team members. It made me realize that creating a truly outstanding game requires collective effort and a unified team. Effective communication is the key to overcoming all obstacles, and I hope to continue gaining fresh and exciting experiences as a game designer and producer, transforming my ideas into reality.



🕹️ Game Development Part

↓



WITCHER RPG

Jan 2023 - Present

My Role

Game Developer

Summary

This is an MMORPG game built on Minecraft, featuring an ARPG-style combat system. It incorporates elements such as puzzles, cooperation, and character development, aiming to enhance player social interactions and provide a richer online gaming experience.

TeamWork

This is a medium-sized team project that has been ongoing for over a year, with the team growing from 2 members to nearly 15. In this project, I have been responsible for various roles, ranging from programming and game design to project management.

Team

Lizhuoyuan Wan Zhaoyang Gong Xuanyu Wang Qianyu Zhang
Hongyu Yang Guanping Peng Zekun Yang Yiyang Mao
Zhou Dong Xijun Ma Wang Lang Jialong Chen Junxuan Li Xiwei Huang

P1

The combat system is the core of my game development, as it is the key to player experience in an ARPG.

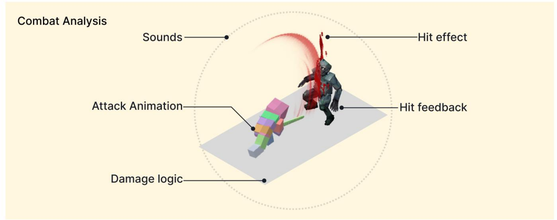
Combat 🏹

Key aspects of development:

- Optimizing player-monster interaction: Making battles more dynamic and engaging.
- Improving combat immersion: Refining mechanics for a more intense and realistic experience.
- Ensuring responsiveness: Making every player action impactful and responsive.
- Balancing complexity and playability: Combining complex mechanics with accessible gameplay.

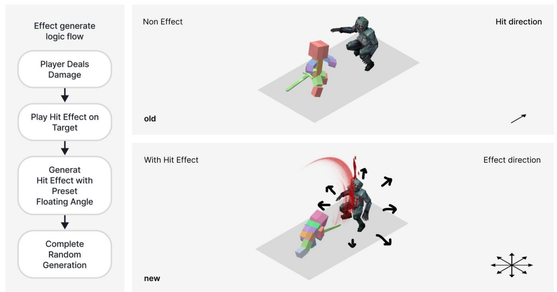
Combat Overall

Improve the overall combat experience feedback through multidimensional analysis.



Damage Effect Display

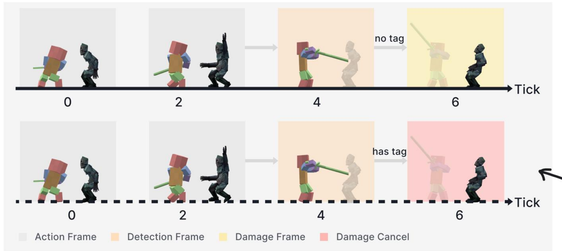
Use a more realistic approach to display hit effects.
For instance, consider the following bleeding hit effect: the direction of the effect changes according to the player's attack movement, and the bleeding angle randomly fluctuates within a certain range.



Damage Order

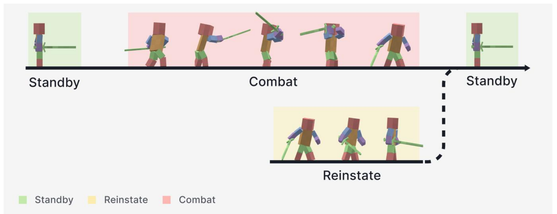
Make the code logic more aligned with real combat behavior.

The following damage logic resolves the issue of visual inconsistency during combat, making the effect more realistic by having the opponent stop their attack action when being hit.



