> The Industry And The Market

In 2018, total video game sales exceeded 43.4 billion US dollars.[1] 9,050 games were released on the exceptionally popular game distribution platform “Steam” the same year(the highest amount since 2004).[2] We need to explore and become part of this rapidly growing vertical now more than ever.
However, great care and planning must go into making sure our products both support and help the developer and the gamer. CloudFlare Workers already provides an amazing platform to deploy code and only a few touch ups(game engine integration,a functioning game marketplace,and accessibility upgrades for small-scale developers) are needed in order to make it suitable for this industry.
Our immediate plans should be to further research both sides of the “coin”(players and developers) and pinpoint what the issues they have with the current system of play. This can be done by:

- Actively attending game developer conferences(for both indie and large scale developers) like GDC and IndieCade and learning what can be improved from existing frameworks and actual developers.
- Sponsoring game development hackathons for indie developers and taking note/asking them about the problems they had with their tools during the event.
- Going to events like E3 and talking to players on what they feel could be improved about how they get their games and the quality of said games.
- Finally, a team should be assigned to research social media outlets like Twitter and popular gaming forums to find criticism about how games are developed and deployed.

> The Product (Goals & Vision)

Clouflare Workers’ strength in easily deploying serverless code is undeniable. All we need to do is capitalize on these strengths and add features that make it more hospital for all types of developers and gamers.
Right now, full attention should be given into adding a game engine to our current server architecture. At its core, a game engine is a premade set of tools and environment that help developers add features and cosmetics to their games quickly without losing track of quality. Popular examples of engines are Godot,Unity, and Unreal Engine. ***We need to make our engine especially easy to use and accessible*** as our first target developer audience will not be the AAA game studios like Valve but the smaller but successful indie-game teams/single developers like StudioMDHR(Makers of

CupHead).
Bigger firms already have entire teams dedicated to deployment and targeting them first would be a mistake(we need to quickly break in the market and grow enough for them to notice us). The n*ext* ***necessary feature is a game store/marketplace analogous to Steam*****.** We need to be able give developers a place to monetize their games and an easy way for the users to interface with said games. The power of distributed computing and CloudFlare Worker's easy deployment will be on full display here as they will be able to play resource intensive 3D games online.Finally, it is necessary that we ***continuously add features to the engine*** and have a very close knit relationship with them. Trends in gaming come and may go (like VR) but we should be ever ready to support developers no matter how big their dreams!This tight knit relationship with our developers (by engaging them via social media, events and actively listening to their feedback) will help us make adjustments and updates to our system during deployment of our product and small test runs of the system that allow small groups people to develop and push games into the marketplace at hackathon like events will show us the adjustments we need to make right before we launch. Finally, during development of this product we need to quickly break development into steps/prototypes (e.g getting the engine done,the server-side code to break up computing resources for players,etc.) and look back and constantly check for mistakes and flaws**(getting a product out that is clunky and horribly designed will do us terrible damage as this industry is very fickle when it comes to innovation).**

> Risks and Challenges

- Our single biggest risk is being associated with failed game streaming services like Stadia. People have very negative ideas about streaming games through the "cloud" → Solution: the streaming as mentioned above should still be subscription based but when users buy from our platform they "own" the game(i.e can still download offline executables from out site).
- The next biggest risk is not capturing the attention of developers. If there are no exclusive games on our platform, then there is no incentive to switch over to our system → Solution: keeping them as an active stakeholder(in terms of responding to their feedback and concerns) during and post development of the system and making a system that makes them the **top priority.**
- **We can easily measure our success by reviewing how many developers join our platform and how many players actively use our streaming service and marketplace. Our main goal in the earlier stages will be *growth* and later on we will shift to measuring market share(our main competitor not being Stadia like services but Steam like marketplaces).**