

Projet Piccadilly Circus

Rapport de début de projet tutoré

Keller Guillaume - Georgelin Tom - Achterfeld Matéo - Jossic Martin
S3B

Description du sujet :

Notre projet consiste en une application web massivement interactive s'affichant sur support connecté soit de type virtuel (télévision, écran d'information, écran publicitaire géant) soit de type physique (installations robotiques, projection type morphing).

L'accès au jeu sera donné par l'application sur le support général, il pourra notamment utiliser le système de QR Code, disponible sur l'écran principal de l'application avant que celle-ci soit lancée. Dès ce moment, les joueurs seront positionnés en phase d'attente jusqu'au lancement des actions (quand suffisamment de personnes seront connectées).

Le système utilisera par la suite le navigateur web du support choisie par l'utilisateur (téléphone, tablette...) pour charger une page correspondant à son "interface de jeu". Celle-ci évoluera en fonction de son avancement dans le jeu.

Le gérant de l'application aura la possibilité de choisir entre un nombre cible d'utilisateurs ou une date de début pour lancer l'application. Le support devient alors un moyen de communication interactif efficace pour toute personne cherchant à diffuser un message ou à réunir les foules. L'objectif principal serait alors de proposer un divertissement facilement accessible à un grand nombre de personnes, d'effectuer rapidement des sondages ou encore de différer des traditionnelles publicités statiques en ce qui concerne l'interaction humaine.

Le fait de passer par une application web contrairement à une application mobile classique permettra de créer facilement de l'audience (accès au service facilité), ce qui est recherché par les différents services de communication.

Le concept peut par la suite être exploité à plus grande échelle, sur les chaînes de télévisions par exemple qui pourraient alors se passer des jeux par SMS et des frais de communication mobiles qu'ils engendrent. Des jeux en direct avec affichage du score des téléspectateurs sur la télévision ou des publicités interactives contenant des mini jeux seraient plus agréables pour les téléspectateurs mais tout aussi efficaces.

Étude de l'existant

I. AirConsole

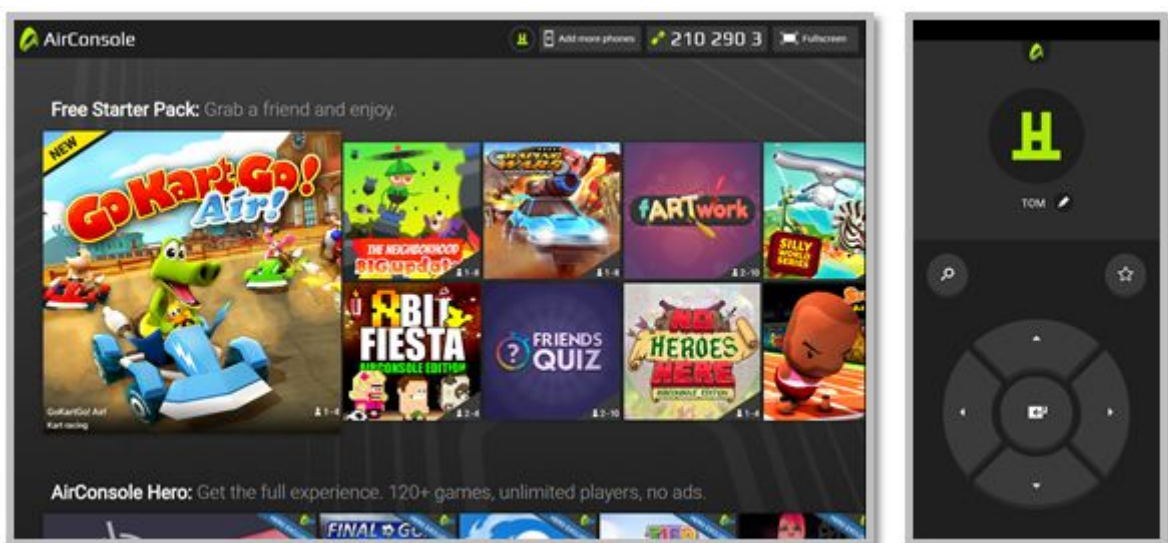
AirConsole est un jeu de salon créé par la startup N-Dream. Le but de ce jeu est que chaque joueur se connecte sur l'écran principal à l'aide de son smartphone qui va donc lui servir de "contrôleur".

