

[Proposal](#) (WIP)

Consolidated breakdown of our product proposal

Positioning

Release and Support Plan

How we position the game in our marketing copy, community communications, marketing initiatives, content road map will vary vastly between these types of products. This must be explicitly defined and accounted for *before* the first release to players, as it determines the majority of the support plan (from both the development and marketing side), and sets the tone for community expectations and engagement. *It is very difficult to pivot after the game is public.* Any disparity in “Player conception of the product” and “True reality of the product” will have EXTREME negative repercussions including, but not limited to:

- Negative Reviews/Metacritic score
- Player Churn
- Loss of Brand Trust/Customer Loyalty (Videocult/Akupara)
- Lack of Sales
- Lack of Biz Dev opportunities due to negative player perception

Note that while Akupara will have specific recommendations for how to implement each of the bulletpoints below, ultimately the more important aspect is *knowing what the solution is for each item*, and planning accordingly.

Live Ops vs GaaS

The biggest thing to determine is whether we are considering this game a Live Ops or true GAAS model. This will set the tone for community expectations around post-launch content and cadence.

GaaS aims to create an ongoing revenue stream and keep players engaged with new content and features.

Live Ops focuses on maintaining the current state of the game, ensuring player satisfaction, and addressing immediate operational needs.

GaaS

- **Definition:** GaaS is a business model where games are developed and maintained as an ongoing service rather than as a one-time product. This model focuses on providing continuous content updates, features, and engagement to players over time.

- **Revenue Model:** Typically involves various monetization strategies such as microtransactions, season passes, and downloadable content (DLC). The idea is to keep players invested and spending money over the long term.
 - This often includes seasonal, time-limited or subscription-model based payments.
- **Development Focus:** Emphasis is on creating a game that can be expanded and updated regularly. This often involves designing the game with future content updates and live features in mind from the start.
- **Examples:** "Fortnite," "Destiny 2," and "Apex Legends" are prominent examples where the game evolves with new content and features introduced regularly.

Live Ops

- **Definition:** Live ops is primarily concerned with ensuring a smooth player experience, and responding to live player feedback. This includes the management of in-game events, updates, community engagement, and troubleshooting.
- **Revenue Model:** This is usually positioned as a single premium title.
 - Can include optional cosmetic purchases or paid DLC, although DLC is often delivered as free updates.
- **Components:** Live Ops can include seasonal events, special promotions, bug fixes, server maintenance, player support, and community engagement efforts.
- **Examples:** In a game like "World of Warcraft," live ops might involve managing in-game events like holiday celebrations, addressing bugs, and rolling out patches.

Early Access Vs Open Beta Vs Premium

Key Differences

Development Status:

- **Early Access:** The game is still in development; incomplete and subject to change.
- **Closed Beta:** The game is early in development, closed beta is used to test core features, balance gameplay mechanics, and identify critical bugs with a smaller group.
- **Open Beta:** The game is nearly complete; mainly testing for final adjustments.
- **Premium Titles:** The game is fully complete and polished.

Access and Cost:

- **Early Access:** Paid access to an unfinished game.
- **Closed Bets:** Free, temporary access to an in development game for a limited time.
- **Open Beta:** Free access to a nearly complete game for a limited time.
- **Premium Titles:** Paid access to a finished game, with additional content available for purchase.

Purpose and Feedback:

- **Early Access:** Gather feedback and refine the game.

- **Closed Beta:** Gather feedback, generate hype, build community.
- **Open Beta:** Test stability and performance, generate hype.
- **Premium Titles:** Deliver a complete, polished experience with less emphasis on ongoing changes.

Early Access

- **Definition:** Early Access is a model where a game is released to the public before it is fully finished or officially launched. Players can purchase and play the game while it is still in development.
- **Purpose:** The main goals are to gather player feedback, test gameplay features, and refine the game based on real-world use. It helps developers identify and fix issues before the final release.
 - Early Access can be used to help fund the development of the game while it is still ongoing.
 - Players are more invested in the outcome of the title as they have paid for access, and expect to be more involved in the direction of the game leading up to release. They feel a sense of both *ownership* and *entitlement* compared to Open Beta players.
 - Early Access generally lasts longer than Open Beta - an indefinite, sometimes several year period.
- **Access:** Players generally pay to access the game, even in its unfinished state. The price can vary and often reflects the game's current level of completeness.
 - This will often include a discount to the full retail price, as well as exclusive content such as Founder's Packs.
 - This can also grant specific rewards within the community, such as Discord titles/access, private previews and events, etc.
- **Development Stage:** The game is in active development. Features, content, and systems might be incomplete or subject to significant changes based on feedback and ongoing development.
- **Examples:** "Hades," "Stardew Valley," and "Subnautica" were available in Early Access before their full official releases.

Closed Beta

- **Definition:** A closed beta is a limited access test period used to refine, polish, and test the game in a controlled environment before a broader release.
- **Purpose:** Since access is limited, closed betas provide more control over who participates, allowing us to focus on key demographics, devices, or regions, and to gather targeted feedback from those groups.
- **Access:** A closed beta is only open to a select group of players, typically by invitation, application, or selection from sign-ups. Sometimes, access may also be granted through early purchase or pre-order bonuses.

- **Development Stage:** A closed beta can happen at any stage of development or as multiple closed beta periods throughout early development. It usually precedes a more general open beta period closer to release.
- **Examples:** “Valorant,” “Diablo IV” and “Apex Legends Mobile” all had closed beta periods.

Open Beta

- **Definition:** An Open Beta is a testing phase where a game is made available to the public for free, usually shortly before or leading up to the official launch.
- **Purpose:** The main objectives are to test the game’s stability, performance, and balance under real-world conditions and to attract a larger audience. It’s also used to build hype and gather feedback from a broad player base.
 - Players would anticipate the game being in a mostly feature-complete state, and for the full launch to be *no greater* than 1 year away, but generally closer to 6 months.
- **Access:** Free to all players or to those who sign up. Open Beta periods are typically limited in duration and may have restrictions or limitations compared to the final release.
 - There can also be timed where Open Beta is only for a limited window of time, and game access is restricted again leading into full release.
- **Development Stage:** The game is nearing its final release. While it might still have bugs or incomplete features, the core game is mostly in place, and the feedback gathered is used to polish and finalize the product.
- **Examples:** “Battlefield V,” “Destiny 2,” and “Call of Duty: Modern Warfare” have had open beta phases before their full releases.

