



Pixel Conqueror

Notre mission, si toutefois nous l'acceptons, est d'organiser et développer un auto battle révolutionnaire !

About the team

Ressources

❓ [Réunion Client](#)

🧠 [Brainstorming](#)

💣 [C4](#)

📝 [ADR \(Architecture Decision Records\)](#)

🎨 [Domain Driven Design](#)

💵 [Budgetisation](#)



Réunion Client

Le concept:

Créer un jeu auto-battler. Le but est d'être le dernier survivant parmi 8 joueurs.

Déroulement d'une partie :

1. Choix des Héros :

Les joueurs choisissent un héros parmi une sélection aléatoire. Chaque héros a une capacité unique qui influence la stratégie.

2. Phase de recrutement.

Lors de la phase de recrutement les joueurs obtiennent un nombre d'or définit. Lors de la phase de recrutement les joueurs utilisent leurs or pour acheter des serviteurs (cartes) dans la taverne qui les aideront à améliorer leur armée. Ils peuvent aussi payer pour actualiser la liste de serviteur proposé. Ils peuvent aussi payer pour améliorer le niveau de la taverne et permettre de proposer de meilleur serviteur.

Le but est donc d'acheter, vendre, combiner des serviteurs pour améliorer son armée

3. Phase de combat

Après la phase de recrutement, les armées des joueurs s'affrontent automatiquement. Les serviteurs attaquent à tour de rôle, suivant des règles spécifiques.

Le perdant du combat perd des points de vie en fonction de l'armée encore en vie de son adversaire

Les phases de recrutement et de combat s'alternent jusqu'à ce qu'il ne reste qu'un seul joueur en vie.

Les besoins

Le client souhaite une application bureau et une application mobile.

Une application web sera nécessaire pour la partie marketing, présentation du jeu, achat, inscription etc.

Le jeu passera par un serveur backend (API possible) et le choix du stockage est laissé à notre équipe.

L'application doit pouvoir soutenir un **trafic élevé** (plusieurs milliers de joueurs connecté ensemble) **24/7**. La répartition des joueurs se fera sur plusieurs serveurs (au moins 3: EU, Asie, US).

Le jeu devra être traduit dans un **maximum de langue**.

Un système de match d'entraînement (PVE), normal (PVP) et classé (PVP avec classement) à mettre en place

Business Modèle

Une monnaie dans le jeu les “pierres runiques” qui s’achètent avec de l’argent réel.

Certains achats peuvent s’effectuer avec soit des pierres runiques soit de l’argent réel.

D’autres ne peuvent être obtenus qu’avec des pierres runiques.

Les achats disponibles sont :

- des skins pour les héros
- des skins pour le terrains (background du jeu)
- des emotes (animations).

- un pass de combat
- des goodies

Le passe de combat permet d'obtenir des skins de héros, des emotes et un avantage en jeu. L'avantage est d'avoir le choix entre quatre héros plutôt que deux en début de partie.



Brainstorming

Hébergeurs

Paiement

Catalogue

Moteurs Graphiques / Physiques

Systèmes de Gestion de Base de données

Monitoring et gestion des erreurs

Réunion de BrainStorming

Budgetisation



Hébergeurs

	AWS (Amazon Web Services)	Microsoft Azure	Google Cloud Platform	OVH	Alibaba Cloud
Présent EU, NA, ASIE	International	International	International	International	Majoritairement présent en Asie de l'est, quelques serveurs présents en Europe et Amérique du Nord
Coût	Tarification à l'heure	Tarification similaire à AWS, avec possibilité de payer en mont (sur base d'estimation)	Tarification compétitive, souvent moins chère pour certaines ressources par rapport à AWS ou Azure.	Tarification au mois	Tarification compétitive, particulièrement dans les régions asiatiques.
Avantages	- Évolutivité massive et capacité à gérer des charges de trafic élevées. - Large gamme de services et d'outils pour le développement, la	- Intégration étroite avec d'autres produits Microsoft, utile si vous utilisez déjà des outils Microsoft. - Bonnes capacités	- Hautes performances, particulièrement en matière de traitement des données et d'analytique. - Bonnes capacités		- Idéal pour une meilleure performance en Asie. - Large gamme de services cloud.

	AWS (Amazon Web Services)	Microsoft Azure	Google Cloud Platform	OVH	Alibaba Cloud
	<p>gestion de bases de données, l'analytique, etc.</p> <ul style="list-style-type: none"> - Bonne réputation en matière de fiabilité et de sécurité. 	<p>d'évolutivité et de disponibilité.</p> <ul style="list-style-type: none"> - sécurité et conformité. 	<p>d'évolutivité et options de personnalisation.</p> <ul style="list-style-type: none"> - Engagement fort en matière d'énergie renouvelable et de durabilité. 		<ul style="list-style-type: none"> - Bonne évolutivité et options de sécurité.



