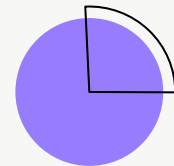




HearthStone Battleground

Conception d'un jeu en ligne international

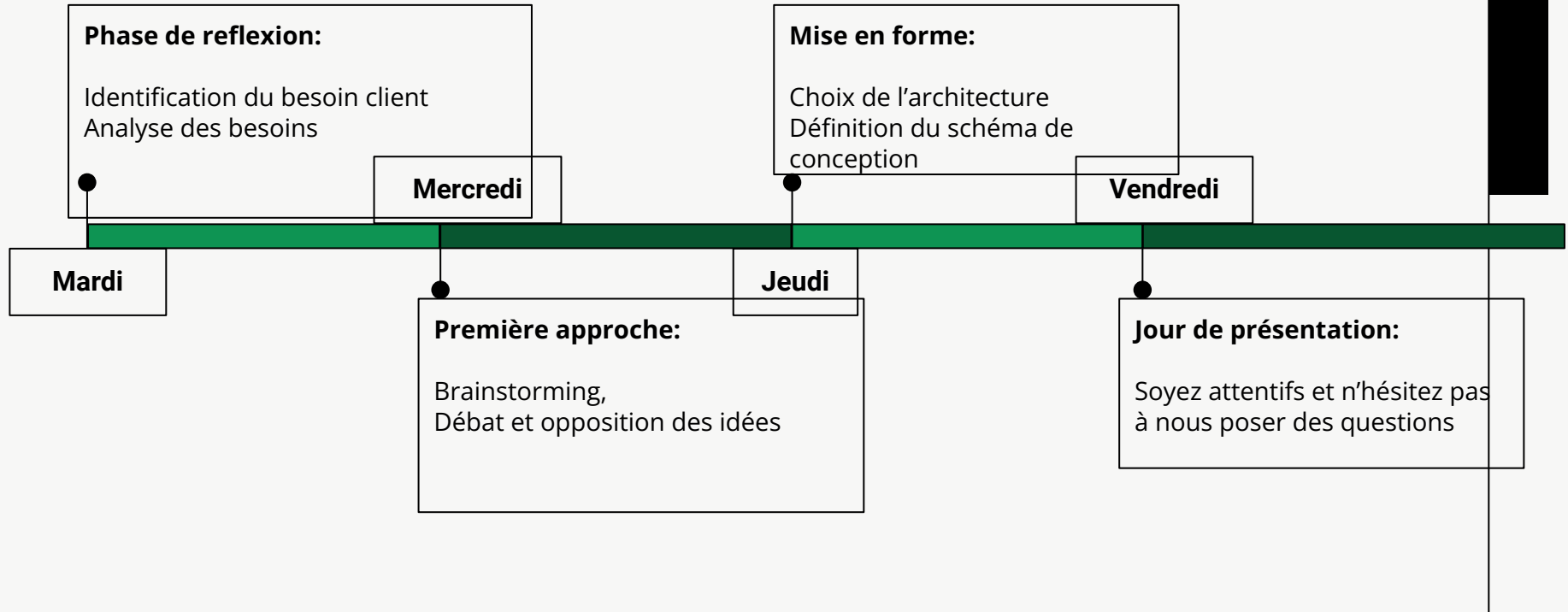


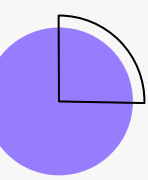
1

Introduction



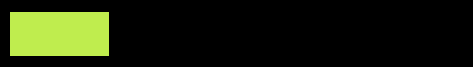
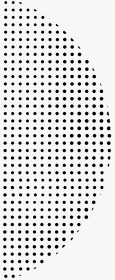
Les étapes de conception