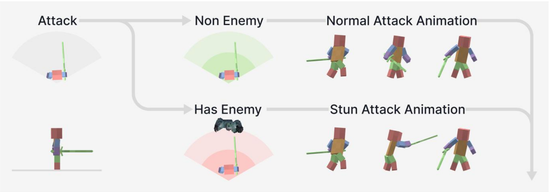
Action Transition

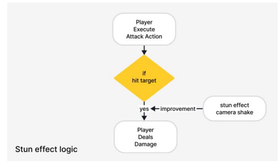
Optimize the transitions between combat actions.
To address the issue of a jarring instant idle after an attack, introduce transition animations to make the shift from attack to idle state smoother and more seamless.



Stun Effect

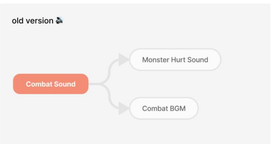
By introducing a damage feedback logic flow, pause effects and camera shake effects are added during attacks to enhance the sense of impact during combat.





Combat Sound

Compared to the previous version, we have added more comprehensive sound effects to achieve a more immersive and satisfying combat experience.



System

P2

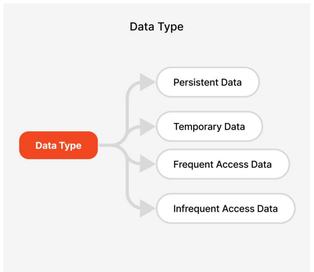
The various systems in this game connect various player experiences, ensuring a cohesive and engaging gameplay environment.
The core of my development is focused on data read and write operations.

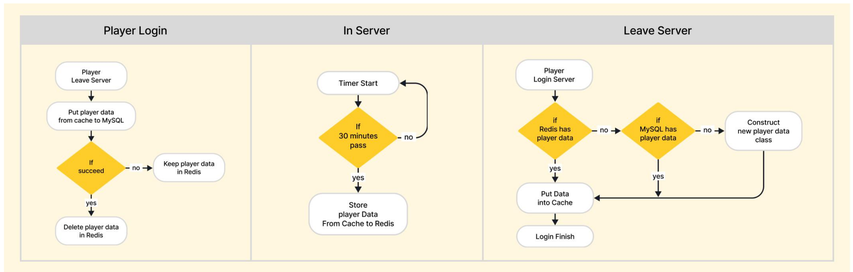
Key aspects of development

- Enhancing data security: Preventing data loss in case of server crashes and ensuring safe data handling.
- Improving data processing speed: Optimizing algorithms for fast and seamless gameplay.
- Increasing code coupling: Strengthening interaction between modules for cohesive functionality.
- Enhancing code extensibility: Creating a modular structure to support future updates and expansions.

Data Management

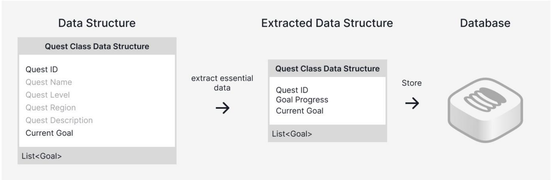
I believe that the core aspect of an MMORPG game lies in handling data. Therefore, I first categorized the game data. For each type of data, I have a general processing framework.



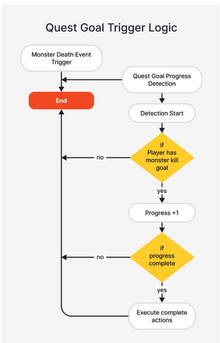


Quest System

Due to the large volume of data in the quest system, I devised a method to extract and store only the key information in order to achieve faster data retrieval and reduce the data processing load on the server.