Paiement

Le paiement doit être accessible à l'**international**. Ne pas se limiter à l'euro (€).

	Paypal	Stripe	PaySafeCard	Cryptomonnaies	Google/Apple store
Disponibilité	202 Pays	46 pays	50 pays	Interdit dans beaucoup de pays : Chine, Qatar, Cameroun, Guyane, Gabon, etc.	Beauuuuuuuucoup de pays
RGPD friendly	👍	👍	(👍) ça a l'air ok	👍	👍
Devises	25	135	26	23642 cryptoactifs	
Frais / Taxes	2,9% + 0,35c (transaction nationale) // entre 0,50 & 2% (internationale)	1,5 % + 0,25 € par transactions	3€ par mois (à partir de 2 mois d'utilisation)	Frais de gaz	30% de frais de transactiont mais 70% vont dans nos poches #riche 💰
Montant max/transactions	60k\$ par transaction		50€ par transactions		



Catalogue

- Cartes
- Héros
 - Nom
 - Capacités
 - Capacités spéciales
- Skins
 - skin de héros
 - skin de terrain
- Events
 - Skin thématique (selon les saisons ou des événements esport)
- Goodies/produits dérivés (Pull, mugs, peluche, figurine)



Moteurs Graphiques / Physiques

	Unity	Unreal Engine	Godot	CryEngine	GameMaker
2D	✓	⚠	✓	✗	✓
3D	✓	✓	✓	✓	⚠
Prix (licence commerciale)	Gratuit jusqu'à 100k\$ / année	Gratuit jusqu'à 1M\$ / année (puis 5% royalties)	Free	Gratuit jusqu'à 5k\$ / année (puis 5% royalties)	79.99\$ - 99,99\$ / mois
Langages	C#	Blueprint / C++	GDScript, C++, C#, Python, Nim et D	C++, C# et Lua	Drag n' Drop & GML
Type (A, AA, AAA)	AAA	AAA	AA-AAA	AAA	A
Affinité ✨					

- ✓ : Support complet
- ⚠ : Support partiel ou limité
- ✗ : Aucun support



Systèmes de Gestion de Base de données

	Oracle	MariaDB	MongoDb	MySQL	PostgreSQL	Redis
Avantages	<ul style="list-style-type: none">- Haute performance et fiabilité pour les grandes entreprises.- Fonctionnalités avancées de sécurité et de gestion des données.- Support étendu pour les transactions complexes et les grandes bases de données.	<ul style="list-style-type: none">- Open source et gratuit, avec une communauté active.- Compatible avec MySQL, facile à adopter pour ceux qui connaissent MySQL.- Bonnes performances et fonctionnalités pour la plupart des applications web.	<ul style="list-style-type: none">- Base de données NoSQL, idéale pour gérer des données non structurées ou changeantes.- Haute scalabilité et flexibilité dans la gestion des données.- Bonne performance pour les applications en temps réel et les grandes bases de données.	<ul style="list-style-type: none">- Très populaire, largement utilisé dans le développement web.- Open source avec une grande communauté et beaucoup de ressources.- Bon équilibre entre fonctionnalités, performance et facilité d'utilisation.	<ul style="list-style-type: none">- Open Source et Gratuit : Accessible et modifiable, avec une large communauté de support.- Supporte des requêtes SQL complexes, des types de données personnalisés, et des fonctions de stockage.- Fiable et sécurisé	<ul style="list-style-type: none">- Haute Performance : Stockage en mémoire pour des opérations rapides, idéal pour le caching et les sessions- Supporte des structures de données comme les listes, ensembles, et hash.- Facile à scaler et à utiliser dans des architectures distribuées.- Offre des options

	Oracle	MariaDB	MongoDb	MySQL	PostgreSQL	Redis
						pour la persistance des données malgré le stockage en mémoire.
Inconvénients	<ul style="list-style-type: none"> - Coût élevé, moins adapté aux startups ou aux petits projets. - Peut être complexe à configurer et à maintenir. - Moins flexible pour les changements rapides ou les environnements agiles. 	<ul style="list-style-type: none"> - Moins de fonctionnalités avancées par rapport à Oracle. - Peut nécessiter des optimisations pour des charges de travail très lourdes. 	<ul style="list-style-type: none"> - Moins adapté pour les applications nécessitant des transactions complexes et des relations strictes. - La conception de la base de données peut être un défi pour ceux habitués aux SGBD relationnels 	<ul style="list-style-type: none"> - Moins de fonctionnalités avancées pour la gestion de très grandes bases de données. - Certaines limitations en termes de scalabilité par rapport à des solutions comme Oracle. 	<ul style="list-style-type: none"> - Complexé à configurer pour les débutants - Peut être moins performant que certains autres SGBD pour des charges de travail extrêmement lourdes. 	<ul style="list-style-type: none"> - Limitation de la taille des données par la mémoire du serveur - Moins adapté pour les requêtes complexes ou les relations entre les données.

