Hypothesis: Game developers suffer for their art and rather spend their time working directly on game assets/mechanics than infra. They are adverse to introducing dependencies from non-game developers, being very protective of their game, and so the product will have to be heavily personalised to, PoCed, and endorsed by influential personalities in the gaming industry.

lucy@u.nus.edu

1. Identify key market segments for deeper personalisation.
   - by platform: mobile (ios, android), PC (windows, mac)
   - by genre: MOBA, MMORPG, RTS, Open-world RPGs etc
   - by heavy-resources (stuff to be cached for better load times) e.g. graphical assets, game worldstate

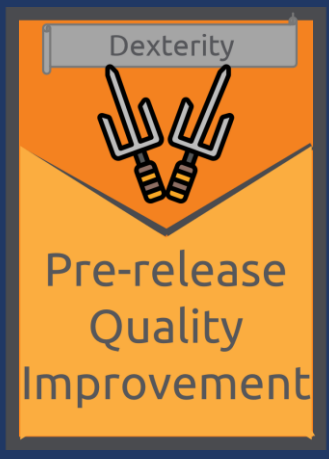
2. Identify game dev personalities
   - by company size (indie vs established players), are there dedicated developers working on improving game load times? What's the existing solution to the problem, and what's lacking?

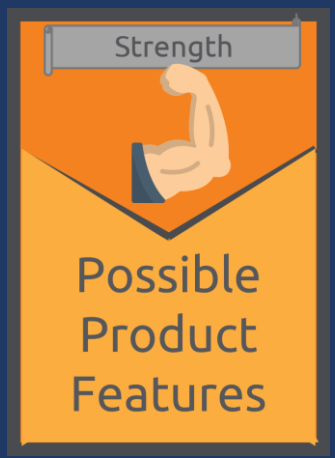
3. Identify genre of games that suffer from the worst loading times between gameplay. Eg GTA5, Witcher 3 famous for long loading screens. Listen to players to discover other existing problems to be solved.

1. Single Library for multiplatform resource fetching: if a developer owns a game title that supports multiple platforms, e.g. Allows for multiplayer across multiple operating systems, allow for assets to load from a single cache point.

2. Shared cache for multiple games: e.g. if a game developer owns a game series with multiple titles, allow them to load shared assets from a single cache point for multiple game projects.

3. Work with sales and pricing team to see how to smartly bundle other essential services with this. E.g. Classic Cloudflare DDOS protection to mitigate unnecessary requests from the cache.





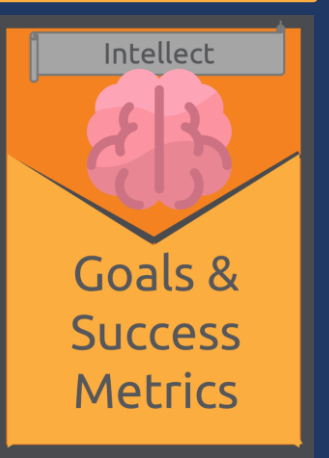
1. Closed Beta testing with Game developers and game professionals e.g. Game QA testing teams, Professional gamers, critics.

2. Stress test on a same game but with different configurations:
   e.g. With different graphical resolutions and settings enabled, , For multiplayer, with different numbers of players, across different geographical regions

3. Feedback gathering with game developers given closed-beta access to the service API to gather areas for improvement. Could be naming conventions of API, documentation quality, ease of testing, etc.

Goals
1. Get service adopted by well-known gaming houses or applied to established title known for long load times /problem that the Cloudflare Worker for Gaming solves. Helps with industry product recognition.

Success Metrics:
1. Focus sessions/Surveys/Feedback from game developers and QA testers. % increase in load times before&after use of this service. Influence of product – is the gaming community hyped about it?
2. If used, % of functions in the API library that gets used. Measures if provided service is useful on a whole, highlights redundant API fns.



Possible Risks:

1. Product is not well received by game developers
   - Actively reach out for collaboration with specific game titles/indie developers to push a PoC to the market to quantify and qualify performance bonus when using the Cloudflare service.

2. Product does not accommodate certain types/formats of assets.
   - Possibly collaborate with game developers to prioritise the redesign of certain resource formats.
   - On Cloudflare's side, communicate clearly through documentation supported resources and release design guidelines to developers.