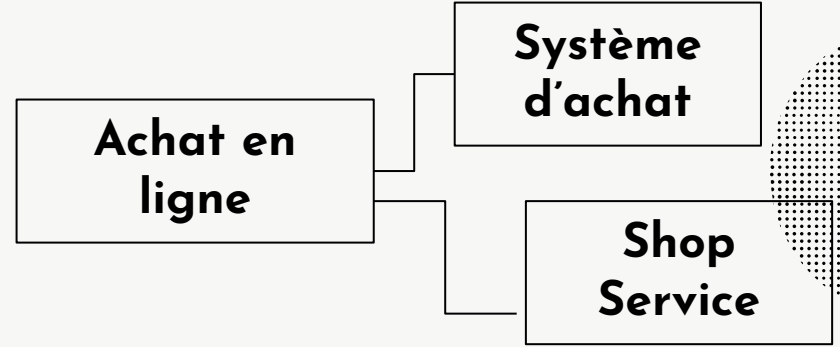



L'analyse des besoins

- **Jeu multijoueur**
- **Boutique**
- **Disponible partout**
- **Système de ladder**
- **Matchmaking**
- **Authentification**









Traduction des besoins clients

- Optimisations
- Gestion des régions
- Gestion des charges
- Sécurité
- Mises à jour & compatibilité
- Interface utilisateur & expérience de jeu
- Analyse de données & personnalisation
- Accessibilité
- Monétisation
- Feedback & support
- Monitoring
- Tests
- Déploiement & coûts



2

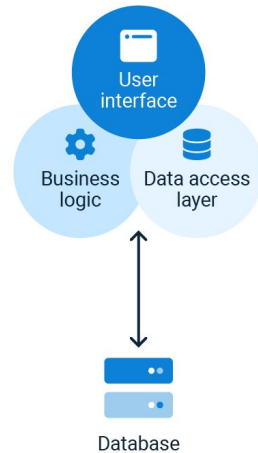
Conception



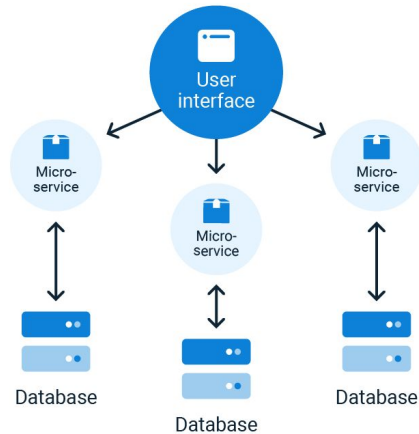
Le choix de l'architecture

Monolithic VS Microservice

Monolithic Architecture



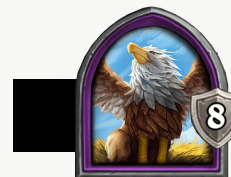
Microservice Architecture



Le choix de l'architecture

Tableau comparatif :

Critère	Architecture Microservices	Architecture Monolithique
Complexité Initiale et à Long Terme	Plus complexe initialement, mais offre une meilleure adaptabilité et facilité de maintenance à long terme.	Plus simple au départ, mais peut devenir plus complexe et moins flexible avec le temps.
Performance	Distribution efficace des ressources, améliorant les performances globales.	Performances initialement adéquates, mais peuvent diminuer avec la complexité croissante.
Résilience	Haute résilience; défaillance d'un service impacte peu les autres.	Moins résiliente ; une défaillance peut entraîner l'arrêt de l'ensemble du système.
Scalabilité	Facilement scalable, ajustement aisé des ressources.	Scalabilité limitée, changements d'échelle plus difficiles.
Maintenance	Plus aisée et modulaire, permet des mises à jour parallèles.	Peut devenir lourde avec l'augmentation de la taille et de la complexité du système.
Adaptabilité	Très adaptable aux changements technologiques et exigences du marché.	Moins adaptable, changements souvent lourds.
Extensibilité	Facile à étendre avec de nouvelles fonctionnalités.	L'extensibilité est souvent limitée, nécessitant parfois une refonte majeure.
Sécurité	Sécurisation fine et spécifique pour chaque service, mais gestion de la sécurité plus complexe.	Plus facile à sécuriser initialement, mais une faille peut affecter tout le système.
Coûts	Coûts initiaux plus élevés, mais potentiellement plus rentable à long terme.	Moins coûteux au départ, mais peut devenir plus cher à long terme, en particulier pour l'adaptation et la maintenance.