Premium

- **Definition:** Premium Titles refer to fully completed games that are released for purchase as a finished product. They are generally priced higher than free-to-play or early access games.
- **Purpose:** The game is complete and polished, intended to provide a complete experience at launch. There is often less emphasis on post-release content updates compared to GaaS models.
- **Access:** Players must buy the game at full price.
 - There might be additional costs for downloadable content (DLC), expansions, or other extras, but the core game is sold as a complete package.
- **Development Stage:** The game is finished and has undergone extensive testing and quality assurance. It’s ready for full release and is not expected to change significantly in terms of content or features.
- **Examples:** “The Last of Us Part II,” “Red Dead Redemption 2,” and “God of War” are examples of premium titles that are released as fully completed products.

Road Map

Regardless of whether we choose an Early Access or Open Beta model, a Road Map is NON-NEGOTIABLE. The Road Map should outline a general release cadence, with some definitive planned features and tentative dates, along with some “teases” that allow for marketing activations, and some fluidity based on player feedback during the EA or OB phase.

The Road Map should outline, at minimum, a clear visual representation of the following:

- **Regular Updates:** Communities look for frequent updates about the development progress, even if there are delays or changes to the roadmap.
- **Specific Milestones:** Detailed milestones and goals for content updates, including new features, bug fixes, balance changes, and expansions.
- **Timelines and Progress Indicators:** Estimated release dates for upcoming content, even if these are tentative and subject to change.
- **Regular Content Drops:** Regular additions of new content to keep the game fresh and engaging, along with descriptions of upcoming features, improvements, and changes.

Outside of our visual Road Map (which can be handled in many different ways), we should simultaneously offer the following community engagement opportunities:

- **Feedback Mechanisms:** Opportunities for the community to provide feedback, such as through forums, surveys, or in-game reporting tools.
- **Responsive Development:** Evidence that player feedback is being considered and, where appropriate, acted upon in subsequent updates.
- **Q&A Sessions:** Regular Q&A sessions where developers address community questions and concerns directly.
- **Bug Fixes and Performance Improvements:** Regular updates that address bugs, performance issues, and other technical problems.
- **Balance Updates:** Adjustments to game mechanics to ensure fair and balanced gameplay.

Competitive vs Casual

Many of our design choices on both the development and marketing side will be determined largely by whether we are positioning the game as *competitive* or *casual*. These are completely different audience segments, with different core gameplay motivations and expectations around the gameplay experience - both the moment to moment gameplay, and the meta/gameplay loop/features surrounding the gameplay itself.

Key Differences

It is worth noting the [12 gamer motivations](#), and which of these we are:

- Trying to appeal to
- Effectively appealing to

Competitive Multiplayer Games

Content Focus:

- **Gameplay Balance:** Players expect frequent updates to ensure game balance, preventing any single strategy, character, or weapon from becoming overpowered.
- **Competitive Integrity:** A strong emphasis on fair play and anti-cheat measures to ensure a level playing field.

Gameplay Systems:

- **Ranked Systems:** Well-defined ranking systems that provide clear indicators of player progress and skill.
 - This would also include *visual indicators* of Prowess, including rare, skill and completion. These players are driven by Prestige.
- **Tournaments and Events:** Regular competitive events and tournaments with meaningful rewards and recognition.
- **Skill-Based Matchmaking:** Effective matchmaking systems that pair players of similar skill levels to ensure fair and challenging matches.
- **Training Tools:** Features like practice modes, detailed statistics, and replay analysis to help players improve their skills.

Content Updates:

- **Frequent Patches:** Regular updates that address balance issues, introduce new content, and fix bugs.
- **Patch Notes:** Detailed patch notes that explain changes and their intended effects on gameplay.

Community Management:

- **Direct Communication:** Direct and frequent communication from developers, often through forums, social media, and developer streams.
- **Community Feedback Integration:** A strong focus on integrating community feedback into game updates and balancing decisions.

Casual Multiplayer Games

Content Focus:

- **Diverse Content:** A wide variety of content, including new maps, game modes, and cosmetics, to keep the experience fresh and engaging.
- **Inclusive Design:** Features that make the game accessible to a broader audience, including casual gamers and families.
- **Fun Factor:** Emphasis on fun and accessible gameplay rather than intense competition.

- **Ease of Play:** Simple and intuitive game mechanics that are easy to pick up and play, catering to a wide range of skill levels.
- **Engaging Storylines:** Focus on worldbuilding, story elements and lore to keep players engaged and invested in the game world.

Gameplay Systems:

- **Social Interaction:** Strong social features that encourage playing with friends, such as robust friend lists, in-game chat, and party systems.
- **Community Events:** Regular in-game events and activities that encourage community participation and social interaction.
- **Casual Progression Systems:** Progression systems that reward players without requiring intense commitment, such as daily challenges and seasonal rewards.
- **Non-Competitive Goals:** Goals and achievements that focus on exploration, creativity, and personal milestones rather than competitive rankings.

Content Updates:

- **Replayability:** A wide variety of content, including new maps, game modes, and cosmetics, to keep the experience fresh and engaging.

Community Management:

- **Community Recognition and Engagement:** Community initiatives that focus on the community itself - fan art contests, community shout-outs and MVPs, etc.

Monetization

What is our plan to make money, and what content is gated by purchases? What kinds of purchases can be made, and what is the price point?

Are we selling cosmetics only, or seasonal/time-limited content, or gameplay-relevant content?

Our release plan and monetization plan have to support one another, and will determine what kind of content we need to make, how *much* of that content we need to make, and how we distribute it to users both in-game and on the platform/storefront side.

Revenue Models

- Premium
 - One time purchase up-front
 - Subsequent DLC can be free or paid, or a mix of both
 - As an example, DLC can be paid when it includes new content/maps/characters, but free when it includes new features
 - We can release paid, optional, cosmetic DLC packs alongside each free content update

- All DLC can be paid
 - All DLC can be free
- Season passes are also an option in this model, granting players access to all current and subsequent paid DLC updates and cosmetic packs for a smaller up front fee than buying them individually. These will unlock automatically with each new release.
- Freemium
 - Free-to-Play with In-App Purchases
 - Players can download and play the game for free but pay for in-game items, currency, consumables or cosmetic upgrades.
- Subscription-Based
 - Players pay a recurring fee to access the game or its premium features. This is common in GaaS models.
 - Season passes, where players upfront for access to time-limited content, including exclusive missions, challenges, or items.
- Virtual Currency
 - Players buy virtual currency with real money, which they can then spend on in-game items or features.
 - Use both premium and soft currencies, with premium currency often purchased with real money.
- Gacha/Loot Boxes
 - Limited-time loot boxes tied to special events or promotions.
 - Players spend money to receive a randomized reward, often in the form of loot boxes or gacha pulls.