Liste de base de données :

Base de Données des Utilisateurs

Pour stocker : Utilisateurs, Profils, Statistiques, Amis, Paramètres.

- **Choix Suggéré** : MySQL ou MariaDB

- **Choisie:** Postgres
- **Pourquoi :** Ces bases de données relationnelles sont idéales pour gérer les données structurées des utilisateurs, telles que les profils, les statistiques et les paramètres. Elles offrent une bonne performance pour les opérations de lecture et d'écriture fréquentes, tout en assurant l'intégrité des données et la gestion des relations.

Base de Données du jeu :

Pour stocker : Héros, Serviteurs, Types de Serviteurs, Capacités, Historique des Parties.

- **Choix Suggéré :** MongoDB
- **Choisie :** MongoDB
- **Pourquoi :** MongoDB, en tant que base de données NoSQL, est bien adaptée pour gérer les données dynamiques et potentiellement non structurées du jeu, comme les informations sur les héros et les serviteurs. Sa flexibilité dans la gestion des schémas et sa capacité à gérer de grandes quantités de données rapidement sont des atouts majeurs.

Base de données de Matchmaking :

Pour stocker File d'Attente, Parties en Cours, Résultats, Classements.

- **Choix Suggéré :** Redis avec persistance des données
- **Choisie :** Redis avec persistance des données en MongoDB
- **Pourquoi :** Le matchmaking nécessite des opérations rapides et souvent temporaires. Une base de données en mémoire comme Redis peut gérer efficacement les files d'attente et les appariements en temps réel avec une latence minimale.

Base de données Boutique :

Pour stocker Articles, Transactions, Historique des Achats, Promotions.

- **Choix Suggéré** : MySQL ou MariaDB
- **Choisie** : Postgres
- **Pourquoi** : La gestion des transactions, des historiques d'achats et des articles de la boutique nécessite une base de données relationnelle pour assurer l'intégrité transactionnelle et la fiabilité.

Base de données Récompenses et Progressions

Pour stocker: Missions, Récompenses, Progressions des Utilisateurs, Succès.

- **Choix Suggéré** : MySQL ou PostgreSQL
- **Choisie** : Mongo
- **Pourquoi** : Ces bases de données sont efficaces pour gérer les données relationnelles complexes et structurées, telles que les progrès des utilisateurs, les missions et les récompenses, tout en offrant de bonnes performances et une fiabilité élevée.

Base de Données de jeu :

Pour Stocker : Serviteurs Disponibles, Niveaux de Taverne, Serviteurs par Niveau.

- **Choix Suggéré** : MongoDB

- **Pourquoi** : La flexibilité de MongoDB est avantageuse pour gérer les différents types de serveurs et leurs attributs, qui peuvent varier considérablement et nécessiter des mises à jour fréquentes



Monitoring et gestion des erreurs

- Grafana & prometheus pour le monitoring des ressources utilisées
-



Réunion de BrainStorming

Problématiques

- Choix du moteur pour la réalisation du jeu
- Choix d'une solution d'hébergement (Web, Base de donnée et Jeu)
- Choix du système de gestion de base de donnée et répartition.
- Choix des technologies (développement et organisation des tâches), avec explications
- Choix du système de paiement (international)
- Servers de test interne et public (PBE)
- Internationalisation
- Structure (monolithique ou micro service)

Solutions

Le jeu sera disponible en téléchargement via le site web pour la version bureau (Windows & Mac)

Choix du moteur graphique → **GODOT** :

- Gratuit
- Open Source
 - Possibilité de fork/ajouter des fonctionnalités au moteur si besoin
- Prise en compte de la 2D
- C#, C++
- Affinité de l'équipe avec le moteur

Choix de la solution d'hébergement → **Serveurs dédiés chez Amazon Web Service** :