Le choix de l'architecture

Organisation du projet

**Combinaison de l'architecture
MicroService N-tiers et du
modèle MVC**



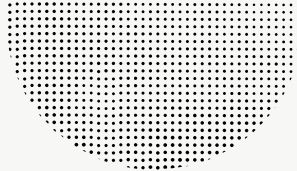
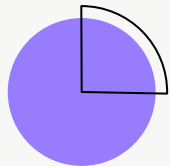
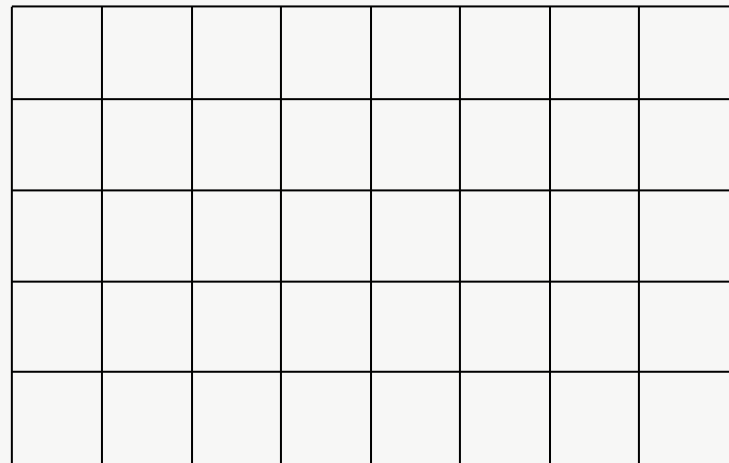
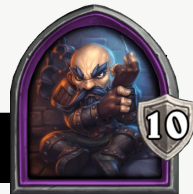
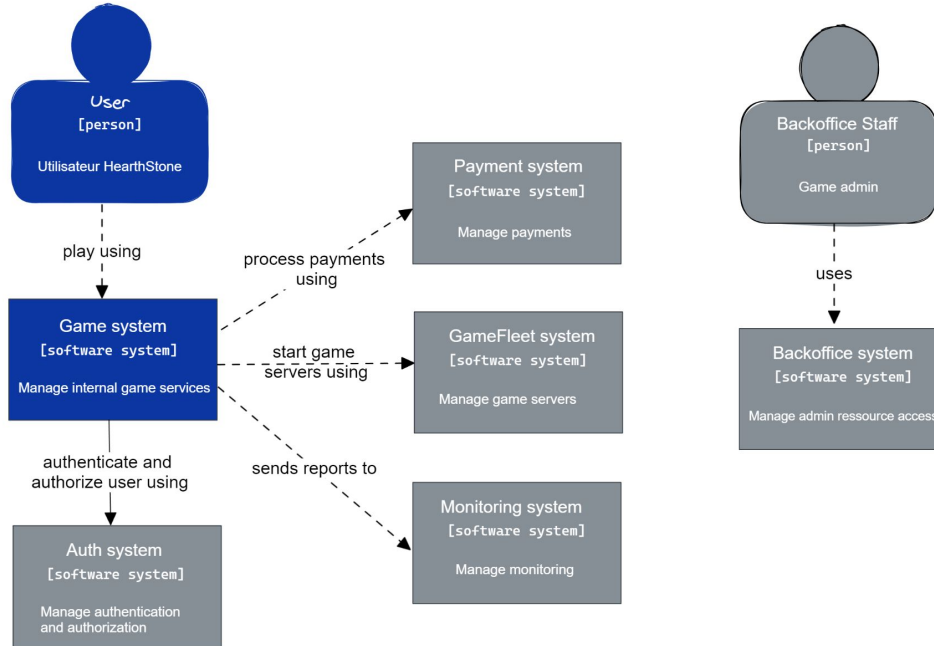


Schéma de conception !



