Tasnim Tallman

Adding Flare to the Cloud Gaming Industry

Cloudflare has the opportunity to break into the gaming sector, the largest sector in the entertainment industry, by capitalizing on the advantages that serverless computing provides. Though AWS, GCP, and Azure already deliver serverless products for game developers, Cloudflare Workers can utilize its distinct qualifications and learn from the successes of its competitors to deliver a higher quality, more distinguished product.

There are three main aspects to cloud gaming that CFW can decide to implement based on potential ROI and ease of implementation: peer to peer gaming (communication between client devices), progressive downloading (allowing a game to be played as it partially downloads), and game streaming (direct video streaming from the player's device to those of spectators). CFW can play to the advantages of cloud gaming which includes a removal of wait time before playing, the ability to continuously upgrade games without the limitations of hardware, and the reduced potential of lag by utilizing their edge network. It has also already been proven that due to Cloudflare's usage of a V8 engine, isolates, and thousands of distributed machines around the world, its speeds far outperform its fiercest competitors in the market: Lambda and Lambda@Edge.

By marketing itself as the fastest yet most affordable cloud gaming tool on the market, Cloudflare can set itself apart from its competitors while implementing the following key necessities to its platform. First, CF Workers must make sure that its software is completely compatible with all workloads and platforms that a game developer may want to use. Video game programmers primarily code in C, C++. Java, HTML5, JavaScript, and sometimes an assembly language. These programmers most often write code that adds onto or customizes what has already been built in game engines (a software environment that eases the development of video games to avoid redundancies and save time). Game developers also have to create different versions of their code to ensure compatibility across all platforms and consoles. So, we have to make sure that CF can support each platform, language, and type of gameplay (i.e. FPS or MOBA). Second, CF Workers needs to prioritize the speed and reliability of its network for gaming. The most important things game developers require are quick frame rates and low latencies, so CF should provide an extremely high SLA level along with a very low latency (which can be executed via a combination of CF's quick edge network along with strategies like lag compensation, time dilation, interpolation, and extrapolation). Third, CF Workers can provide security and protection, which are already Cloudflare's specialties.. Specifically, Workers can provide protection from DDOS attacks, bot detection, and cheating and fraud prevention. Lastly, Cloudflare can continue to implement its auto scaling feature to limit costs for its clients, therefore preserving its affordability.

Cloudflare Workers can gain an additional edge against competition by leveraging their gaming initiatives to offer additional perks to users. Some features they can provide include cross-platform play, serverless leaderboards, and quick matchmaking. They should also support large databases for keeping track of player records, inventory systems, and game data. Something that CF can adapt from AWS game tech's platform is their accessible game analytics that developers can use along with their reliance on machine learning to direct liveops, push notifications, speech recognition, behavior prediction, and fraud detection.

A couple more features to look into are AR/VR gaming support, real-time translations between international players, and built-in project templates to aid in the development process.

Conducting further user and market research to understand the needs of the consumer and the trends of the market would strengthen the preliminary findings of this research. After segmenting the market to find target customers, actual customer feedback via focus groups can help focus our priorities. We can also conduct more research on the competitive landscape to discover our weaknesses and find ways to become unique. The data and knowledge we have gained from our other similar ventures can help predict the potential value of this idea. After developing proof of concepts and administering enough research, we can move on to implementation.

The product must undergo extensive testing and meet every one of our requirements before it is ready for production. For instance, through user testing, our goal should be to prove that our software is intuitive and capable of meeting users' every need. The product should meet all functionality metrics and match the UI/UX design of Cloudflare's other offerings. The software should have no bugs, and maintain stability against crashing. To bolster the product's integrity in the eyes of our consumers, marketing material should accurately portray the product, the sales team must be trained to provide informative demos, and the product should be ready to be onboarded.

Once the product has been released, we can monitor a number of KPIs as a measurement of success. From a business standpoint, we can measure the product's revenue, number of contracts, number of potential deals, and retention or churn rates. These metrics allow us to measure how effective our marketing campaign is in attracting game developers, and how well our product is at keeping them. From a product perspective, we can look at how often a client uses our software and which features they use most, how many customer service calls/messages we received, and how long it took for the client to be onboarded. To increase synergy, we should consult with the marketing, sales, and finance departments and analyze their own KPIs to see how Cloudflare Workers for Gaming can improve as a whole.

Part of developing a successful product is to also consider the possible risks that could lead to the product's failure. As with any new undertaking, the company should prepare for the risk of not being able to successfully execute the product as planned. Even if the product does launch, we should be prepared for the possibility that the product does not meet performance expectations. The product may also fail to produce forecasted returns due to market risk. It could also be impacted by competition, changes in the economy, new regulation, or bad release timing.

With the proper measures, Cloudflare can undertake the necessary initiatives to mitigate the product's risks. For one, it is better to release a product with limited functionality but reliable stability, rather than vice versa. The company can adhere to this approach by alloting ample time towards development and allowing flexibility for deadlines to ensure functionality before release. Finally, we can assure the product's quality by consistently employing various tests at each sprint such as regression testing, unit testing, and integration testing to make sure all code works in conjunction. Beta testing would also allow us to learn faster before releasing to the general public.