

Sujet: Piccadilly Game

Dans le cadre de notre projet tutoré, nous avons pour but de réaliser un jeu massivement multi-joueurs composé d'un support principal affichant le jeu et d'un nombre variable de joueurs chacun connectés via leur propre support.

Afin de réaliser ce projet, il nous faudra donc répondre à diverses problématiques pour que le rendu final soit fonctionnel pour tous les utilisateurs aussi nombreux soient-ils. Dans un premier temps, nous détaillerons le sujet de façon à rendre le sujet plus clair et à déterminer les missions attendues, dans un second temps, nous effectuerons une étude de l'existant qui nous permettra de comparer et récupérer les points (positifs comme négatifs) qui pourraient servir lors de la conception de notre jeu, enfin, nous explorerons les différentes solutions permettant de mener à bien notre développement grâce à une étude technique.

Le but est donc de réaliser un jeu proposant un support principal unique pour tous avec lequel on peut interagir et faire avancer le jeu depuis un support externe relié à l'affichage principal par un serveur. Il serait possible de prendre part au jeu à l'aide d'un lien fourni par l'hôte et d'y participer grâce à un support connecté à internet qui communiquerait ses actions et changerait l'affichage de l'écran commun.

Pour réaliser ce projet, il nous faudra donc développer un jeu capable d'accueillir un nombre important de joueurs, pour cela il nous faudra donc utiliser un langage orienté réseau qui pourra gérer le nombre variable de participants. Le langage qui nous semblait le plus adapté était donc le javascript et plus particulièrement le node.js qui est une plateforme logicielle libre qui est orientée vers les applications réseau et qui est capable de gérer une charge variable d'utilisateurs. Pour la mise en page du support principal ainsi que des pages en ligne des utilisateurs, l'utilisation de l'HTML couplé à du CSS nous paraissait la meilleure méthode pour proposer un affichage facilement disponible pour tous et adaptable à tous supports (téléphones, ordinateurs...).

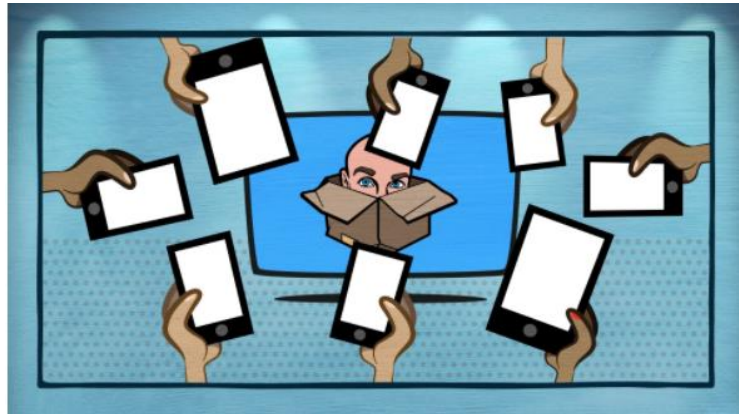
Il nous faudrait également pouvoir héberger le jeu afin de pouvoir y accéder directement en ligne afin que chaque utilisateur puisse y prendre part plus facilement, de plus cela rendra possible le partage de données entre le serveur et les clients, pour cela, notre choix se porterait sur l'utilisation d'un Raspberry Pi: plus pratique et simple d'utilisation, il nous permettrait de découvrir l'utilisation et la configuration d'un serveur.

Pour le gameplay, de nombreuses possibilités s'offrent à nous. Le sujet étant assez libre, plusieurs idées nous sont déjà venues. Dans un premier temps, il serait possible de proposer une course de voitures: la piste principale serait affichée sur l'écran principal. On y verrait chaque voiture associée à une couleur se déplacer, puis sur chaque écran individuel, on aurait des boutons de contrôle et un indicateur de la couleur pour que l'utilisateur puisse se repérer sur la course. Un autre concept serait celui d'un Tetris géant basé sur le vote des joueurs: la pièce à venir apparaît sur leur écran ainsi que des boutons pour l'orienter comme ils pensent le mieux, on récupérerait ensuite les différentes positions choisies par chacun puis la pièce serait placée en fonction des statistiques où la majorité de personnes ayant choisi le même placement verront leur position choisie.