Le principe de ce jeu est simple : on connecte un écran à un serveur (pour cela, il suffit d'accéder à leur site). Ensuite, nous obtenons un code (capture ci-dessous).



Dans un second temps, chaque joueur se connecte à un serveur à l'aide de son smartphone et du code, le tout en passant par l'application préalablement téléchargée sur le store. Le smartphone devient alors une manette qui permettra à chaque joueur de jouer.

Une fois connecté, nous avons accès à des dizaines de jeux :



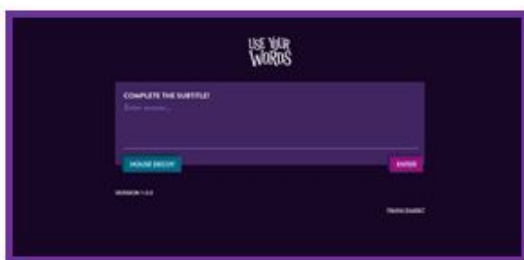
II. Use Your Words

Use Your Words est une fois de plus un jeu de salon où chaque joueur joue sur son smartphone. Cependant, il n'est cette fois pas nécessaire d'installer une application sur son smartphone : il suffit de se connecter à un site internet. Il faut néanmoins installer le jeu sur son ordinateur.

Une fois le jeu lancé, il génère un code à saisir sur le site à l'aide de son smartphone (capture ci-dessous).



Une fois tous les appareils connectés, le jeu se lance et des images ou des vidéos sont alors montrées sur l'écran central. Il suffit de les sous-titrer avec une phrase drôle et d'envoyer notre réponse (capture de gauche ci-dessous). Une fois que tout le monde a soumis sa réponse, nous voyons la vidéo avec les différentes réponses et nous pouvons voter pour celle que nous trouvons la plus drôle (photo de droite ci-dessous).



III. Les publicités interactives

Il existe énormément de publicités interactives différentes. De la simple publicité qui évolue au cours du temps (photo de gauche ci-dessous) à la publicité nécessitant une action de la part des consommateurs (photo de droite ci-dessous).



IV. Les sites de questionnaires en ligne

De nombreux sites proposent d'effectuer des quizz en ligne. Parmi ceux là, il y a le site Socrative.

Il est utilisé dans l'éducation par certains enseignants pour vérifier la compréhension du cours par des questions. Il peut s'apparenter à la première application que nous souhaitons mettre en place à savoir une plateforme de sondage interactive.

V. Comparaison avec notre projet

Notre projet est en quelque sorte un condensé de tout ce que nous avons pu trouver d'existant. En effet, nous voulons pouvoir créer des jeux disponibles sur un seul écran tout en étant contrôlé par chaque consommateur grâce à son smartphone.

Sur les 2 jeux que nous avons trouvés, il y a 2 façons différentes de faire la connexion avec les joueurs :

- Dans le cas de « AirConsole », chaque utilisateur doit avoir l'application sur son smartphone. C'est exactement ce que nous voulons éviter car cela représenterait une perte de consommateurs. Son installation représente une perte de temps pour beaucoup d'entre nous.

- Dans le cas de « Use Your Words » il n'y a aucune application à installer côté client. Il faut par contre que le jeu soit installé sur un ordinateur pour pouvoir diffuser les différentes images/vidéos.

Pour les publicités, il n'existe pas encore de projet comme ce à quoi nous avons pensé. Nous pouvons innover de ce côté là.

Pour finir, les sondages durant les émissions télévisées se font grâce à l'envoi de SMS payants (avec possibilité de remboursement). Or, avec notre projet, nous pourrions rendre cela beaucoup plus interactif et gratuit pour le téléspectateur. En revanche, la chaîne de télévision devra changer ses méthodes de monétisation (le coût d'envoi des SMS). Pour pallier cette perte elle pourrait faire sponsoriser ses sondages par des marques (publicités dans l'application). Cela lui permettrait donc de ne pas perdre d'argent, le tout avec un plus haut taux d'engagement. Tout le monde serait gagnant.

Choix techniques envisagés

Le projet doit se présenter sous la forme d'une application web (site web évolué) pour que les utilisateurs puissent facilement lancer le jeu en flashant un QR code (pas de téléchargement d'application préalable).