SKUs

Depending on our pricing and release models, we may want to create different bundles for purchase.

- Founders Pack/Pre-Order Bonus
 - Founders Pack bundles in video games are special packages offered to early supporters or backers, often during a game's development phase or early access period. These bundles are designed to reward players for their early investment.
 - Can include exclusive in-game items, in-game currency, physical goods, special recognition both in-game and in community channels, exclusive access or features, and community perks.
- Deluxe bundle
 - By default includes art book and OST bundled into base game
 - Can also include all current and future DLC, cosmetics, etc
- Season Pass
 - Can be sold either in a subscription model or as part of a standalone bundle

Granting Entitlement and In-Game Rewards

Depending on our pricing and release models, there are various ways we can grant ownership and entitlement to users. It is possible to use a key-based system so that exclusive rewards can be given to community members or influencers.

- Account-Based
 - **Game Accounts:** Players create accounts with the game or platform (e.g., Steam, Xbox Live, PlayStation Network) where their ownership and entitlements are managed. This allows for cross-device access and tracking
 - **Account Linking:** Likewise, we can enable linking between different platforms or services (e.g., linking a game account to a social media account) to consolidate ownership and entitlements.
- Platform-Based
 - **Digital Storefronts:** Ownership is managed through digital storefronts like Steam or console stores. Purchases are tied to the player's account on these platforms.
 - **In-Game Wallets:** Some games use an in-game wallet or inventory system to manage purchased items, currency, or other entitlements.
- In-Game Systems
 - **In-Game Storefronts:** Players purchase in-game currency, items, or content directly within the game. Entitlements are managed by the game's internal system and often synced with the player's account.
 - **Progress Tracking:** Tracking and granting entitlements based on player progress or completion of certain tasks within the game.
 - **Skill Based:** Players may need to hit certain criteria or thresholds during seasons or on leaderboards in order to unlock specific rewards.
- Key Redemption
 - **In-Game:** Keys are generated and redeemed from within the game itself, allowing us the opportunities for giveaways and rewards within the userbase.
 - **External:** Likewise, we can generate keys using external mechanisms, granting entitlements to exclusive rewards using the platform systems.
- Third Party Drops
 - **Prime Gaming:** Prime Gaming offers several different levels of API integration that allows us to distribute exclusive in-game rewards. To access Prime Gaming Drops, players need an active Amazon Prime membership.
 - Prime Gaming Drops are often used as promotional tools to drive engagement, and encourage players to try their games or watch certain streamers.
 - **Prime Gaming Website:** Players can browse available drops on the Prime Gaming website. The site lists the games and the specific in-game content available through the drops.
 - **Twitch Streamers:** Some drops are tied to watching specific Twitch streams. Players need to watch participating streams for a certain period to earn rewards.

Currency

- In-Game:
 - **Standard:** Currency used within the game to purchase items, upgrades, or services. It is often earned through gameplay, such as completing missions or defeating enemies (e.g., gold in "World of Warcraft").
 - **Premium:** A higher-value currency that usually speeds up progression or allows for the purchase of premium items. It is often bought with real money and can be used for special items or features (e.g., V-Bucks in "Fortnite").
 - **Crafting/Resource:** Used in crafting systems to create or upgrade items. These materials can be gathered through gameplay, exploration, or other means (e.g., ores or herbs).
 - **Progression:** Experience or skill points used to measure and track player progression and level up characters, gear or skills.
- Real-World
 - **Microtransactions:** Currency bought with real money to spend on in-game items, upgrades, or cosmetic features. This can be a form of premium currency or direct purchases (e.g., buying coins or gems).
 - **Subscription-Based Currency:** Currency received as part of a subscription service, often used for accessing premium content or exclusive items (e.g., monthly rewards in subscription-based games).

User Experience

First Time Player Experience

The first time player experience must have a low barrier to entry and lead to player retention by being:

- Accessible
- Having proper tutorialization
- Allowing the player to get into the action and gameplay quickly
- Providing a quick reward/feedback loop
- Giving a good sense of the game experience

Matchmaking

For a multiplayer product, one of the most important aspects is finding available games. The goal - generally - is to get into gameplay as quickly and effortlessly as possible.

There are multiple potential pitfalls resulting in difficulty matchmaking depending on variable factors such as the size of the audience queuing, their platforms, skill-level, match preferences, region, server, ping, etc. If players are unable to find a game quickly, consistently, and without issue, this is the number one cause of player churn and dissatisfaction - often leading to community sentiment around "dead game." Matchmaking is the core way in which players

interface with a multiplayer game, and any other feature or content updates are moot unless the product provides a strong foundational infrastructure for the multiplayer experience.

- Too many choices for matchmaking can lead to audience segmentation and longer wait times.
- Too few player-driven choices can lead to boredom and churn and limits replayability, customization and individual satisfaction, especially during late-game.
- Friends-only matchmaking excludes players who have no friends to play with.
- No skill-based matchmaking can impede the new player experience, as they matchmake with players far beyond their skill and experience level. Conversely, higher skilled players might experience frustration at seemingly one-sided matches.
- Lack of bots means players can't matchmake during off-peak hours, can't practice their skills autonomously and may get stuck in perpetual matchmaking queues.
- As the audience drops off over the lifecycle of the game, lack of bots renders the game essentially unplayable.

Player Queuing and User Flow

Note that it is possible to do a combination of these options for a middle ground solution.