- Meilleurs qualité/prix du marché

- SAV performant et techniciens dédiés
- Solution de backup et redondance
- Présence internationale
- Choix de serveurs dédiés (plus cher à la location mais pas de frais/requête)
- Service de CDN pour le site web
- L'équipe est à l'aise avec ce provider

Il faudra prévoir 3 datas center (1 serveur principal, 1 serveur de secours, 1 serveur bêta) par continent (Europe, United State, et Asie

Choix du système de gestion de base de données et répartition → **multiple base de données relationnelle, non relationnelle et cache type Redis basés chez AWS**

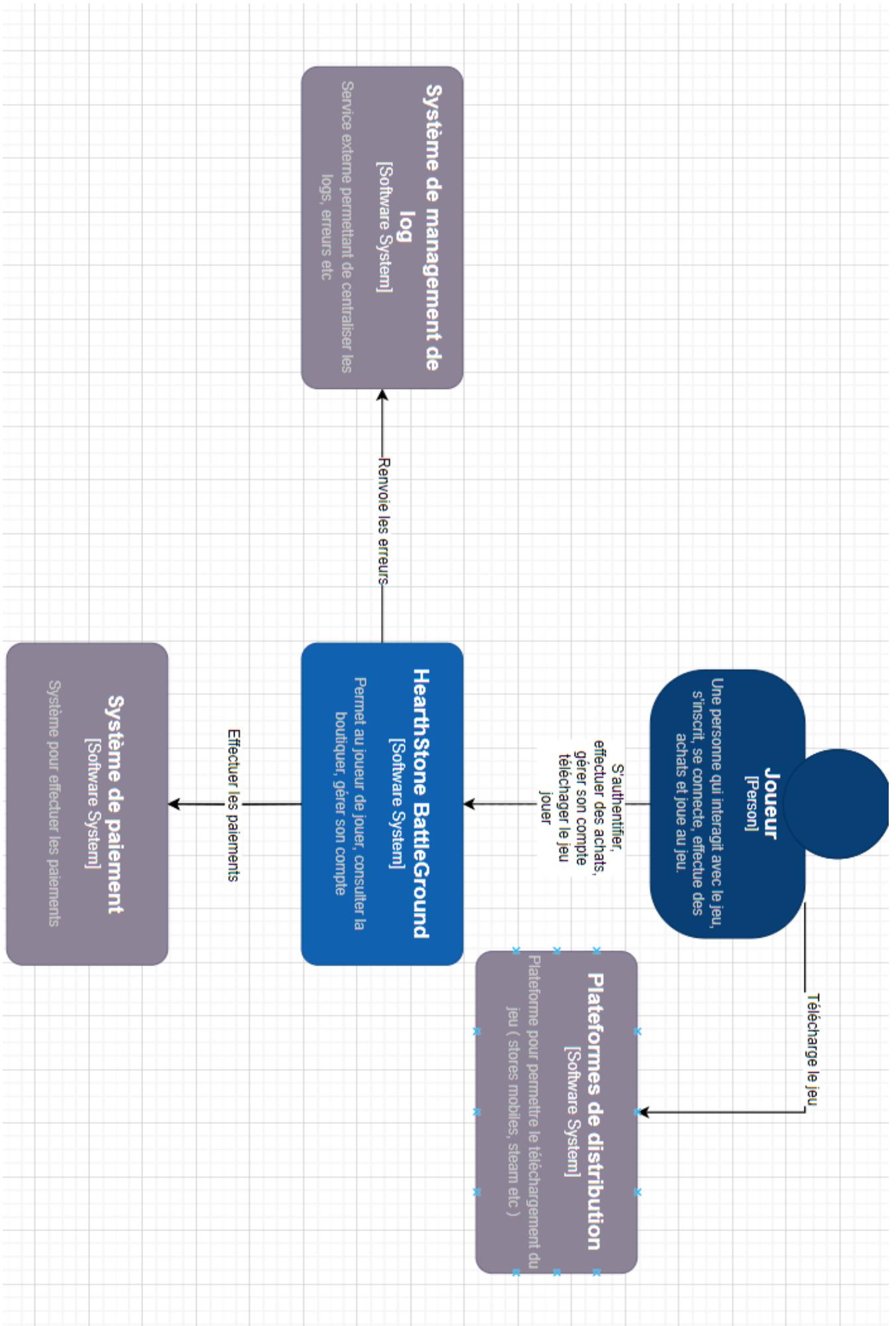
Systèmes de Gestion de Base de données

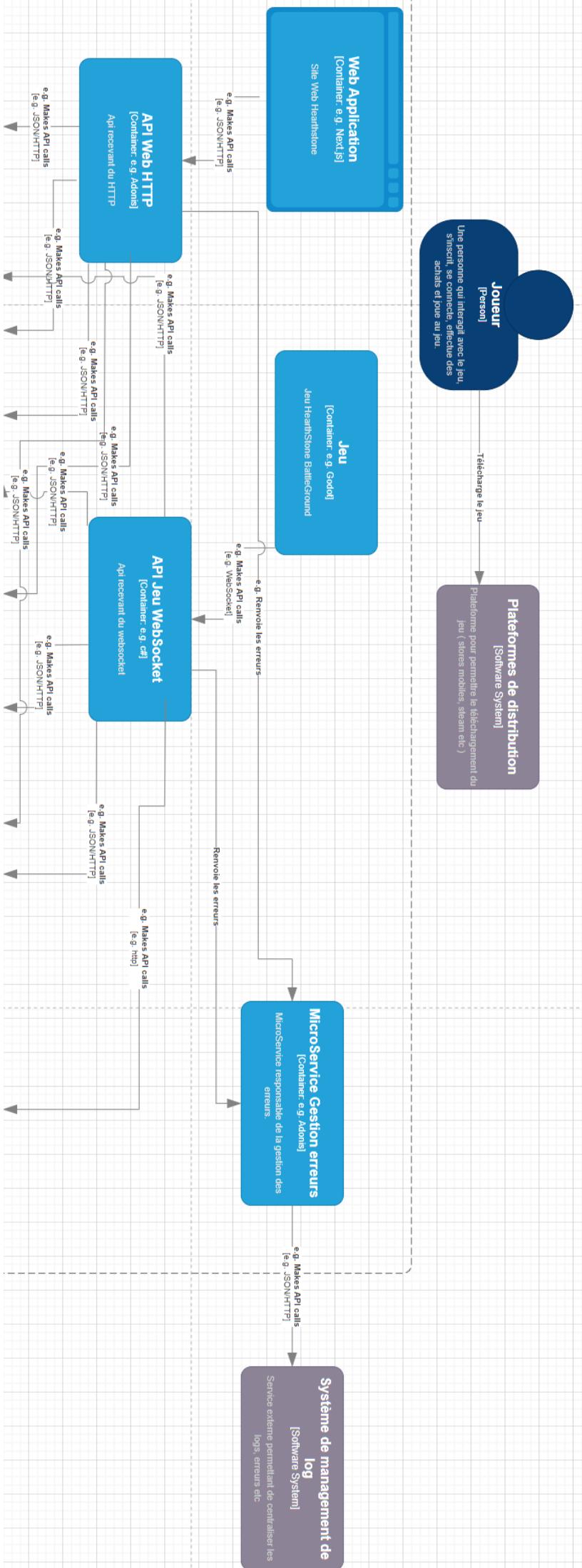
- Centralisation des données
- Réduction des coûts
- Sécurité des bases de données et maintenance de l'infra physique assurée par AWS