Modèle C4

Contexte



Modèle C4

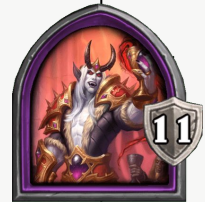
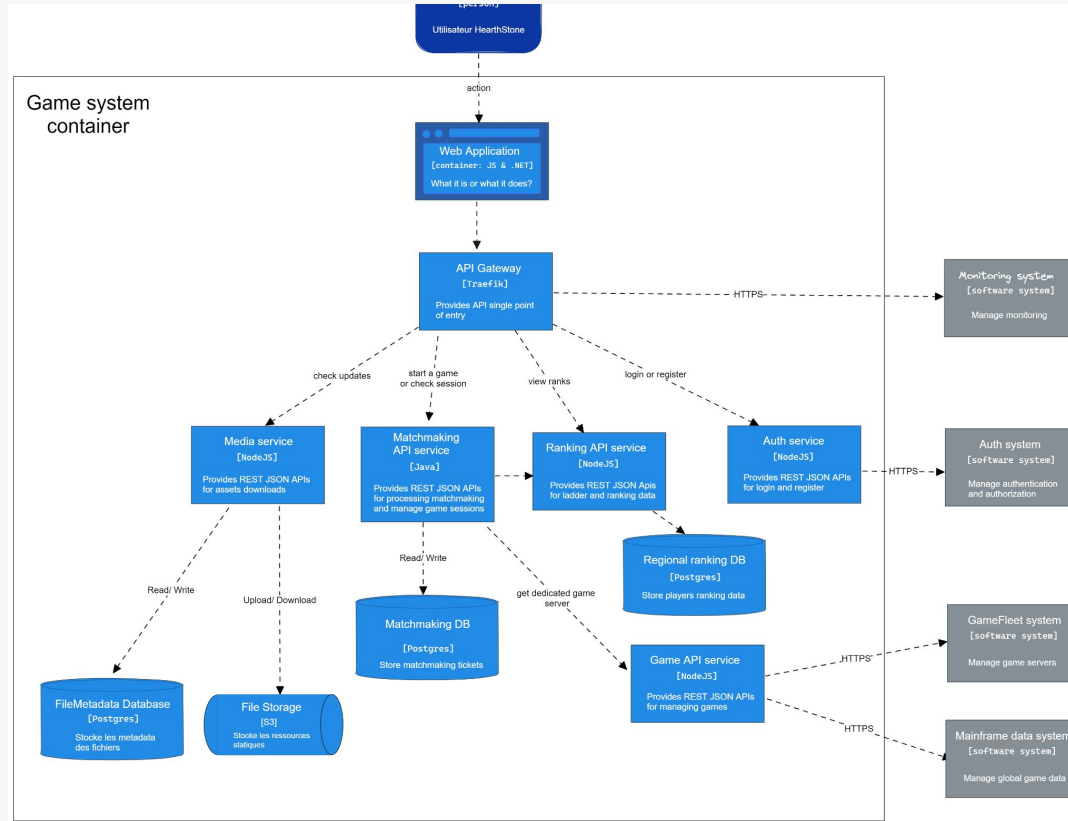