- Skill-Based Matchmaking/MMR
 - Players are matched based on their skill level, often determined by metrics like win/loss ratios, performance stats, or a ranking system.
 - **Pros:** This results in more balanced matches, leading to competitive and satisfying gameplay. This also reduces the chances of matches being imbalanced.
 - **Cons:** New players or those with fluctuating skill levels might struggle to find matches that are appropriate for their current ability.
- Team-Based
 - A variation on skill-based matchmaking, where players are matched against each other based on the combined skill levels of their members.
 - **Pros:** Allows for less restrictive matchmaking that still takes skill level and balance into account.
 - **Cons:** If the system doesn't account for the synergy between players, some teams might still be unbalanced, or get paired with griefers or sandbaggers.
- Progression based
 - Players are matched based on specific completion criteria within the game, usually as part of a narrative story-mode or leveling system.
 - **Pros:** Allows for some level of skill-based gating without needing complex algorithms to weigh skill level.
 - **Cons:** Completion criteria doesn't necessarily translate to skill level, and can still lead to unbalanced games or audience segmentation. Likewise, can make matchmaking with friends difficult, if they have not progressed as deeply within the game.

- Dynamic
 - Adjusts matchmaking dynamically based on real-time data, such as player performance in ongoing matches.
 - **Pros:** Can respond to real-time changes in player performance, potentially leading to more balanced matches.
 - **Cons:** Requires sophisticated algorithms and real-time processing, which can be technically challenging.
- Randomized/Open-Ended
 - Players are matched randomly, with little to no regard for skill, preferences, or behavior. The match preferences are randomized by the game systems.
 - **Pros:** Easy to implement and understand, with no complex algorithms required, and often provides shorter wait times to get into gameplay.
 - **Cons:** Player experience can vary greatly from game to game, impacting overall enjoyment. More skilled, competitive or late-game players may feel boredom at a lack of individual choice.
- Player Choice
 - Players are matched based on their preferences, such as play style, preferred roles, or game modes.
 - This allows players to enjoy games in ways they prefer, increasing overall satisfaction, but can result in longer wait times, especially if players have niche preferences.
- Crossplay/Cross-Platform
 - Implementing cross-platform multiplayer matchmaking allows players on different platforms to play together.
 - **Pros:** Allows for a higher pool of players for matchmaking, and extends to life of the game. This feature also reads as more “professional” and “AA/III” to players.
 - **Cons:** Many technical hurdles and complexities, including designing userflow, platform-agnostic friends lists and/or accounts, QA and live-issues complexity, TRC compliance, managing latency and performance, integration of platform-specific APIs, security, and maintenance of version parity.
- Friends List
 - Using a friends list-only matchmaking system in video games means that players can only be matched with people they have added to their friends list.
 - **Pros:** Playing with known friends reduces the risk of encountering toxic behavior or malicious players, as the players are generally familiar to each other.
 - **Cons:** The matchmaking pool is restricted, which can lead to difficulties in finding matches, especially in games with fewer players or if the friends list is small. If friends are inactive or not online, players may struggle to find a match. It also complicates cross-platform releases, as crossplay is significantly more difficult to implement, and some platforms will have smaller userbases compared to Steam.
- Lobby System
 - A lobby system allows players to find open games, form teams, set preferences, and sometimes even chat before the match begins.

- **Pros:** Allows for on-the-fly match customization without segmenting the audience through matchmaking. Lobbies provide a space for players to interact and socialize before the match, which can enhance community building and player engagement.
- **Cons:** Can potentially be imbalanced teams, or take longer to get into gameplay. Also requires complexity in UI/UX design and technical infrastructure.

Game Modes

Online Modes

Online-only gameplay can make sense for multiplayer games, and be supported using a variety of matchmaking options.

Pros: No need for developing a robust narrative or single player experience.

Cons: Players with poor connectivity or during server outages cannot access the game and the game requires a stable internet connection to play. Likewise, if we ever choose to shut down the servers, the game becomes unplayable, leading to a loss of player investment and title longevity.

Online Matchmaking

- *Human Only:*
 - The core of multiplayer games is the online experience with other human players. If we choose a multiplayer-only approach, we should ensure that there are deep and compelling enough systems, paired with a variety of gameplay to keep players engaged long term. This can be a variety of character classes, maps, modes, gameplay variables and/or match types.
 - **Pros:** Playing against human opponents provides a greater challenge due to their unpredictability and adaptive strategies. Developers can focus resources on other aspects of the game, such as new content, features, or balancing, instead of developing complex AI.
 - **Cons:** There may be times when matches are unavailable due to a lack of players. Lack of bots means lack of tutorialization, leading to a steeper learning curve and the potential for player churn.
- *AI/Bots:*
 - Bots and AI characters can also be included as an option for online play, to help reduce matchmaking time.
 - **Pros:** Ensures that matches are available even during off-peak hours or in regions with fewer players. Allows the potential for more niche options when matchmaking.
 - **Cons:** Bots may not provide the same level of challenge or unpredictability as human players, potentially leading to a less engaging experience. The presence of bots can break the immersion for some players, especially if the bot behavior is noticeably different from human players.

Offline Modes

While human-only games are the preference for the majority of players, there are several reasons to include an offline component, mainly to extend title longevity and allow for gameplay when matchmaking is unavailable for any reason.

- Campaign/Story Mode
 - A narrative-driven mode where players follow a storyline, often with missions, cutscenes, and character development.
 - **Pros:** This helps immerse players in the lore of the world, and allows them to build skills and unlock options for the multiplayer experience. Some players prefer a single player experience to a multiplayer one.
 - **Cons:** Requires significant design and writing time to create a satisfying offline story mode.
- Mission/Scenario/Objective Mode
 - Predefined scenarios or missions with specific goals and conditions, offering a focused gameplay experience.
 - **Pros:** Can offer an offline experience that familiarizes the player with the world and the gameplay mechanics that mainly repurposes existing multiplayer maps and content, and doesn't depend on a fully realized narrative. Also creates an offline mechanism for unlocking specific multiplayer rewards.
 - **Cons:** Additional scope of work for design and engineering.
- Practice/Bot Mode
 - A mode designed for players to practice their skills, learn game mechanics, and get familiar with maps or characters without the pressure of competitive play.
 - **Pros:** New players can learn the game mechanics and improve their skills by playing against bots, which are often less challenging than human opponents and can often be customized in terms of difficulty.
 - **Cons:** Requires significant development time to create robust and satisfying AI competitors.
- Couch Co-Op/Split Screen
 - A human multiplayer experience that is played on the same console and screen.
 - **Pros:** Very few games currently offering this mode, and it's still a favorite among retro gamers and families in particular.
 - **Cons:** Unique challenges for optimization, map design and UI/UX in particular.
- Daily Challenge

Game and Community Features

Core Features

Meta and Gameplay Loop

The core gameplay loop is the sequence of actions that players repeatedly perform during a game. This loop is designed to be engaging and rewarding, keeping players hooked and

encouraging them to continue playing. This is everything that is happening outside of the moment-to-moment gameplay, including leveling, unlocking rewards, customization, socializing, exploring, etc. This is the glue that ties the in-game experience together.