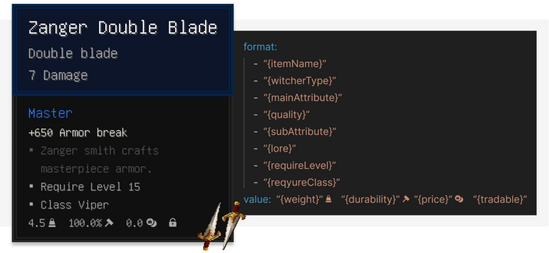


For quest objectives, instead of using periodic loop checks, I implemented an event listener approach. This allows the game to update players' quest progress more quickly and efficiently, resulting in better performance.



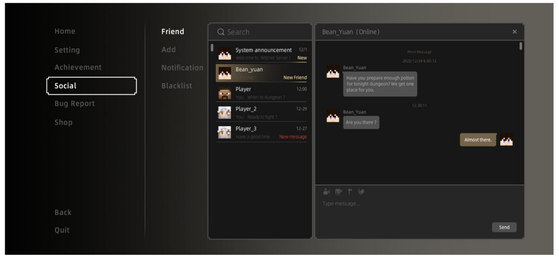
ItemStack System

Anticipating the numerous item types and their corresponding text formats in the game, I developed a modular, customizable text format editor.



Social System

During the chat system development, I initially expected adaptive interface displays to be the biggest challenge. However, real-time updates proved more complex. To ensure consistency and avoid redundant development, I centralized event handling, reducing the risk of missing conditions or requiring manual code updates.



Level

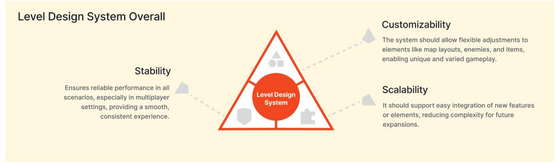
Level design system development primarily emphasizes providing designers with a stable, highly customizable, and extensible development tool.

Key aspects of development:

- High customizability: Allowing for flexible level design.
- Stability: Ensuring consistent performance across levels.
- Extensibility: Supporting future content additions and modifications.

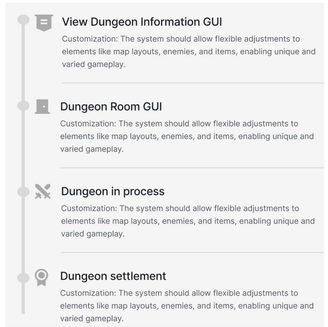
Level Design System Development

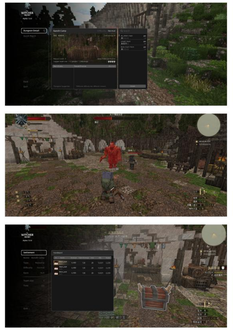
As both a designer and developer, I focused on creating a process-oriented level design system. I aimed for high customization to allow independent design, ensured code extensibility for future needs, and prioritized stability for frequent player interactions.



Dungeon Gameplay Process

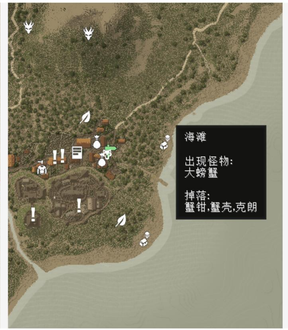
Dungeons are a crucial part of the level design. During development, I established the following process as the core sequence for players to experience dungeons.

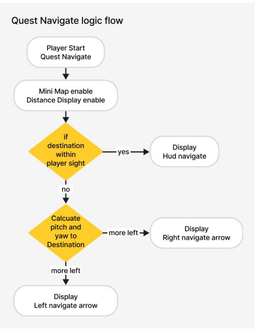




Integrated Navigate System

In level design, guidance often seems intuitive to developers but can be confusing for players, as observed during multiple tests. Therefore, providing clear and concise guidance is crucial. Initially, we designed a HUD navigation system, but it proved insufficient as players could miss objectives when facing away from them, and there was no clear sense of distance. To address this, we introduced screen-edge navigation and a mini-map, enhancing the overall guidance experience and ensuring that even first-time players could navigate the game with ease.





Code Sample