Enfin, une idée plus simple mais ludique serait celle d'un quizz où la question et les différentes propositions s'affichent sur l'écran principal et le joueur peut sélectionner la réponse qu'il pense juste sur son propre écran, à l'issue de cela, les statistiques des réponses s'afficheraient sur le grand écran et le joueur verrait sur son support si la réponse qu'il a proposée était la bonne.

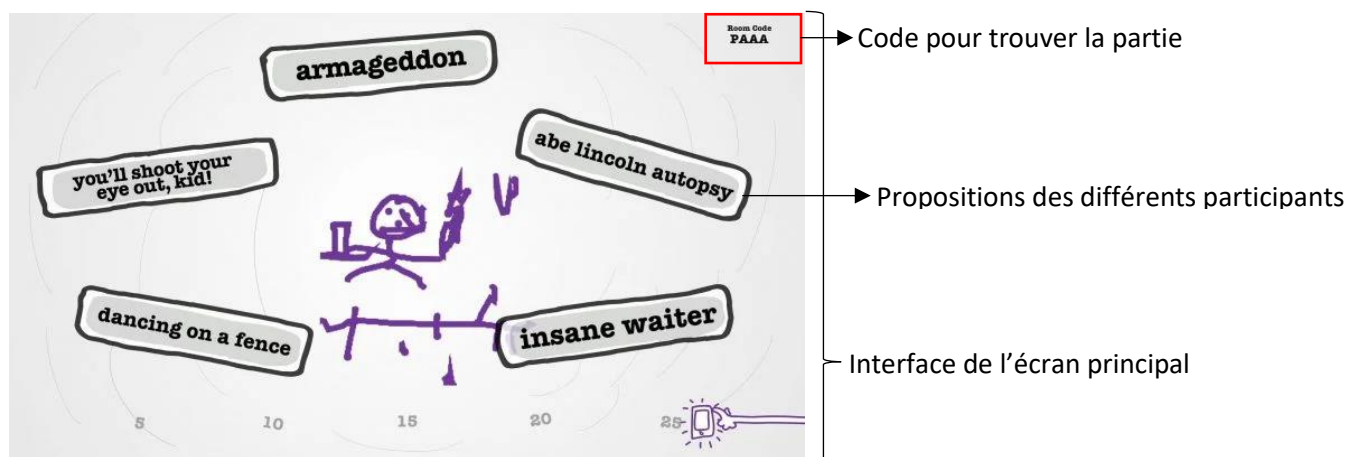
Étude de l'existant.

Le premier jeu se rapprochant le plus de l'idée générale de notre projet serait « JackBox Games », il s'agit d'un jeu basé sur le concept d'un écran principal et d'un jeu accessible via un lien ou code d'invitation pour les autres joueurs. Il est possible de s'y connecter avec n'importe quel support tant que celui-ci possède un accès à Internet (téléphone, ordinateur...). Le jeu propose un pack de nombreux mini-jeux différents.



Ici nous parlerons d'un de leur jeu: Drawful.

Le but du jeu est simple, chaque joueur reçoit un mot ou une expression insolite qu'il doit essayer de dessiner, cependant ce dernier ne possède pas de gomme ou de moyen de revenir en arrière sur son dessin ce qui complique le tout. Une fois ceci fait, la deuxième phase débute alors: tous les dessins sont montrés et chaque joueur doit donner un titre qui correspond au dessin de manière anonyme. Le but du jeu est de trouver la bonne réponse tout en trompant les autres joueurs et en leur faisant sélectionner le faux titre que vous venez de créer.