The meta refers to the overarching strategy, trends, and player behavior that emerge outside of the core gameplay loop. It involves the broader decisions and systems that affect how the game is played over time. The meta can evolve based on updates, player strategies, and community interactions.

The core gameplay loop keeps players engaged on a moment-to-moment basis, providing immediate satisfaction and feedback. The meta provides long-term goals, strategic depth, and a sense of progression, encouraging players to invest more time and effort into the game.

Player Retention Plan

What is our plan not only to reach new players, but to retain players and build a consistent, engaged multiplayer community?

- What are we doing to target New Vs Lapsed Vs Retained Players?
 - Replayability, variety and end-game content
 - Completionism and achievement hunting
 - Notoriety and skillfulness
 - Exploration, discovery and immersion
 - Community-based events and social challenges
 - New content, challenges and features

DLCs/Content Updates

There are many different ways to release additional, post-launch content. Each of these methods can additionally be paired with various live ops models.

- Expansion Packs
 - Content may include new maps, game modes, and environments, additional story content or lore, new characters, weapons, and abilities, expanded progression systems, like new levels or skill trees.
 - **Pros:** Provides a large amount of content, which can reignite player interest and attract new players. Players often perceive expansion packs as providing good value for their money. Can rejuvenate the player base and encourage lapsed players to return.
 - **Cons:** Typically more expensive than other forms of DLC, which may deter some players. If not all players purchase the expansion, it can split the player base, especially in terms of multiplayer matchmaking. Requires significant development time, which can delay the release of new content.
- Season Pass
 - Bundled DLC content released over time, often including new maps, modes, characters, cosmetics, and additional story missions. Usually contains exclusive items or early access to content for pass holders.

- **Pros:** Encourages players to stay engaged with regular content drops. Provides developers with early revenue that can support ongoing development. Players feel they are getting a deal by purchasing content in bulk rather than individually.
- **Cons:** Players may be hesitant to buy a season pass without knowing the exact content they'll receive. Developers face pressure to deliver high-quality content consistently, or risk disappointing pass holders. If some players have the pass and others don't, it can lead to disparities in content access.
- **Cosmetic-Only**
 - Includes paid DLC only for skins for characters, weapons, and vehicles, emotes, sprays, and other non-gameplay affecting items and customization options like banners, avatars, and profile customization.
 - **Pros:** Doesn't impact gameplay balance, ensuring a level playing field for all players. Can be a lucrative source of income, especially with popular or themed cosmetics. Easy on the development side to create.
 - **Cons:** Some players may feel cosmetic DLC doesn't offer enough value, especially if prices are high. Constant release of cosmetic items might lead to player fatigue or devaluation of older items.
- **Free Updates with Optional Paid Content**
 - Free updates that include new maps, modes, balance changes, and bug fixes for all players paired with optional paid DLC that might include exclusive skins, characters, or additional content.
 - **Pros:** Ensures that all players have access to core gameplay updates, keeping the player base united. Players who wish to support the game can do so by purchasing the optional DLC, while others still enjoy the core experience. Still keeps the full community engaged with regular updates that everyone can access.
 - **Cons:** Relying on optional paid content might limit revenue compared to more aggressive monetization strategies. Some players may feel the paid DLC doesn't offer enough value if too much content is given away for free. Developers may face challenges in balancing the content offered for free versus what is behind a paywall.
- **Event/Timing-Based**
 - DLC may include time-limited events with exclusive rewards, challenges, and game modes or thematic content drops, such as holiday-themed skins, maps, or modes.
 - **Pros:** Events can drive player engagement, encouraging them to log in and participate before the event ends. Keeps the game feeling dynamic and alive, with regular injections of new content. Limited-time events can drive sales and participation due to the perceived scarcity of the content.
 - **Cons:** Players may feel pressured to participate in every event, leading to burnout. New players or those who miss the event may feel left out or at a disadvantage. Introducing new modes or mechanics temporarily can disrupt game balance or create bugs.

- Story-Driven
 - Content most regularly includes additional campaign missions or story chapters, or new characters with unique backstories and gameplay mechanics.
 - **Pros:** Expands on the game's story and world, providing players with more context and immersion.
 - **Cons:** Might not appeal to players focused solely on multiplayer aspects. Story-driven content typically requires significant development resources, including writing, voice acting, and animation.

Chat

For multiplayer titles - especially team based and competitive-driven ones - a chat component is a community expectation. This helps players communicate moment-to-moment strategies, improves the sense of community, and aids in the feel of competition. That being said, there are considerations for how and when to implement chat, and some titles opt not to do it at all despite it being a desire from their community.

- Text Chat
 - *Global/Map/Lobby Chat*
 - Messages are visible to all players within the game or game session.
 - **Pros:** Facilitates broad communication and community interaction. Easy to implement and use.
 - **Cons:** Potential for spam and toxic behavior. Moderation can be challenging, especially with regards to reporting and GDPR compliance. This can often become a full time job for a community manager or support staff member.
 - *Team Chat*
 - Messages are only visible to players on the same team.
 - **Pros:** Enhances team coordination and strategy.
 - **Cons:** The moderation considerations are the same.
- Native Voice Chat
 - *Proximity Voice Chat*
 - Players can hear each other based on their in-game proximity.
 - **Pros:** Increases immersion and realism, dynamic and context-sensitive.
 - **Cons:** Can limit strategy if too far removed from the map. Requires significant engineering resources, and moderation considerations remain mostly the same as for text-based chat.
 - *Team Voice Chat*
 - Voice communication limited to team members.
 - **Pros:** Enhances team coordination and real-time strategy.
 - **Cons:** Requires significant engineering resources, and moderation considerations remain mostly the same as for text-based chat.
- Platform-Only Voice Chat
 - Integration with external chat platforms like Discord, or native console-based chat functionality.