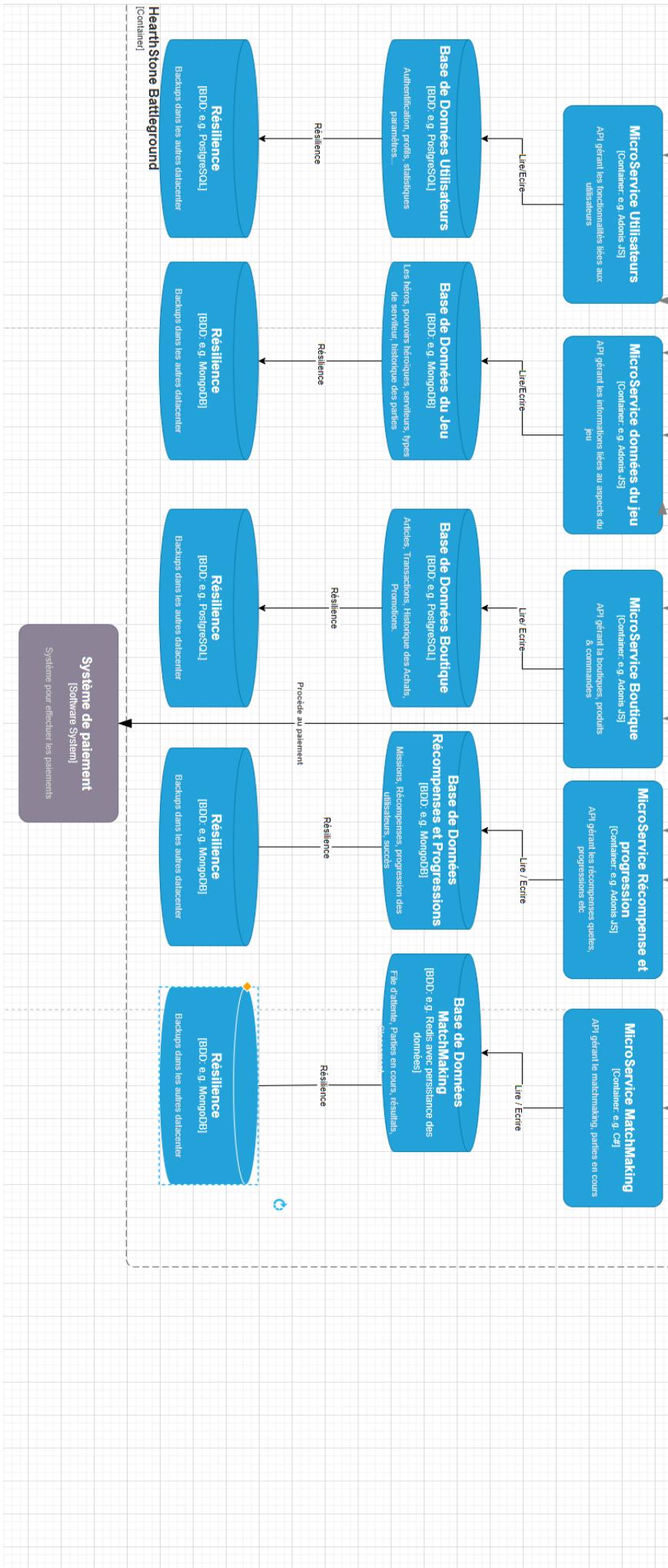
Choix des technologies :

- **Application web**
 - Next.js (front)
 - Permet de gérer un CRUD simplement (gestion des comptes)
 - Permet découper le code en composants
 - Rendu côté serveur
 - Moins lourd pour le client
 - Meilleur référencement
 - Adonis.js (backend)
 - Framework Node.js performant
 - Communauté active

- Framework et packages officiels activement maintenu par la core team
- Adapté à la monté en charge
- **Jeu (bureau + mobile)**
 - Godot (moteur graphique)
 - Gratuit
 - Adapte pour les jeux 2D
 - Open Source
 - Propose de bons outils pour aider au développement du jeu
 - Permet de compiler vers plusieurs plateformes (Bureau, mobile, console, etc.)
 - C# (backend)
 - Langage compilé, performant
 - Adapté au projet
 - A fait ses preuves
 - Maitrisé par l'équipe
- **Systèmes de paiement**
 - Stripe et Paypal pour la diversité des pays accessibles, et le nombre de devises autorisées
 - Apple/Google Store pour les achats sur le jeu mobile, imposé par les deux stores en question, des coûts supplémentaires sont à prévoir. Reste néanmoins simple à utiliser.
- i18n (traduction) : Traduction complète (via un outil communautaire comme weblate ou via des traducteurs professionnels)
- Déploiement
 - Kubernetes
 - Docker
 - CI / CD pour déploiement automatique sur les différents environnements (dév, test, preprod, prod)









ADR (Architecture Decision Records)

Pour le site web

Titre : Utilisation de Next.js & Adonis pour notre interface web

Date : 13 déc. 2023

Statut : Accepté

Contexte: Pour notre tout nouveau jeu (desktop & mobile), nous souhaitons créer une page internet afin que les utilisateurs puissent s'inscrire et se connecter via une page internet. Avec la naissance de notre projet et la nécessité d'une interface utilisateur réactive, interactive et qui a de l'impact, il est essentiel de choisir des technologies efficaces et durables dans le temps.

Décision: Nous avons décidé d'utiliser Next.js & AdonisJS pour le développement de notre site internet. Next.js est un framework gratuit et open source s'appuyant sur la bibliothèque JavaScript React, et donc gère la partie du front-end. AdonisJS est un cadre back-end qui aide à créer des applications web axées sur les données. Avec ce framework, il est possible de gérer les requêtes HTTP, interroger la base de données, authentifier les utilisateurs et télécharger des fichiers.

Conséquences:

Positives:

- Permet de gérer un CRUD simplement (gestion des comptes)
- Permet de découper le code en composants
- Rendu côté serveur
 - Moins lourd pour le client
 - Meilleur référencement

Négatives:

-
-
-

Références:

- Documentation officielle de Node : <https://nextjs.org/>
- Documentation officielle de Adonis : <https://adonisjs.com/>

Pour le jeu (desktop + mobile)

Titre : Utilisation Godot comme moteur graphique

Date : 13 déc. 2023

Statut : Accepté

Contexte: Pour notre tout nouveau jeu (desktop & mobile), basé sur du combat automatique nous devions choisir un moteur graphique qui correspondait le plus à nos besoins.

