

FICHE : LANCEMENT "KICKOFF" DU PROJET

IDENTIFIER VOTRE CHOIX DE SUJET

Titre du projet : Diddy Kong et ses amis

URL si applicable :

DÉCRIVEZ VOTRE PROJET (2 PARAGRAPHES MINIMUM)

Je voudrais améliorer mon jeu vidéo en ajoutant une option multijoueur. Pour cela, j'utiliserai une architecture basée sur des serveurs dédiés grâce à une implémentation utilisant WebSocket avec Node.js. Chaque joueur pourra créer un salon privé ou public, où d'autres joueurs pourront se connecter en temps réel. Les communications entre les joueurs et le serveur seront gérées via des WebSocket pour garantir des interactions rapides et fluides.

Le défi principal sera de concevoir une logique serveur qui permet de gérer la création dynamique de salons. Chaque salon sera autonome et se détruira automatiquement dès qu'il sera vide. Les joueurs pourront interagir en temps réel grâce à WebSocket, et les données de jeu, comme les cartes ou les configurations, seront récupérées depuis une base de données hébergée sur Hostinger. La transition vers WebSocket permettra une meilleure flexibilité, tout en réduisant la dépendance aux solutions tierces comme NetCode.

NOMMEZ 3 COMPÉTITEURS À VOTRE PROJET

qui vous inspireront pour les fonctionnalités (avec les liens)

Nom du compétiteur 1	https://atelier801.com/index
Nom du compétiteur 2	https://osu.ppy.sh/
Nom du compétiteur 3	https://www.leagueoflegends.com/en-us/

LISTER LE MATÉRIEL DE CE PROJET

- **PC personnel Windows**
- **Logiciels nécessaires :**
 - **Node.js (pour le backend des WebSocket)**
 - **Unity (pour la partie client du jeu)**
 - **Hébergement Hostinger (pour la base de données et éventuellement pour héberger le serveur WebSocket)**

Est-ce que tous les éléments matériels nécessaires au projet sont en votre possession ? Sinon, lesquels ?

Oui

Aimeriez-vous que le département vous prête un composant, une pièce ? Si oui laquelle ?

Non

IDENTIFIER DEUX ASPECTS TECHNOLOGIQUES

Aspects : Serveurs, Services, Algorithmes, Web, IoT, etc

Aspect 1 : Serveurs dédiés utilisant Node.js et WebSocket

Aspect 2 : Jeu en ligne avec interaction en temps réel

PRÉCISEZ L'APPRENTISSAGE TECHNIQUE QUI SE FERA PENDANT LE PROJET

soyez aussi précis que possible (nom de la technologie, de la librairie, url)

Technologie : WebSocket via Node.js et Unity (C#)

URL : https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
<https://medium.com/unity-nodejs/websocket-client-server-unity-nodejs-e33604c6a006>
<https://www.youtube.com/watch?v=13HnJPstnDM>

Quelle utilisation en ferez-vous à l'intérieur du projet ?

Les WebSocket serviront à gérer toutes les communications temps réel entre le serveur (Node.js) et les clients (Unity). Lorsqu'un joueur crée un salon, un serveur dédié (processus Node.js) sera créé pour héberger ce salon. Les WebSocket permettront de transmettre des informations comme les mouvements des personnages, les scores, et les choix des cartes. Toutes les données essentielles seront synchronisées via cette technologie pour assurer une expérience fluide et réactive.

Le serveur Node.js communiquera également avec une base de données distante (hébergée sur Hostinger) pour charger ou sauvegarder des éléments de jeu comme les cartes personnalisées ou les statistiques des joueurs.