- **Pros:** Often preferred by established gaming communities. Encourages building community outside of the game. Allows us to offboard moderation and compliance mainly to those external platforms with the existing infrastructure.
 - **Cons:** Dependency on third party tools.
- Emotes and Quick Chat
 - Predefined animations or expressions to convey messages non-verbally.
 - **Pros:** Universally understood and can transcend language barriers. Reduces the risk of toxic behaviors. Adds character and personality to the game.
 - **Cons:** Limited communication capabilities.

Moderation

Depending on which, if any, chat feature is implemented, there is a strong chance that a moderation system will also be a requirement. Even in some instances *without* chat in multiplayer games, there may still be benefit to implementing moderation tools like blocking, as certain players may still grief, troll or otherwise disrupt the gameplay experience for other players even *without* chat functionality. There are varying levels of moderation both in and out of game including:

- In-Game Reporting
 - Players can report others for toxic behavior directly in the game.
 - **Pros:** Requires human or AI moderator to vet, triage and action on reports.
 - **Cons:** In-game reporting systems can be misused, even as an additional avenue for griefing players and developers. Can be used for spam, and requires someone to actively monitor reports. Requires GDPR compliance.
- In-Game Support
 - Dedicated in-game support staff or GMs respond to reports or tickets. This is most commonly used in MMORPGs. Action can be taken directly against players in-game, if necessary.
 - **Pros:** Players can directly take action on toxic behavior where it occurs for immediate or quick results. Easy to verify misbehavior.
 - **Cons:** Requires dedicated support staff, and many specific UI/UX and backend engineering tasks to support.
- External Reporting
 - Dedicated out of game support staff or CMs respond to reports or tickets using a third party tool. Action can then be taken against players in-game, if necessary.
 - **Pros:** Doesn't require in-game tools and functionality for reporting.
 - **Cons:** Challenging to tie back to specific instances, players or accounts, and to verify the truth of the report compared to in-game reporting systems. Still requires external vetting, triage, response and actioning.
- Automated Reporting
 - AI can analyze in-game chat and detect toxic language, hate speech, and harassment. This includes filtering out offensive words, phrases, and even contextually harmful language.

- **Pros:** Doesn't require much additional, human support.
- **Cons:** Can miss less obvious cases of harassment and toxicity.
- **Blocking/Muting**
 - Players who have been reported multiple times for toxic behavior can be automatically muted by the system, or players can manually mute or block players to avoid being matched with them again in future games.
 - **Pros:** Simplest solution to deal with toxic players, especially if they are verbally aggressive. This can result in not hearing/seeing them in chat and/or no longer matching with them in future games.
 - **Cons:** Toxic players can potentially create new accounts to continue misbehavior. Requires some awareness of persistent accounts, if handled through in-game systems vs platform systems.
- **Reputation Systems**
 - Players are rated based on their behavior in games. A low reputation score can result in restrictions, such as longer matchmaking times or being matched with other low-reputation players.
 - **Pros:** Discourages toxic behavior and allows for the community to mostly self regulate bad actors.
 - **Cons:** Can be misused if enough bad actors team up to manipulate the system. Can potentially result in people incorrectly being restricted.
- **Bans/Suspensions**
 - For minor infractions, temporary bans or suspensions can be implemented to give players time to reflect on their behavior. For repeated or severe toxic behavior, permanent bans can be enforced to remove the player from the community.
 - **Pros:** Permanently removes toxic players from the community and reinforces behavior expectations.
 - **Cons:** Some banned/suspended players will continue harassment campaigns on additional third party platforms.
- **Community Moderation**
 - Trusted players can be given moderation privileges to help manage toxic behavior in real-time.
 - **Pros:** Encourages the community to self moderate. Potentially rewards the best among the community with additional prestige and clout.
 - **Cons:** Varying results when non professionals take on this role. It can create a sense of hierarchy or preferential treatment within the community, or create friction between community members who do or do not have the status.

Leaderboards

If a product is competitive in nature it carries the expectation of visible status, clout and prestige. For games where challenge and skill is the focus, the majority of players are motivated by being better than other players within the space, and the recognition from others in their status and ability.

- Global
 - Displays the top players from around the world based on a specific metric, such as rank, score, or wins.
 - **Pros:** Encourages intense competition among the top players. Offers a platform for players to gain widespread recognition.
 - **Cons:** New or casual players may feel demotivated if they are too far behind the top players. Can become a target for cheaters trying to reach the top, which can harm the leaderboard's integrity.
- Local/Regional
 - Displays the top players within a specific region or locality.
 - **Pros:** Players expect fair competition with others in the same region, considering factors like server latency. Encourages local community to grow, compete and connect. More attainable for players as they compete within a smaller pool.
 - **Cons:** Less prestigious compared to global leaderboards. Can divide the community if regions are too isolated from one another.
- Seasonal/Time-Limited
 - Resets the leaderboard periodically (e.g., every month or season), allowing players to compete for the top spot during a specific period.
 - **Pros:** Keeps the competition fresh and gives new players a chance to compete on equal footing. Players are motivated to play regularly to maintain or improve their ranking.
 - **Cons:** Constant resets may lead to burnout for players who feel pressured to keep competing. Players who fall behind early in the season might disengage until the next reset.
- Skill-Based
 - Ranks players based on their skill level, often using a rating system (e.g., ELO, MMR).
 - **Pros:** Ensures players are matched against others of similar skill, leading to more balanced games. Players can see tangible improvement as their skill rating increases.
 - **Cons:** New players might find it difficult to break into higher tiers if they start at a lower skill rating.
- Metric or Mode-Based
 - Instead of a single leaderboard, multiple leaderboards are created based on different metrics, such as kill/death ratio, objectives completed, or support actions.
 - **Pros:** Encourages a wider range of playstyles by recognizing achievements beyond just winning. Players may feel less pressured to focus solely on winning, reducing toxic behavior.
 - **Cons:** Having too many leaderboards can dilute the significance of each one. Players might be confused about which leaderboard is the most important or relevant.

Rank

In addition to leaderboards, players may expect other ways of identifying rank within a multiplayer game. This can be visual signifiers, titles, rewards, unlockables, leveling or other methods.

- ELO or MMR-Based
- Tiered/Divisional
- Seasonal
- Progression-Based
- Hidden/Soft
- Role-Based
- Point-Based
- Title-Based
- Visual Indicators

News Hub

For a multiplayer game with a focus on community-driven updates, events, initiatives, showcases, rewards, etc - it may be helpful to integrate an in-game splash screen that serves as a landing page for all the latest and most relevant news. It can be a great tool for highlighting key community members and keeping players engaged and informed.