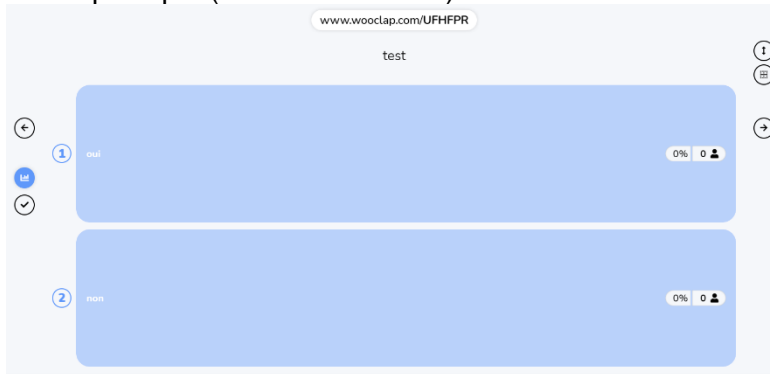


La différence de JackBox avec notre projet est d'abord la nécessité qu'au moins une personne possède l'application (payante), contrairement à nous, pour qui le site de jeu serait disponible gratuitement en ligne. La plupart de leurs jeux sont limités à 8 personnes, or il nous faut pouvoir proposer un support capable d'accueillir de nombreux joueurs. Enfin, leur catalogue de jeu, bien que plutôt rempli, ne propose pas de jeux type actions, course de voiture comme nous avons proposé dans la première partie.

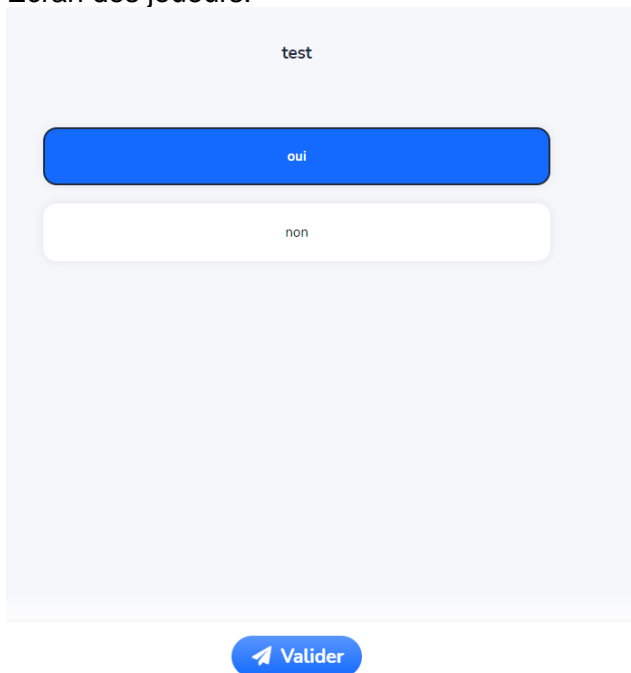
Sujet: Piccadilly Game

Le produit que nous avons ensuite comparé est celui de Wooclap, il s'agit d'un système de vote interactif qui permet de créer des questionnaires. Les parties se rejoignent grâce à des liens personnalisés. La fonctionnalité de quizz ressemble à une de nos idées de projet et il serait donc intéressant de se baser sur les idées de ce site pour la réalisation, notamment pour l'idée de l'hôte créant la partie puis les résultats s'affichant individuellement sur chaque écran de joueurs. De plus, à l'issue de chaque question, la bonne réponse est affichée accompagnée d'un pourcentage de bonnes réponses.

Écran principal (avec les résultats).



