

Optimise docker

<https://gist.github.com/noelbundick/6922d26667616e2ba5c3aff59f0824cd>

Succès

- Rocket + Okapi permet Swagger facilement
- tokio::join pour rouler TCP + HTTP threads
- Static mut:
 - pub static CORALINE: Lazy<Mutex> = Lazy::new(|| Mutex::new(Coraline::default()));
 - pub static mut DB: Option<redis::Connection> = None;
 - Pour ne pas avoir de conditions de mutex entre plusieurs threads (set_queue_handler -> queue_task = impossible)
- Queue_task run in background + list + abort
 - pub static QUEUES: Lazy<Mutex> = Lazy::new(|| Mutex::new(Queues::default()));
 - tasks: HashMap<String, JoinHandle<()>> ,
 - queues.tasks.remove(&lobby2.id).abort();
- Redis Index -> Redis SortedSet + sets etc
 - lobby = hash
 - lobby.players = list
 - lobby_ids = set
 - queue_mmr = sorted set par queue -> peut chercher range pour trouver un lobby dans le bon mmr et la bonne queue
 - game.players + player.game -> double references help searching and dont cost much

Bugs

- Bug copier/coller player.lobby dans get_game()
 - Pas de stacktrace
 - Pas facile de debug Rust, du moins sur Windows
 - workarounds avec des permissions étranges
- Bug accès api Rocket
 - <network_mode: host> empechait de bien fonctionner

Pains

- Arc.id: Arc<Mutex>
 - client.get_id().lock().await;