- Static Splash Screen
 - Display a static image or a series of images with text on the game's loading screen or immediately after the game launches. This could include news about updates, events, and promotions.
 - **Pros:** Since it appears on launch, all players will see the news, ensuring that important updates are communicated. This is a straightforward method that requires minimal development resources.
 - **Cons:** Players cannot interact with the content, which limits the ability to provide detailed information. This can be difficult to update without pushing a full game/build update.
- Dynamic Carousel
 - A dynamic splash screen featuring a carousel of news updates that rotates automatically or can be manually navigated by players. This could include images, text, and short video clips.
 - **Pros:** Can capture attention with multimedia content and highlight different news items in a single screen. Can potentially be updated independently of the game by using a web-based JSON solution.
 - **Cons:** Complex dynamic elements could impact loading times or game performance, particularly on lower-end systems. If too many items are featured, players may feel overwhelmed and miss key updates.

- Interactive News Hub
 - Create a dedicated, interactive news hub accessible from the main menu. Players can browse through various news items, updates, patch notes, and events.
 - **Pros:** Players can choose when to access the news, reducing the feeling of interruption. Allows for more in-depth updates, including images, videos, and links to further information.
 - **Cons:** Players expect the news hub to be updated frequently with the latest information. Players have to actively check the hub, which means some might miss important updates. More complex to implement and maintain compared to a static splash screen.
- Pop-Up Notifications
 - When players log in or enter the main menu, a pop-up window appears with the latest news, which they must acknowledge or dismiss before proceeding.
 - **Pros:** Ensures that critical information is seen by all players. Can be used for urgent announcements, such as server maintenance or hotfixes.
 - **Cons:** Players might quickly dismiss pop-ups without reading them, especially if they become a frequent occurrence.
- In-Game Mail System
 - Players receive in-game messages or mail with news updates, which they can read at their convenience. This system often includes a notification icon or alert when new messages are available.
 - **Pros:** Allows for more personalized and detailed information to be delivered directly to players. Players can keep track of past updates or important announcements.
 - **Cons:** Players might not check their in-game mail immediately, potentially missing time-sensitive information. Requires a system to manage, archive, and organize messages effectively.

Product Features

Platforms

- Steam
- Key Resellers
- EGS/GOG
- Console

Analytics/Telemetry

Analytics and telemetry are essential for live ops games with community driven development. It allows for developers to receive true, objective data and statistics about things like player churn and retention, demographics, CCU, DAU, popular features/weapons/loadouts/vehicles, etc. By relying exclusively on a survey-based approach, we will inherently skew our data by only

involving positively engaged players in our data analyses. If there is no analytics integrated at launch, the game will always suffer from incomplete and inaccurate data.

In-Game Analytics

- Embedded SDKs
 - Using third-party Software Development Kits (SDKs) integrated directly into the game code. These SDKs are provided by analytics platforms (e.g., Unity Analytics, GameAnalytics, Firebase) and typically offer pre-built functionality for tracking events, user behavior, and other metrics.
 - **Pros:** SDKs are usually well-documented and easy to integrate into the game. They offer ready-to-use features like event tracking, user segmentation, and real-time analytics. Many SDKs support multiple platforms (PC, mobile, console), simplifying multi-platform development and allowing a complete picture of the playerbase, especially if crossplay is implemented. These solutions also tend to be GDPR compliant out of the box, and have easy to use dashboards for exporting and reviewing data.
 - **Cons:** While SDKs are easy to use, they may not provide the level of customization needed for specific game features. Integrating multiple SDKs can increase the game's size and impact performance. Some SDK solutions require payment or subscriptions to store data, which is a requirement of GDPR, and many of these costs are not clear upfront.
- Custom Solutions
 - Developing a custom analytics and telemetry system tailored specifically to the game's needs. This involves creating bespoke tools and databases for collecting and analyzing game data.
 - **Pros:** Developers have complete control over data collection, storage, and processing, allowing for greater flexibility and security. Tailored to the game's specific requirements, allowing for more detailed and relevant data collection.
 - **Cons:** Building a custom solution is resource-intensive, requiring significant development time and expertise. Higher initial costs due to development and infrastructure needs, sometimes difficult to scale. Most teams don't have in depth knowledge of compliance needs that SDK services offer natively.
- Platform Analytics
 - Using available dashboards, metrics and analytics available on each of the platforms, such as Steam, Xbox (and Azure/PlayFab), Sony or Nintendo.
 - **Pros:** Native, and does not require any additional integration or support. Dashboards and data collection are done in the web-based backend, and have exporting capabilities.
 - **Cons:** This data is often extremely limited, as well as inaccurate. Useful for data like crashes and purchases/installs, but less so for actual in-game player behaviors. Likewise, the data is not 1:1 between platforms, which makes creating a true complete picture of the userbase difficult, inaccurate, and manual-labor intensive.

- Third Party
 - Using dedicated third-party platforms (like Google Analytics, Mixpanel, or Amplitude) that aren't specifically designed for games but can be adapted for game telemetry.
 - **Pros:** These platforms often provide advanced data visualization, segmentation, and analysis features. Many of these platforms offer APIs and documentation that simplify integration.
 - **Cons:** These platforms may not have features tailored to gaming, requiring extra work to adapt them for game telemetry. Using third-party platforms may raise concerns about data privacy and compliance with regulations like GDPR.
- Server-Side Logging
 - Events and telemetry data are sent from the game client to a backend server, where they are logged and stored for analysis. This can be done using APIs or other communication protocols.
 - **Pros:** Server-side logging can handle large volumes of data and is easier to scale as the player base grows. All data is stored on the server, making it easier to manage, analyze, and secure.
 - **Cons:** Data transmission from client to server can introduce latency, particularly in regions with poor connectivity. Requires a robust backend infrastructure and expertise in server management. Handling sensitive player data on the server requires strict security measures.
- Client-Side Logging
 - Data is logged directly on the player's device (client) and can be sent to a server in batches or processed locally. This approach can be used for offline games or where immediate data transmission isn't necessary.
 - **Pros:** Can operate even when the player is offline, with data being uploaded later when a connection is available. Reduces the amount of data that needs to be transmitted to the server in real-time, lowering server demands.
 - **Cons:** There's a risk of losing data if the player uninstalls the game or if there's a crash before data is transmitted. Storing data locally on a player's device can expose it to tampering or hacking.