Décision: Nous avons décidé d'utiliser Godot comme moteur graphique. ce dernier est un moteur de jeu multiplateforme, c'est-à-dire un logiciel permettant de créer des jeux vidéo qui est compatible avec différents systèmes d'exploitation.

Conséquences:

Positives:

- Son utilisation est gratuite
- Adapté pour la 2D
- Open Source
- Propose de bons outils pour aider au développement du jeu

Négatives:

-
-
-

Références:

- Documentation officielle de Godot : <https://docs.godotengine.org/fr/4.x/index.html>

Pour le jeu (desktop + mobile) #2

Titre : Utilisation C# comme langage de développement back-end

Date : 13 déc. 2023

Statut : Accepté

Contexte: Pour notre tout nouveau jeu (desktop & mobile), nous avons besoin d'un langage de développement backend qui correspond parfaitement à nos besoins.

Décision: Le choix de C# a été unanime. C'est un langage de programmation orienté objet, fortement typé, dérivé de C et de C++ et ressemblant au langage Java qui correspond parfaitement à notre projet.

Conséquences:

Positives:

- Langage compilé, performant
- Maitrisé par l'équipe
- À fait ses preuves

Négatives:

- La syntaxe est particulièrement compliquée
-
-

Références:

- Documentation officielle de Node : <https://learn.microsoft.com/fr-fr/dotnet/csharp/>

Pour les systèmes de paiement

Titre : Utilisation de Stripe, Paypal, Apple & Play store

Date : 13 déc. 2023

Statut : Accepté

Contexte: Notre jeu sera complètement gratuit, mais il sera possible d'acheter des objets en jeu. Des skins pour les héros, skins pour les décors, des animations (emote), un pass de combat et des goodies.

Décision: Nous avons décidé d'opter pour plusieurs options de paiement, car elles sont toutes complémentaires. Stripe, Paypal, Apple & Play store. Grâce à ces 4 solutions, les utilisateurs peuvent acheter ce qu'ils veulent en tout confiance et simplicité.

Conséquences:

Positives:

- Disponibles dans plusieurs pays
- Grand choix de devises.
- Sécurisé

Négatives:

- Frais à payer par transactions
-
-

Références:

- Documentation officielle de Node : <https://learn.microsoft.com/fr-fr/dotnet/csharp/>



Domain Driven Design

Domaine : Jeu de carte Auto-battler

- **Gestion des Héros et Serviteurs** : Gère les différents Héros et les serviteurs
- **MatchMaking et Combat** : Mise en relation entre les joueurs et combats automatiques
- **Règles de Battlegrounds** : règles spécifiques du jeu

Modélisation du Domaine :

Entités :

- **Héros** : Entité qui représente le personnage choisi par le joueur avec un pouvoir unique.
- **Serviteur** : Entité principale du jeu avec des caractéristiques (Attaque et Point de vie), des types et des effets.

Valeur d'Objet

- **Niveau de Taverne** : Indique le niveau de taverne d'un joueur. Chaque serviteur possède un niveau de taverne. Les serviteurs de la taverne possède aussi un niveau de taverne. Un joueur ne peut acheter que des serviteur dont le niveau de taverne est inférieur ou égal au sien.
- **Statistique des Serviteurs** : Attaque, points de vie, types d'un serviteur
- **Effet d'un serviteur** : action qu'engendre un serviteur après un évènement particulier

Langage Ubiquitaire :

- **Héros** : Personnage que le joueur choisit en début de partie et qui va représenter son personnage pour le reste de la partie
- **Serviteurs** : Cartes recrutées par le joueur pour combattre contre les autres joueurs
- **Taverne** : La “boutique” où les joueurs achètent les serviteurs, elle peut être améliorer, actualiser ou geler pour la conserver d'un tour à un autre
- **Combat automatique** : Phase de jeu où les armées des joueurs s'affrontent automatiquement en 1 contre 1

Contexte Bornés :

- **Gestion des cartes** : Gère les caractéristique des serviteurs et Héro ainsi que leurs disponibilité dans la taverne.
- **MatchMaking** : Apparie les joueurs par niveau pour les faire jouer ensemble.
- **Gestion des combats** : Gères les combats aléatoires et les interactions en utilisant les données des cartes

Services de Domaine :

- **Service de Taverne** : Permet aux joueurs de recruter des serviteurs, d'améliorer le niveau de taverne, acheter et vendre des serviteurs
- **Service de combat** : Organise les combats entre les joueurs en appliquant les règles de combats et en déterminant le résultat



Budgetisation

Budgétisation totale du projet (salaires, serveurs, etc.)