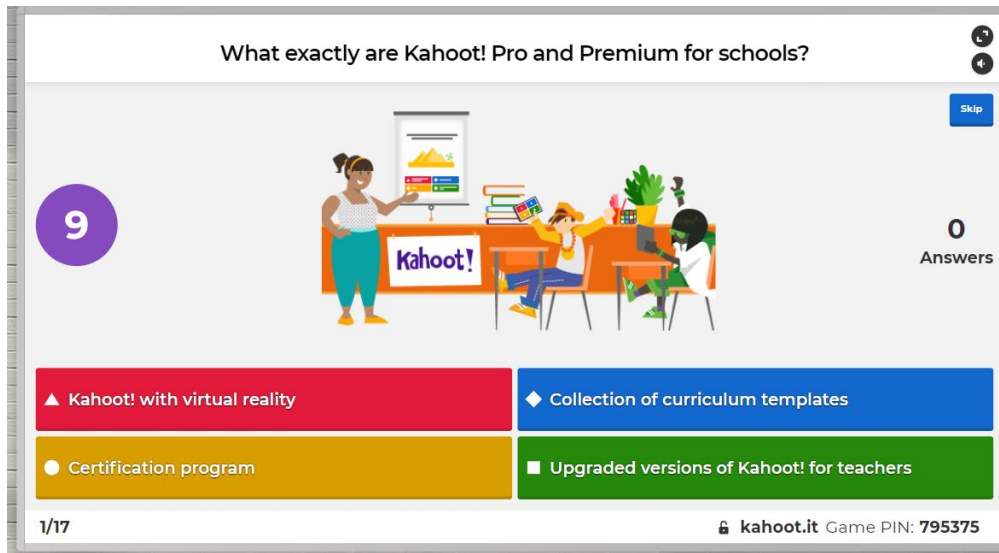
Écran des joueurs.



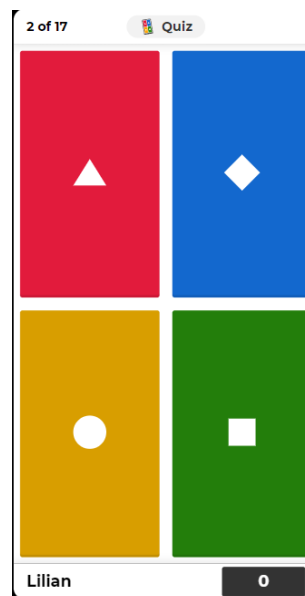
La différence avec notre projet est principalement l'utilisation de l'écran, qui, pour les joueurs dans le cas de Wooclap est simplement une copie de l'écran principal, ce qui donne peu d'intérêt à l'aspect d'écrans partagés. Le classement n'est pas donné en temps réel, ce qui pourrait être intéressant à proposer dans notre cas en cas d'un nombre de joueurs important.

Sujet: Piccadilly Game

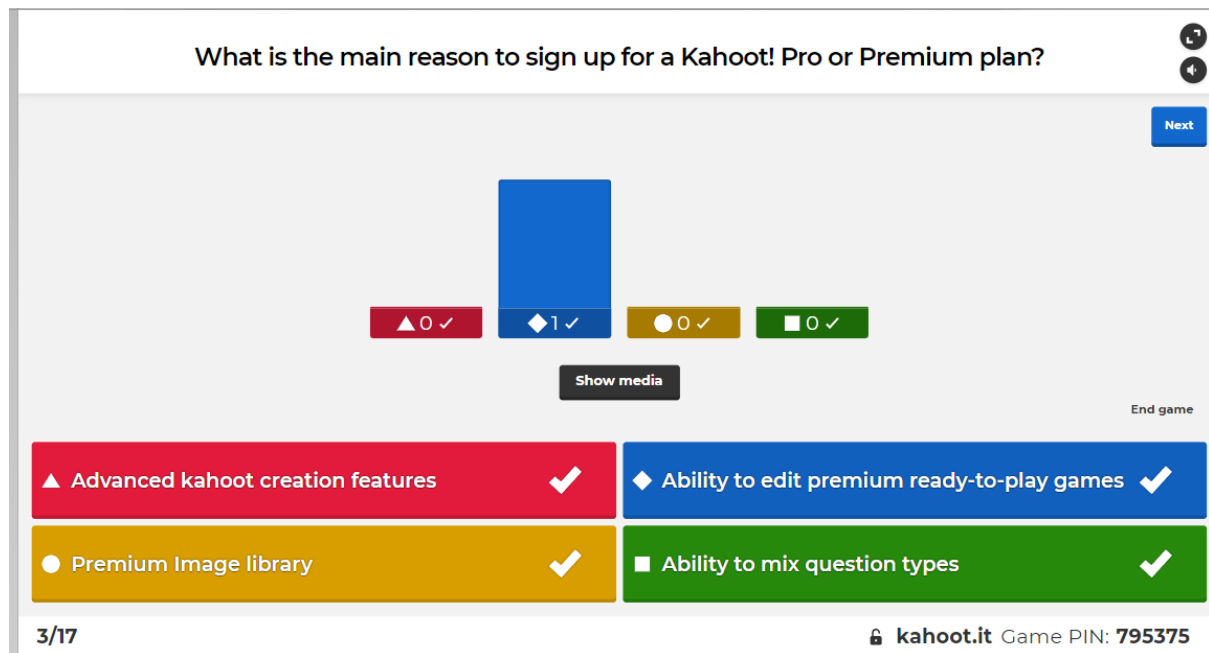
Kahoot est comme Wooclap: une plate-forme d'apprentissage utilisée dans le cadre éducatif, ses jeux d'apprentissage sont des questionnaires qui permettent à plusieurs utilisateurs de jouer simultanément, le site est accessible via navigateur web sur ordinateur et appareil mobile. Le site se compose de deux types de personnes: le maître de la partie (ou écran principal) ainsi que les différents joueurs. Contrairement à Wooclap, il y a une énorme différence entre écran principal et l'écran des joueurs. En effet, l'écran principal possède cet affichage:



Et les joueurs possèdent sur leur écrans uniquement les symboles et couleurs liés aux réponses et ils doivent alors appuyer sur le bouton qui correspond à la réponse qu'ils ont choisi.



Une fois que tous les participants ont répondu ou que le temps est écoulé, il est possible de visualiser l'écran des réponses qui indiquent là où était la bonne réponse, ainsi que le nombre de personnes ayant sélectionné une réponse.



Notre projet est différent de Kahoo au niveau du ton du site, là où Kahoo est un site d'apprentissage proposant un service au professeur et aux élèves, notre projet quant à lui aura un ton plus axé sur la compétition et se rapprochera plus d'une émission comme questions pour un champions ou qui veut gagner des millions dans le cas de l'idée du quizz. Cette différence nous permet aussi d'aborder plus de thèmes que le site (dans le cas des sujets de questions par défaut).

Enfin, un autre produit intéressant serait l'application Twitch Play, il peut s'apparenter à notre projet car le jeu se déroule sur un écran principal qui serait un Stream avec les joueurs qui le commande en tapant des messages dans le chat. Pour la plupart des jeux, tout se passe en temps réel et la commande est choisie au hasard dans le chat. Mais pour certains jeux plus compliqués comme Dark Souls une pause de 30 secondes a été ajoutée et dans ce contexte c'est l'option la plus choisie qui sera exécutée.

Exemple de l'utilisation de cette application: <https://www.youtube.com/watch?v=5-cJJD070U>

Cependant, ce système est plutôt compliqué à mettre en place, mais surtout il n'est pas utilisable par tout le monde dans le sens où n'importe qui ne peut pas lancer une partie (s'il ne possède pas de jeu adapté ou un pc capable de tout supporter). De plus, il y aura une latence assez importante dû à la diffusion en direct, ce qui rend parfois compliqué le gameplay quand il s'agit de commandes précises à effectuer. Enfin, les jeux supportés par ce produit sont des jeux déjà existants, nous pourrions plutôt développer un jeu original et plus adapté à ce type de gameplay (ne pas nécessiter de commandes trop précises, proposer des boutons à sélectionner plutôt que des commandes à taper pour rendre le jeu plus simple à comprendre...).

Sujet: Piccadilly Game

Comme dit précédemment Node.js nous semble être la meilleure option pour un projet multi-joueur demandant des interactions en temps réel. L'utilisation de Node.js a plusieurs avantages en commençant par le fait qu'il est facile d'apprendre à l'utiliser dans le cadre de nos cours de javascript que nous avons eu en S2. Nous ne partons donc pas de zéro. De plus, cette technologie est réputée pour sa rapidité à effectuer diverses actions avec le serveur et gérer plusieurs connexions simultanées. Ensuite la communauté utilisant Node.js est très active et il existe donc de nombreux tutoriels et librairies pour nous aider à mener à bien notre projet. Cette technologie possède donc une grande extensibilité qui permet d'intégrer de nombreux outils comme différentes API par exemple. Node.js est vraiment une technologie de plus en plus utilisée sur le marché est c'est la technologie qui a le plus de chance de devenir La référence dans le monde du développement web.