GDPR Compliance

GDPR (General Data Protection Regulation) compliance is crucial for video game developers who handle personal data of players within the European Union (EU). Non-compliance can result in significant fines and damage to the company's reputation. Developers must have a legitimate reason for collecting and processing personal data. This could be based on consent, contract necessity, legitimate interest, or legal obligation. If relying on consent, it must be freely given, specific, informed, and unambiguous. Players should be able to opt-in to data collection, and consent should be easily revocable.

Any developers who participate in data collection that can be tied back to an individual user or account, must adhere to GDPR laws. This is true in both cases of telemetry/analytics as well as live issues support collection.

- Considerations and Requirements
 - *Right to Access*
 - Players have the right to access their personal data and know how it is being used.
 - *Right to Rectification*
 - Players can request corrections to inaccurate or incomplete personal data.
 - *Right to Erasure*
 - Players can request the deletion of their personal data, also known as the "right to be forgotten."
 - *Right to Data Portability*
 - Players have the right to request their data in a structured, commonly used format and transfer it to another service.
 - *Right to Object*
 - Players can object to the processing of their data, particularly in cases of direct marketing.
 - *Data Retention*
 - Define and adhere to data retention periods, ensuring that personal data is not kept longer than necessary for the purposes for which it was collected.
 - Implement processes to securely delete data when it is no longer needed or upon user request.
 - *Privacy Policy and Transparency*
 - The game's privacy policy must clearly explain what data is collected, how it's used, who it's shared with, and how long it's retained. It should be easily accessible to players.
 - Players should be informed about data collection practices at the point of data collection, with clear explanations of why the data is needed.
 - *Parental Consent*
 - For games targeted at children or likely to be used by them, implement mechanisms to obtain parental consent before collecting data from minors.
- Data Collection Types
 - *User Accounts*
 - When players create accounts, the collection of email addresses, usernames, and other personal information must comply with GDPR.
 - *In-Game Analytics*
 - Data collected for player behavior analysis must be done with a clear lawful basis, and players should be informed about what data is being collected.
 - *Marketing*
 - If collecting data for marketing purposes, this should be clearly communicated, and the data should not be used for any other purpose without additional consent.

- *Telemetry*
 - When collecting telemetry data to improve game performance or balance, developers should avoid collecting unnecessary personal information and ensure that data is anonymized where possible.

Live Issues Pipeline

For live ops games that are community driven and enter an open beta or early access period, especially multiplayer, server-based games - it is essential to have a robust, easy to use, effective live issues pipeline for reporting, ingesting, reproducing, triaging, fixing, and communicating to users.

Ideally, this reporting system integrates seamlessly with the development project management tool in order to quickly and effectively assign tickets to QA for investigation, and engineers for addressing and merging into upcoming releases.

It is better still if the bulk of the communication is handled directly within the same tool, as it drastically reduces friction and community manager/customer support time, while simultaneously improving the quality of information received.

Keep in mind, continuous tracking and updating require dedicated resources, which can strain smaller teams.

- *Considerations*
 - *Accessibility*
 - Ensure players can easily report issues directly from the game, through official forums, or via social media.
 - *Clear Guidelines*
 - Provide players with clear instructions on how to report issues, including what information to include (e.g., platform, game version, steps to reproduce).
 - *Automated vs. Manual Reports*
 - Decide whether to rely more on automated issue detection (e.g., crash reports) or manual player reports.
 - *Custom Dashboards*
 - Create custom dashboards to monitor the status of live issues, their impact, and resolution progress. Use tags, labels, or categories to help quickly identify and sort issues by type, severity, or area of the game affected.
 - *Player-Facing Dashboards*
 - In some instances, it is helpful to have a player facing dashboard (such as a trello board or other roadmap tool) to easily communicate to players about known issues and their status. This helps players feel as though they are heard and that their complaints are being prioritized and

addressed. This is usually paired with a “known issues” thread or pinned channel.

- *Patch Notes*
 - Regularly publish detailed patch notes that explain what issues have been addressed and any known remaining issues. This is especially important if issues around balance are addressed!
- **Methods**
 - *In-Game Reporting Tools*
 - Allow players to report issues directly from the game interface, often with a built-in form.
 - **Pros:** Provide immediate context and reduce the effort required for players to report issues.
 - **Cons:** Can be limited in detail or require players to interrupt their gaming experience.
 - *Support Tickets*
 - Use a ticketing system where players submit detailed reports that are then logged and tracked.
 - **Pros:** Offer a structured way to gather detailed information and track the resolution process.
 - **Cons:** May overwhelm support teams with a high volume of reports if not properly managed.
 - *Community Forums*
 - Encourage players to report and discuss issues on official forums, where community managers can monitor and escalate issues.
 - **Pros:** Forums and social media allow for broader community involvement and feedback.
 - **Cons:** Reports might be scattered and require significant moderation and aggregation.
 - *Social Media Monitoring*
 - Track social media platforms and gaming forums for issue reports that might not come through official channels.
 - **Pros:** Issues are reported to where developers already have an active presence and resource, and can easily follow up for further info as needed.
 - **Cons:** Easy to miss complaints, requires manual overhead to collect and ingest. Can skew perception both of developers and players about how pervasive issues are. Can lead to public toxic behaviors from players.

Current Scope

AFU Beta Feature List:

- New Bikes - colors/decals - 3 Sizes
- Character presets with individual bikes

- SFX & Implementation
- Dynamic Levels (Trains, Cars, etc)
- All Weapons from Vertical Slice
- All Maps from Vertical Slice

Wishlist:

- Police / Mobs / NPCs
- Emotes

Bitcake Feature List:

- UI/UX for Public lobby room list
 - Basic UI Framework for menus - entire click through chain from main menu to getting into a match
 - Ease of updating assets to UI Framework - set up in a way where replacing temporary assets is simple
 - Automatic input shift from mouse to controller depending on detected input
 - New scripts and assets contained in separate location or folder to properly separate UI systems and make it easier to differentiate
- Join functionality by picking a public room
- Host able to select room properties before game start (such as the map)
- Room Code generation for simpler joining
- Ability to create rooms - Able to mark public or private
- Implementing Steam friends API that both shares room code and let friends join