Cette rapidité d'accès au jeu permet de ne pas décourager l'utilisateur impatient en lui permettant de jouer immédiatement (une application traditionnelle peut mettre du temps à être installée, surtout quand on a une mauvaise connexion internet).

Pour réaliser ce projet, les langages utilisés côté client (navigateur) seront donc HTML, CSS et JavaScript. Côté serveur, différents environnements s'offrent à nous. Apache et PHP sont les outils côté serveur qu'on a appris cette année. On a ensuite d'autres frameworks web orienté serveur (bibliothèques permettant de programmer facilement des sites web) utilisables dans différents langages : Ruby (framework Ruby on Rails), Python (framework Django), ASP.NET, JavaScript (Node.JS)...

Nous avons choisi Node.JS parce qu'il utilise le même langage de programmation que celui se trouvant côté client : JavaScript. Dans le groupe, 2 étudiants se sont déjà familiarisés avec Node.JS. Cela permettra une avancée rapide. Le langage JavaScript quant à lui est déjà connu de toute notre équipe. De plus, notre projet fonctionne en temps réel et Node.JS est bien adapté à cela : il peut gérer un très grand nombre d'utilisateurs en simultané et peut utiliser le protocole WebSocket pour envoyer des données vers le client sans que le client ait à envoyer une requête.

Le protocole WebSocket

Quand un site web utilise uniquement le protocole HTTP, il peut facilement envoyer des données au serveur à l'aide de requêtes HTTP. En revanche, quand le serveur veut envoyer des données au client (nouvelles notifications, action d'un autre joueur dans un jeu...), il doit attendre que le client "s'actualise" en lui envoyant une requête. Le client doit donc envoyer des requêtes d'actualisation à intervalle régulier pour recevoir les données événementielles inopinées du serveur.

Ce système peut fonctionner, mais dans le cas d'événements "temps réel" (actions dans un jeu), le client doit se rafraîchir très régulièrement, ce qui est assez coûteux en bande passante. Le protocole WebSocket remédie à cela en maintenant une connexion TCP ouverte entre le client et le serveur. Le serveur comme le client peut grâce à cette connexion envoyer des données à tout moment vers l'autre pair. Le client n'a plus besoin de requêtes d'actualisation car il peut recevoir des messages du serveur à tout moment (tant que la connexion reste ouverte bien sûr). On économise donc de la bande passante et en plus de ça, le joueur reçoit les données événementielles immédiatement (pas de délai d'actualisation).

Ce protocole sera utilisé dans notre projet car le serveur a besoin d'envoyer rapidement (en temps réel) des données vers le joueur pour l'informer sur l'état de la partie en consommant un minimum de bande passante pour pouvoir gérer un maximum de joueur au même moment.

Infrastructure matérielle

A. Serveur

Parmi les besoins du projet, nous retrouvons des infrastructures matérielles.

Le serveur sera la pièce centrale du fonctionnement du jeu. En effet, celui-ci sera le maître du jeu, mettra à disposition l'application, gèrera l'ensemble des actions et sera relié directement à l'utilisation que nous pourrons faire de nos applications. Un serveur performant sera donc la clé de voûte du côté massif de l'application développée.

Il faudra par conséquent choisir entre la proximité (Serveur tournant sur un Raspberry Pi par exemple) et le coût (infrastructure matérielle plus grande ou location de serveur en ligne).

B. Support d'affichage

Le support d'affichage pourra également varier en fonction des applications choisies :

- Une application de sondage pourra être facilement incorporée sur un écran classique d'ordinateur ou de télévision.
- Une application de jeu pourra quant à elle se décliner sous la forme d'une installation robotique plus complexe (cf. Sujet sur la robotique en essaim)
- Une application de dessin pourrait utiliser les techniques de diffusion (vidéoprojetée) sur des bâtiments.

Conclusion

L'enjeu majeur de ce projet réside dans son aspect massivement multijoueur avec des joueurs ayant des connexions à vitesse variable. Pour surmonter ce défi, il faudra trouver des solutions pour que le joueur qui possède la connexion la plus lente puisse tout de même profiter au mieux de l'application. Un travail devra donc être effectué sur l'optimisation du transfert de données entre clients et serveurs et peut-être même sur l'anticipation des actions des joueurs les moins bien connectés. Le but sera de limiter les dégâts d'une forte latence du joueur sur sa qualité de jeu (retard des actions).