Aa Name	☰ Tags	▽ Billing	# Nbr	Σ Monthly Costs	Σ Yearly Costs	# Costs
<u>Ingénieur architecture</u>	Salaire	Monthly	2	9000	108000	4 500,00 \$US
<u>Développeur jeux-vidéos</u>	Salaire	Monthly	80	256000	3072000	3 200,00 \$US
<u>Développeur web</u>	Salaire	Monthly	15	45000	540000	3 000,00 \$US
<u>Serveur web</u>	Serveur	Monthly	4	20000	240000	5 000,00 \$US
<u>CDN web</u>	CDN	Monthly	15	30000	360000	2 000,00 \$US
<u>Serveurs principaux (EU, US et Asie)</u>	Serveur	Monthly	3	375000	4500000	125 000,00 \$US
<u>Serveurs de secours (EU, US et Asie)</u>	Serveur	Monthly	3	375000	4500000	125 000,00 \$US
<u>Serveurs de secours de secours (EU, US et Asie)</u>	Serveur	Monthly	3	375000	4500000	125 000,00 \$US
<u>Serveur de développement</u>	Serveur	Yearly	1	2500	30000	30 000,00 \$US
<u>Serveur de tests</u>	Serveur	Yearly	1	2500	30000	30 000,00 \$US
<u>Serveur de preprod</u>	Serveur	Yearly	1	2500	30000	30 000,00 \$US
<u>CDN jeu</u>	CDN	Monthly	30	60000	720000	2 000,00 \$US
<u>Création des OST</u>	Achat	Unique	1	300000	300000	300 000,00 \$US

Aa Name	≡ Tags	⌚ Billing	# Nbr	Σ Monthly Costs	Σ Yearly Costs	# Costs
<u>Création du lore</u>	Achat	Unique	1	75000	75000	75 000,00 \$US
<u>Création des textures</u>	Achat	Unique	1	450000	450000	450 000,00 \$US
<u>Création des animations</u>	Achat	Unique	1	100000	100000	100 000,00 \$US
<u>Création des nouveautés</u>	Achat	Yearly	1	16666.666666666668	200000	200 000,00 \$US
<u>Création des maquettes (accessibilité)</u>	Achat	Unique	1	80000	80000	80 000,00 \$US
<u>Marketing</u>	Achat	Monthly	1	20000	240000	20 000,00 \$US

Pour les salaires :

- <https://www.hellowork.com/fr-fr/metiers/architecte-technique.html>
- <https://www.hellowork.com/fr-fr/metiers/developpeur-web.html>
- <https://gamingcampus.fr/metiers/programmeur-de-jeux-video.html>

+ les sites d'annonce d'emploi (linkedin principalement) Pour les serveurs (web & jeu) :

- <https://calculator.aws/#/>
- <https://cloud.google.com/products/calculator-legacy>
- <https://www.ovhcloud.com/fr/bare-metal/prices/>
- <https://azure.microsoft.com/fr-fr/pricing/calculator/>
- <https://www.alibabacloud.com/fr/pricing-calculator#/>

Pour les moteurs de jeu :

- <https://unity.com/>
- [https://fr.wikipedia.org/wiki/Unity_\(moteur_de_jeu\)](https://fr.wikipedia.org/wiki/Unity_(moteur_de_jeu))
- <https://godotengine.org/>
- [https://fr.wikipedia.org/wiki/Godot_\(moteur_de_jeu\)](https://fr.wikipedia.org/wiki/Godot_(moteur_de_jeu))
- <https://www.unrealengine.com/>
- https://fr.wikipedia.org/wiki/Unreal_Engine
- <https://gamemaker.io/>
- <https://fr.wikipedia.org/wiki/GameMaker>

Pour les services de paiement :

- <https://www.paypal.com/us/webapps/mpp/country-worldwide>
- <https://www.paypal.com/us/webapps/mpp/merchant-fees>
- <https://stripe.com/fr/global>
- <https://stripe.com/fr/pricing>
- <https://www.paysafecard.com/fileadmin/Website/Dokumente/B2B/availability-of-products-en.pdf>
- <https://support.google.com/googleplay/android-developer/answer/112622?hl=en>

Pour les databases : Sites officiels