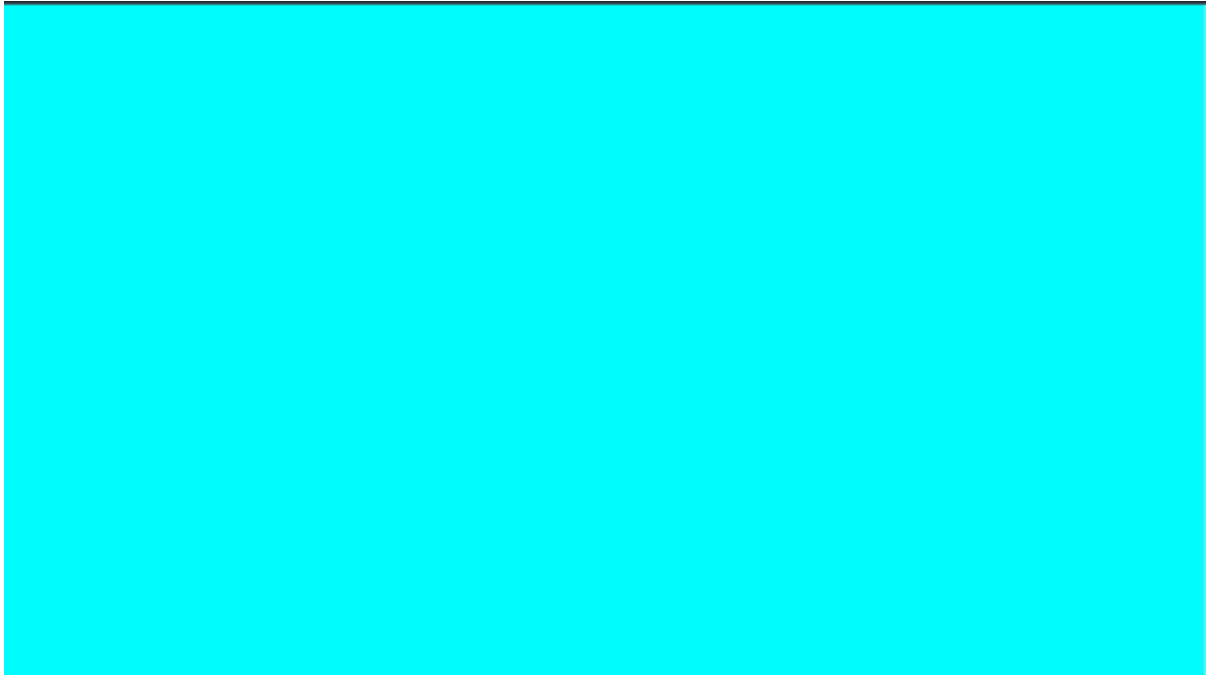


Pour ce qui est de la création du jeu en lui-même nous avons différentes solutions qui s'offrent à nous. Tout d'abord la technologie "Canvas" est un composant du langage Web HTML5 qui permet d'effectuer des rendus dynamiques d'images bitmap en utilisant des scripts dans des langages tels que le javascript. Nous avons réalisé plusieurs petits tests pour découvrir le potentiel de Canvas.

