FICHE : PoC & Choix Technologiques

=> Après avoir rempli cette fiche :convertir ce document en PDF et le téléverser dans votre dépôt Github dans un répertoire nommé 'doc'.

Compte rendu de la veille technologique PoC

Questions de RECHERCHE sur la PoC

Quel est l'opération la plus risquée de votre projet, l'élément qui risque de ne pas fonctionner ? Quelle est l'interaction entre deux technologies ? Quel est le PROBLÈME technique à résoudre ?

L'élément qui risque de ne pas fonctionner dans mon projet est la compatibilité entre le client Unity et le serveur NodeJS pour permettre un mode multijoueurs. Le problème technique à résoudre est donc de faire communiquer le serveur NodeJS et le côté client Unity pour échanger des informations en temps réel.

Quelle est la QUESTION que vous vous posez et que vous demandez à internet de répondre ?

La question que je me pose alors est : "Est-ce que Unity peut communiquer en temps réel avec un serveur NodeJS ?"

PoC = Preuve de concept

Quel genre de preuve de concept minimale pourrait valider que le problème n'existe pas ou qu'une solution a été trouvée ? Décrivez chaque élément du code requis.

La preuve de concept qui pourrait valider que le problème n'existe pas serait de réussir à connecter plusieurs clients Unity au même serveur NodeJS et de voir si les différents clients peuvent s'échanger des informations grâce au serveur.

Les marque-pages identifiés lors de vos recherches

Lien vers une page publique contenant vos marque-pages collaboratifs ou lister les marque-pages directement ici. Pour chaque lien : URL, nom de la page et description sommaire.

1 - https://socket.io/ , Site web de Socket.io, Page contenant toute la documentation de Socket.io aussi bien du côté client que du côté serveur ainsi que plusieurs exemples de morceaux de code sur lesquels je pourrais m'appuyer.

2 - Tutoriel YT, Tutoriel d'un jeu multijoueur avec Unity et Node JS, Il est expliqué dans ce tutoriel comment faire communiquer Socket.io avec Unity

Les preuves de concept

Pour chaque preuve de concept réalisée : identifier le but de la preuve de concept (ce qu'elle vérifie), le lien vers le sous-répertoire de votre dépôt GitHub qui contient le code de la preuve de concept ainsi que les résultats de votre expérimentation, puis, finalement, vos conclusions.

Au moins une preuve de concept doit être documentée et réalisée.

Première POC réalisée

Preuve : Une communication entre Unity et NodeJS est possible

URL Github :

Expliquez votre Poc

Décrivez la Poc en détail.

La preuve de concept que je vais faire pour prouver que NodeJS peut communiquer avec le client Unity est la suivante : je vais essayer de connecter 4 clients Unity au serveur NodeJS grâce au framework Socket.io et tenter d'envoyer un message au serveur avec l'un des clients Unity et faire en sorte que le message après avoir été traité par le serveur, soit également envoyé aux 3 autres clients Unity. Si j'arrive à mettre en place cette POC, cela signifiera que je pourrais faire communiquer NodeJS et Unity et envoyer plus qu'un simple message textuel.

Que PROUVE la Poc ?

La Poc prouve qu'il est bien possible de faire communiquer Unity et un serveur NodeJS en utilisant la bibliothèque Javascript Socket.io côté serveur et la librairie C# socket.io-unity côté client.

Que reste-t-il à prouver ?

Si je réussis ma POC, alors je pense qu'il n'y a plus rien à prouver, étant donné que toutes les autres fonctionnalités sont gérées essentiellement par Unity.

Quels sont vos résultats de la Poc ?

Les résultats de la POC sont :

1 - Un client Unity peut envoyer un message au serveur NodeJS

2 - Le serveur traite le message du joueur et l'envoie aux autres clients Unity connecté au serveur

3 - Les autres clients Unity reçoivent bien le message

Première technologie sélectionnée (la nouvelle)

Technologie : Mise en place d'un serveur à l'aide de NodeJS (et de la librairie Socket.io)

URL : https://socket.io/

Justifier votre choix technologique pour cette technologie

Expliquer à l'aide d'une argumentation rationnelle votre choix technologique. Établir votre justification à l'aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.

J'ai choisi d'utiliser la bibliothèque Javascript Socket.io en tant que nouvelle technologie pour permettre la communication entre Unity et le serveur NodeJS. Cette bibliothèque possède une grande documentation directement via leur site socket.io et de nombreux exemples aussi bien côté serveur que côté client. De plus, quelques tutoriels bien expliqués sont disponibles via Youtube. Cette librairie est également disponible en C# sur Github (elle est open-source) que je pourrais directement importer dans mon projet Unity pour l'utiliser côté client. Enfin, Socket.io est utilisé par de nombreuses personnes, de nombreux projets d'exemples sont disponibles sur Github et de nombreuses résolutions de problèmes.

Deuxième technologie sélectionnée (la connue)

Technologie : Unity

URL : https://unity.com/fr

Justifier votre choix technologique pour cette technologie

Expliquer à l'aide d'une argumentation rationnelle votre choix technologique. Établir votre justification à l'aide de liens avec les fonctionnalités, contraintes et risques de votre projet. Un tableau comparatif permettant de synthétiser votre réflexion pourrait être un apport judicieux à vos explications.

J'ai choisi d'utiliser Unity car c'est un moteur de jeu simple à prendre en main, intuitif et que j'utilise déjà dans le cours de réalité virtuelle. Unity utilise comme langage de script le C# qui ressemble beaucoup au java que je maîtrise bien ce qui me permettra de ne pas être perdu au niveau de la syntaxe. De plus, Unity propose une grande variété d'asset 2D qui me seront très utiles pour le côté graphique de mon jeu. Par ailleurs, il existe de nombreuses librairies (UnityEngine par exemple) qui permettent de faciliter le développement et la manipulation d'objets Unity afin de leur donner des propriétés (liées aux collisions, aux mouvements…).

La fiche contenant la grille de comparaison entre Unity et Unreal Engine récapitule bien mon choix de cette technologie. Vous la trouverez via ce lien : Grille de comparaison (Unity Vs Unreal Engine)