Luis Chelala
Lchel003@ucr.edu
Prospective Product Management Intern

Cloudflare Workers for Gaming

Learning Market Needs

Current market needs a reliable way to manage increased server load for tasks. Gaming platforms run into issues of server overload/capacity issues. Many gaming servers are not able to dynamically connect to their origin server, which places issues for them. The gaming industry relies on being quick, effective, and secure. With Workers, now that you can be serverless and deploy what you need to make your game run effectively with running into hard issues amongst the games developments original physical server. We need to enforce this in order to win the market. With the gaming industry growing rapidly, they need ways to increase game efficiency and reliability.

In order to learn more about the market needs to specifically tailor these great features, some research needs to be done. To understand the market better we need to understand how the gaming industry manages their game server. This is a primary focus for us as we need to understand what points of how a worker will actually benefit them. For primary research on the market needs we will do some quantitative analysis on how long it is taking the games to cache data that is not the game itself and how many failures they come across. This will allow us to understand better how Workers will be able to benefit game developers, in order to position Workers as the correct option for them. Workers are going to help greatly in improving performance of the game, so seeing the numbers will be beneficial. We will look at the market as well and what the current server they use are and what issues they might be having. We need to also learn more about how the market implements security measures for how they cache certain data. With Workers allowing for code to be implemented separately from the game this could be something we would market on in terms of security. For example, saving user data serverless to be more secure than the current way the market saves user data.

Product Changes or additions

In order to appeal to the gaming industry there needs to be the ability to directly manage code integrated into the game from the worker, but also see how the changes and implementation will affect the rest of the current build of the game. This could be done through viewing the server stacks and the caches that build up the game. For example if we are working with a game that is cross platform and using a worker to handle that data, we want to be able to directly see how the worker is managing that data and how it is affecting the code of the game server. Another addition should be able to handle offline scenarios. This could be done using a fetch api to intercept network requests from the game. This will be important as a technique to server resources from the cache when someone using the game is offline. If we can optimize the worker implementation so every asset needed in the game is cached then a user could go completely offline for example.

Methods for improving quality pre-release

We want to release a product as soon as possible in order to start gaining this portion of the gaming market. The first step would be doing voluntary surveys or tests with a gaming platform that would give feedback to how the workers integrate with their system. The vision would be to have enough feedback on how the worker is increasing efficiency and the strength of their game. We would then collect requirements and do feasibility testing on those requirements. Before releasing, we need to also make the game logic run effectively with the worker. Since application logic is very important for game developers, the worker needs to be quick in response time or low latency. We can track the load times with an API for example and then go from there in improving our product quality for the game devs. If we are able to ensure these are met plus the additions to the product, we will have a strong first release that greatly appeals to the needs of the game industry and game developers.

Goals to measure success

Once a version is released in the market, we must keep track of its viability and growth. The first way to measure the success of the product would be to have the Worker cache more than the existing caching mechanisms enabled in the game environment. If this is the case then we know that we are successful in our release. Another goal should be to have the worker impact the experience of game loading greatly to where it feels more fluid. Asking certain questions on how the experience feels will not have a specific metric since it is subjective, but will give an insight if there is a substantial difference in the performance. If there is a feeling of difference then that would help understand if we are successful in the release of the product.

Risks

A potential risk would be debugging and error handling. Once pushed to the workers, it is hard to know what could be incorrect with the code and or data because as the game developer you will not be handling what happens in the backend for that worker. This could lead to unsuccessful launches for a game. Another risk is that the workers are dependent on the V8 engine. Any issues that happen with the engine can force all the workers to stop. Let’s say for online gaming this is a great risk. Any downtime that occurs can hurt the reliability of the online ecosystem. If this were to happen it would damage the reputation of the Cloudflare Workers for Gaming reputation as well, making it difficult to get accepted by other developers.