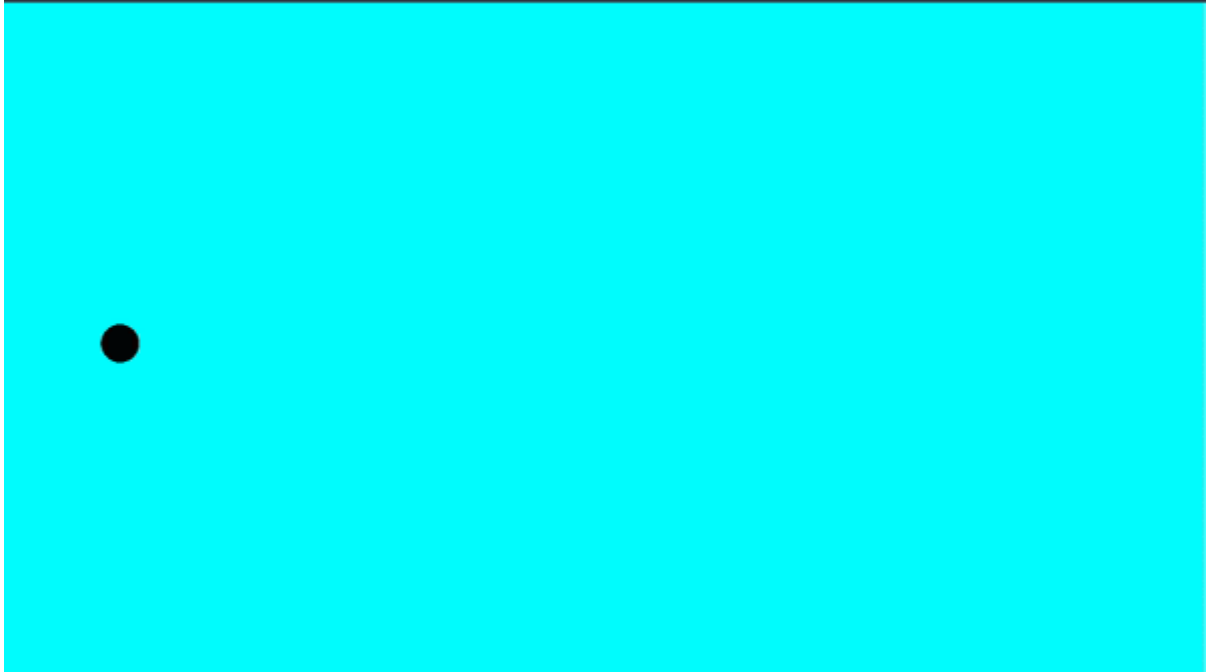


Par exemple, ici nous avons créé un cercle qui suit les mouvements de la souris.

