

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 ?

Le plus dur est de réussir à faire une architecture en serveurs dédiés, c'est à dire que chaque nouveau salon induit la duplication du build serveur de mon jeu pour mettre en relation plusieurs joueurs.

Quelle est l'interaction entre deux technologies ? Quel est le PROBLÈME technique à résoudre ?

Les problèmes résident dans les versions de unity et les versions de Socket.io, pas mal des versions sont obsolètes car elles ne fonctionnent pas entre elles (cela m'a pris 2 séances minimum juste pour installer l'environnement de code)

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

Est-il possible de faire un jeu avec serveurs dédiés en 2 mois en ayant aucune connaissance dans Socket.io ?

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.

1. Héberger le build serveur quelque part (ex: AWS, VPS, ou service cloud).
2. Implémenter un système permettant la duplication dynamique d'une instance serveur lors de la création d'un salon.
3. Assurer que les joueurs puissent se connecter à l'instance serveur assignée et recevoir un message de confirmation indiquant leur salle d'appartenance.

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.

Socket.IO Unity GitHub : <https://github.com/itisnajim/SocketIOUnity>

Contient une implémentation fonctionnelle de Socket.IO pour Unity avec divers exemples.

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 : (Que prouve la Poc)

URL Github : <https://github.com/cegepmatane/projet-sp-cialis-Gauthierpl>

EXPLIQUEZ VOTRE POC

Explication de la PoC :

La PoC met en place un serveur Node.js avec Socket.IO, permettant de gérer les salons de jeu. Unity peut interagir avec ce serveur pour créer, rejoindre et lister les salons en temps réel.

Que PROUVE la PoC ?

- La communication entre Unity et le serveur Node.js fonctionne bien.
- La création et la gestion des salons sont possibles.
- L'architecture multi-salon peut être exploitée.

Que reste-t-il à prouver ?

- La duplication dynamique d'instances serveurs.
- L'hébergement et la gestion d'un grand nombre de salons.
- La stabilité des connexions et des performances.

Quels sont les résultats de la PoC ?

- Connexion établie entre Unity et le serveur Socket.IO.
- Création et listing des salons fonctionnels.
- Tests fonctionnels validés en local.

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : 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.

- **Avantages :**

- Solution simple et efficace pour la gestion des WebSockets.
- Bonne documentation et communauté active.
- Facile à intégrer avec un serveur Node.js.

- **Inconvénients :**

- Certaines versions incompatibles avec Unity.
- Peut poser des problèmes de scalabilité si mal géré.

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : Unity

URL : <https://unity.com/>

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.

- **Avantages :**

- Moteur de jeu puissant et très utilisé.
- Possède une grande communauté et de nombreuses ressources d'apprentissage.
- Compatible avec plusieurs plateformes.

- **Inconvénients :**

- Problèmes de compatibilité avec certaines versions de Socket.IO.
- La gestion du réseau en temps réel peut être complexe.

Conclusion personnelle

La PoC a permis de valider la connexion entre Unity et un serveur Node.js utilisant Socket.IO. Cependant, le challenge principal réside dans la gestion des serveurs dédiés pour chaque salon. J'ai eu beaucoup de problèmes avec la conversion de données JSON et l'afficher sur unity alors que c'était quelque chose que j'ai déjà fait de nombreuses fois par le passé. Je vais donc devoir régler ce problème au plus vite afin de pouvoir passer à la prochaine étape. La prochaine étape consistera à tester et implémenter la duplication des instances serveurs, tout en assurant la scalabilité et la stabilité du système.