WebSocket (Server & Client) using NodeJS & Unity C#

<https://www.youtube.com/watch?v=13HnJPstnDM> &
<https://medium.com/unity-nodejs/websocket-client-server-unity-nodejs-e33604c6a006>

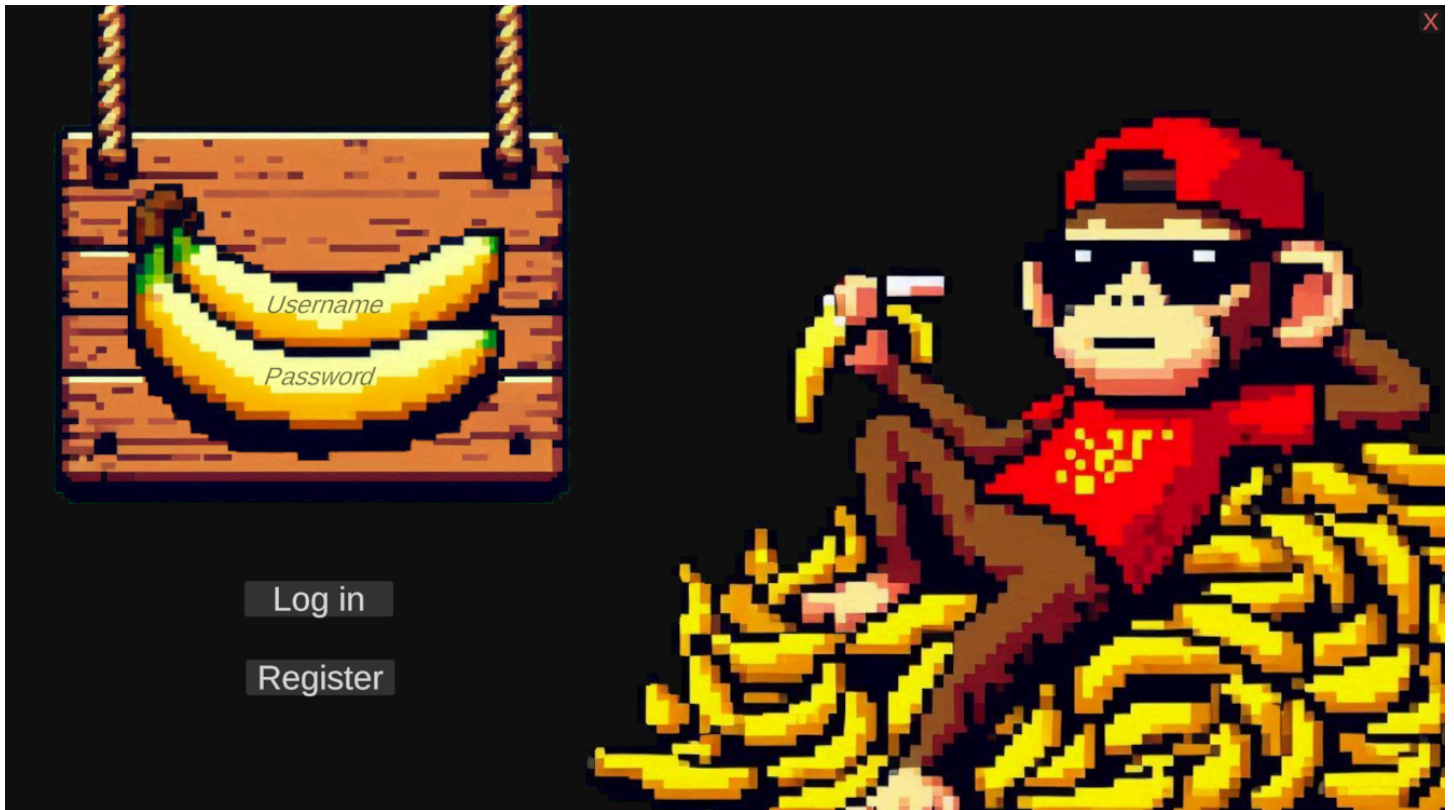
<https://www.youtube.com/watch?v=3psO9x-H-Vc>

<https://www.youtube.com/watch?v=osANZ9aiglg>

<https://www.youtube.com/watch?v=Gq7fenbjehs>

ILLUSTRATION

Si une illustration peut nous aider à mieux comprendre le projet, merci de l'inclure ici.



Vous pouvez inclure plus de matériel directement dans Github.

=> Après avoir rempli cette fiche : veuillez accepter l'assignment GitHub, convertir ce document en pdf et le remettre dans Github directement.

SVP : placer aussi le lien collaboratif du document incomplet avant de commencer.