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 technologie ? Quel est le PROBLÈME technique à résoudre ?

Je fait fonctionner 3 technologies ensemble, le système d'entité du DOTS, le langage ECS du DOTS et leur système de rendu hybrid. Je pense que les problèmes que je risque de rencontrer sont la limite de documentation sur cette technologie assez récente.

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

La question que je vais poser sur internet c'est est ce que je peux générer des objets avant le runtime et les garder après le runtime sur unity.

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.

Je vais faire une génération de cartes très basique.

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.

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 : L'utilisation des Entities avec le code ECS

URL Github : https://github.com/cegepmatane/projet-specialise-2022-TMRomain

Expliquez votre Poc

Décrivez la Poc en détails.

Il y a tout d'abord la conversion de mon gameobject classique en entité, ensuite le script fonctionnement de la rotation comprend les valeurs qui vont être appliquées et spécifie quel élément va utiliser les autres scripts. Le script composant de rotation lui va spécifier ce qui va être modifié sur notre entité.Et pour terminer le script système de rotation spécifie ce qui va être fait avec les valeurs et les composant modifier sur notre entité.

Que PROUVE la Poc ?

Elle prouve que l'utilisation et la combinaison de deux technologies DOTS, l'ecs et les entités.

Que reste-t-il à prouver ?

Maintenant il faut prouver que la génération a partir de rien sans gameobject est possible.

Quels sont vos résultats de la Poc ?

La preuve de concept est un succès et fonctionne comme prévu.

Première technologie sélectionnée (la nouvelle)

Technologie : DOTS

URL : https://unity.com/fr/dots

Justifier votre choix technologique pour cette technologie

Puisque mon jeu se base sur un système de génération de carte en 3D et que ce système sera assez demandant en termes de ressource, le DOTS va permettre de faire une génération plus rapide et moins lourde sur les différents composants de l'ordinateur. Exemple si je génère un cube certain composant comme sa boite de collision ne vont pas m'être utile mais vont quand même demander des ressource au système. Avec le système d'entité je peut me concentrer sur la position ou le mesh de mon cube et ignorer le reste des proprieter.

Deuxième technologie sélectionnée (la connue)

Technologie : Unity

URL : https://unity.com/fr

Justifier votre choix technologique pour cette technologie

Unity est le seul moteur de jeu gratuit que je connaisse qui permet une telle optimisation avec un nouveau langage proche de la machine. De plus je suis familier avec son interface ce qui permet un développement plus rapide.