Diagramme de classe

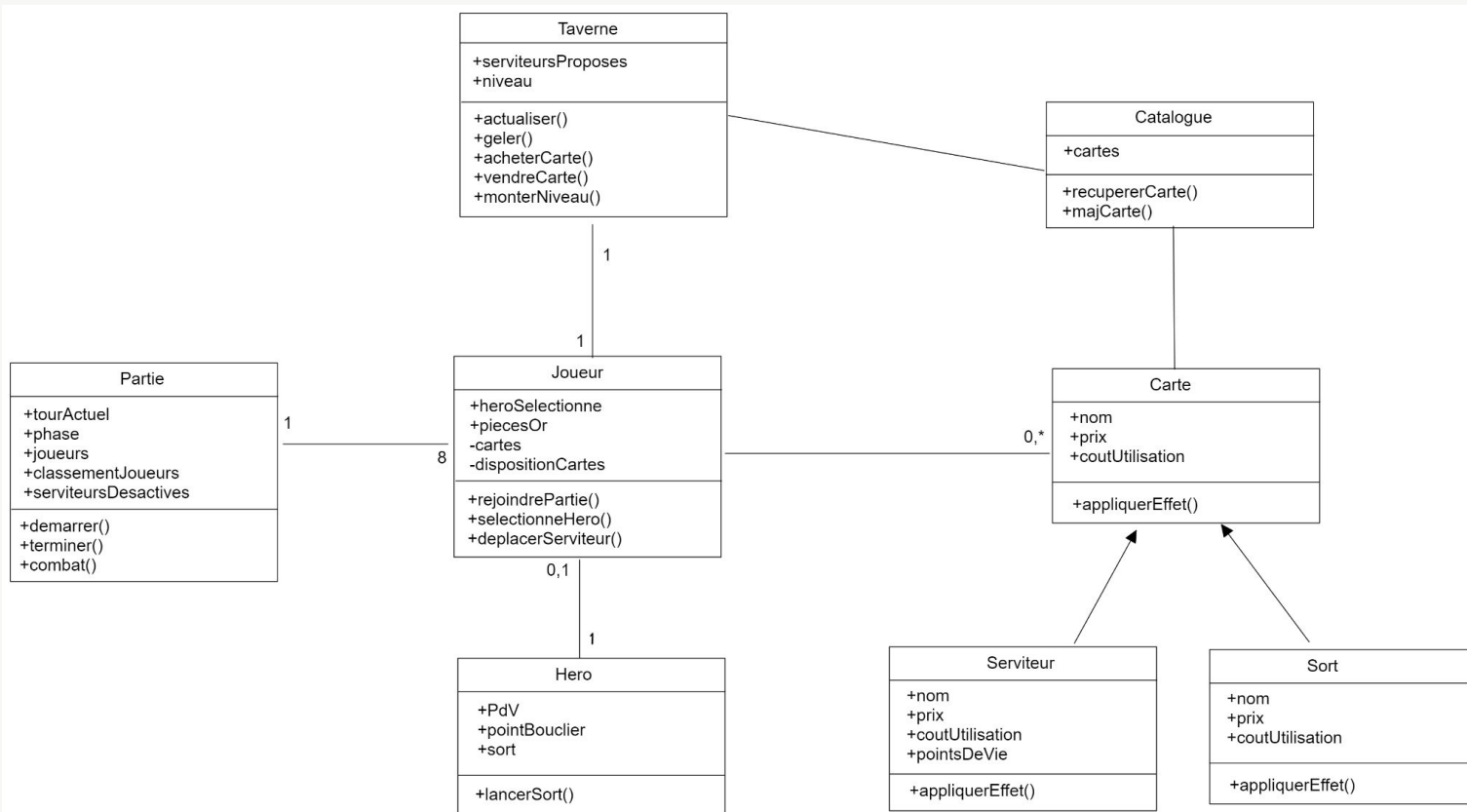
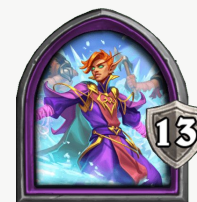
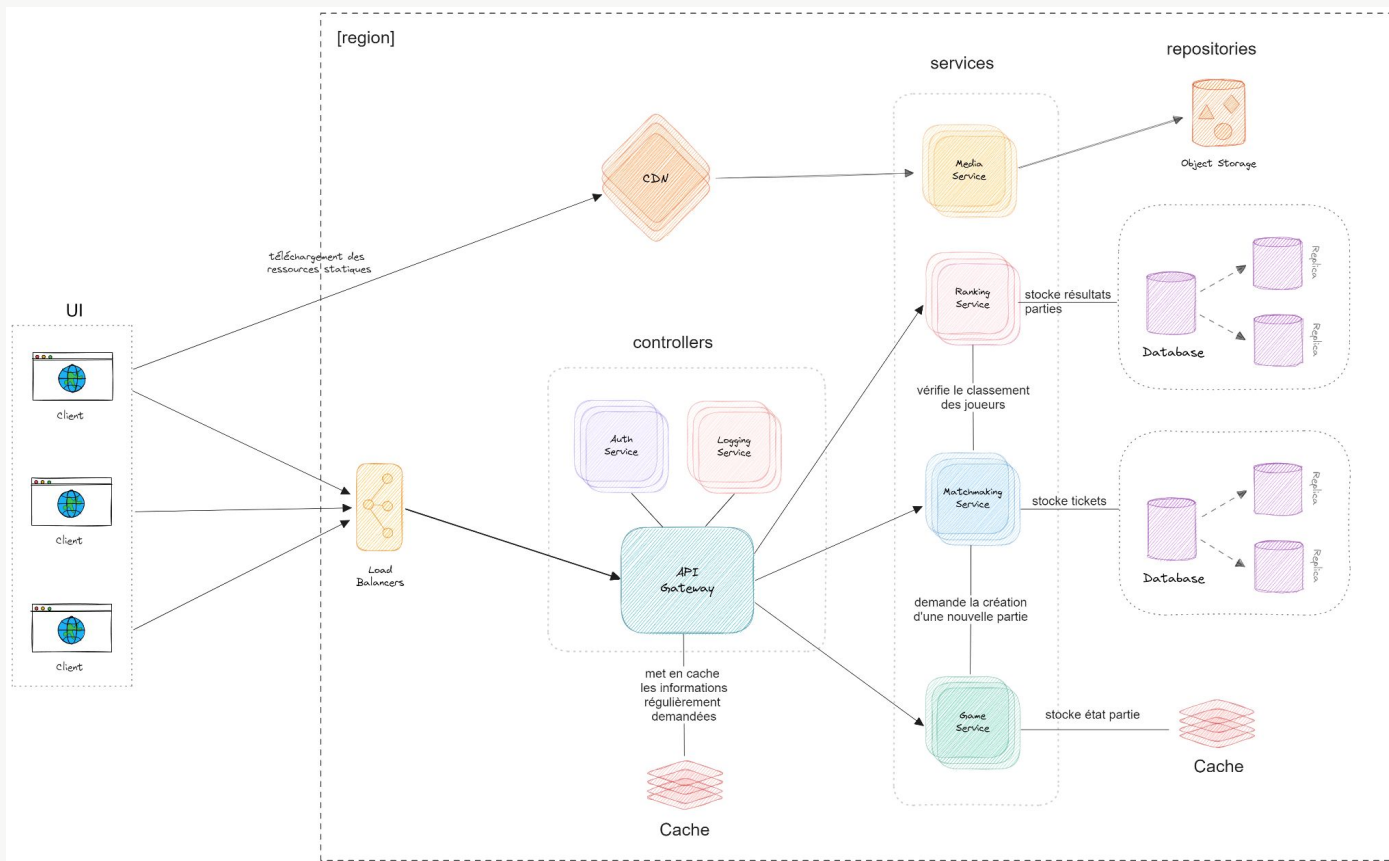


Diagramme de classe



3

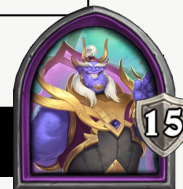
Choix des technologies



Le choix des technos

Le moteur de jeu Unity

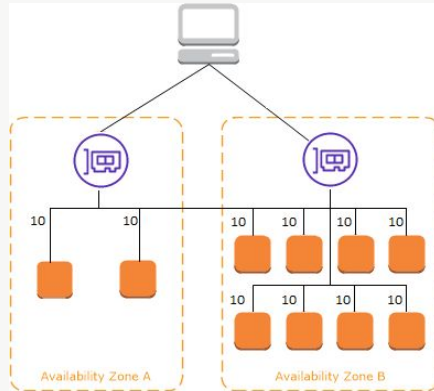
Caractéristiques	Unity	Unreal Engine	Godot
Langage de programmation	C# et javascript	C++, Blueprint	GScript
Licence et coût	Gratuit (personal) ou Abonnement Pro (1 877 €)	Gratuit, Abonnement (1 500 €)	Gratuit et Open source
Graphisme et rendu	Solides capacités graphiques	Rendu exceptionnelles, rendu ultra réaliste	Fonctionnalités graphiques 2D et 3D
Utilisations	Intuitive, grande communauté	Courbe d'apprentissage raide	Interface utilisateur simple
Flexibilité	Modulaire, riche écosystème (plugins)	Extensible avec des plugins C++	Extensible avec un système de plugin



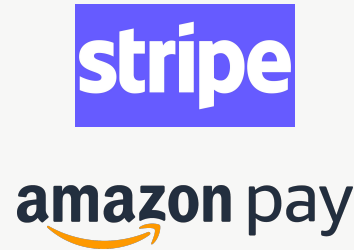
Le choix des technos

Autres fonctionnalités du système

● Le load balancer Traefik



● Système de paiement



● Réplication & Backup





HearthStone Battleground

Conception d'un jeu en ligne international