Sujet: Piccadilly Game

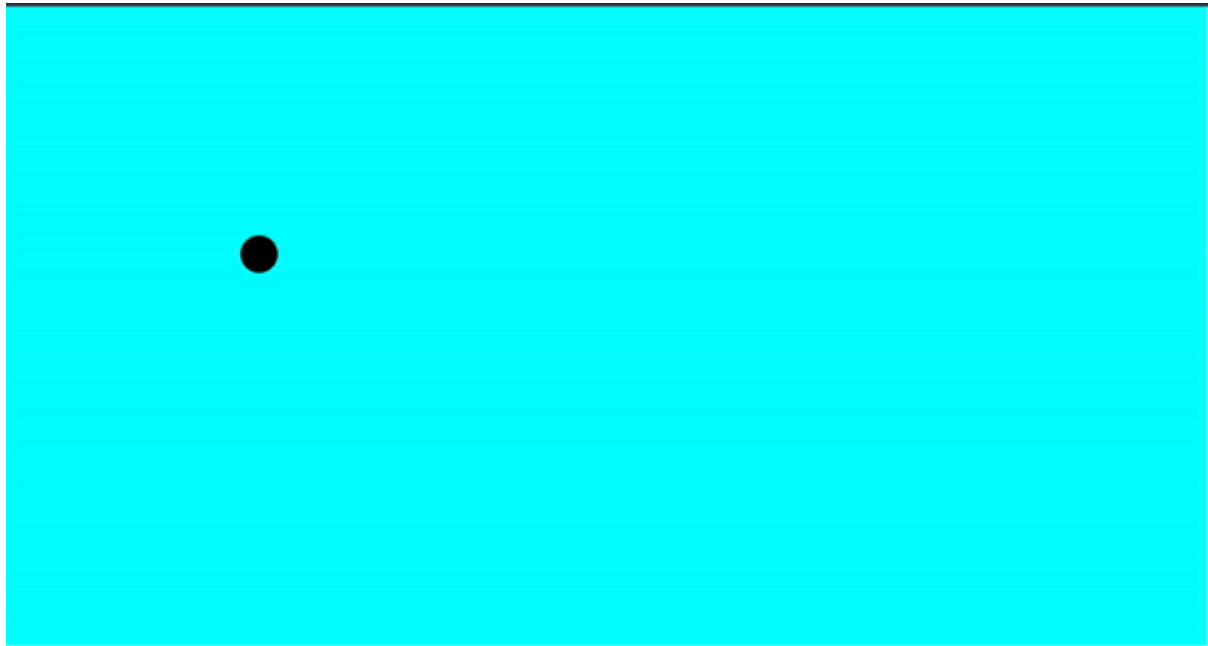


Ou encore un cercle qui fuit la souris en se déplaçant à un endroit aléatoirement si la souris rentre en collision avec le cercle.

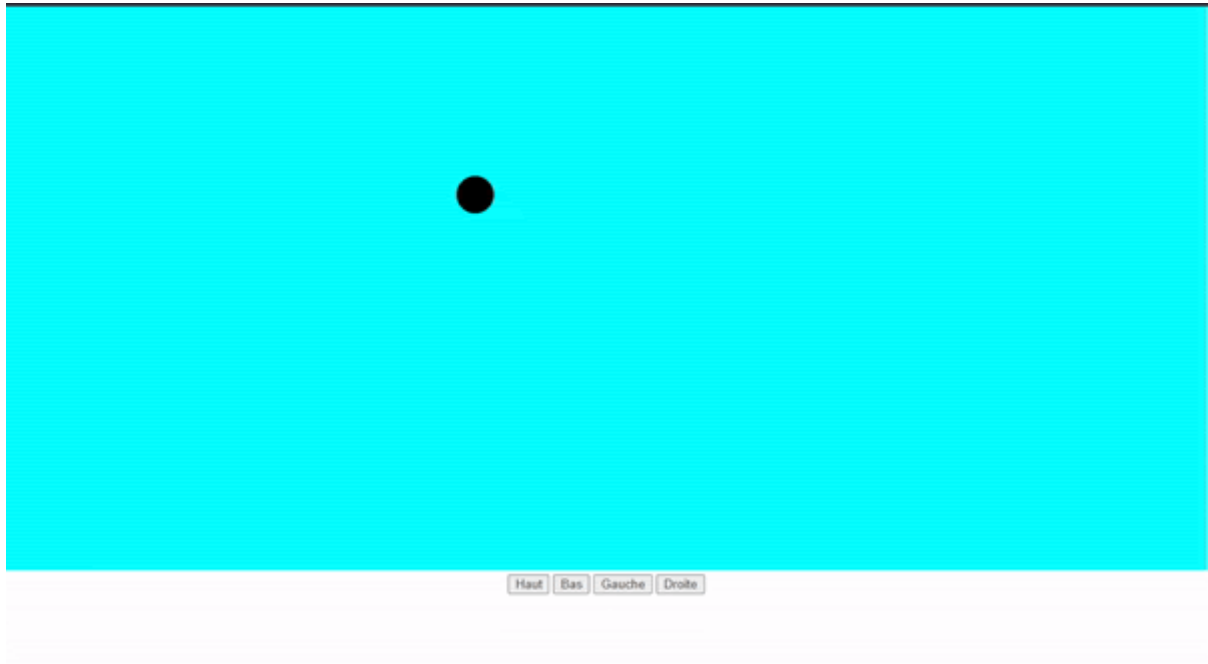


Ensuite nous avons créé un petit jeu test dont le but est de cliquer sur le cercle pour le détruire et un autre cercle apparaît à un autre endroit aléatoire. En plus de cela il y a un compteur de chaque cercle détruit par le joueur.

Sujet: Piccadilly Game



Enfin nous avons créé un Canvas qui permet de déplacer le cercle avec des boutons "haut", "bas", "gauche", "droite". Nous pouvons également le faire facilement avec les flèches du clavier.



La technologie Canvas paraît alors être un bon moyen pour réaliser différents jeux que nous pourrions imaginer et elle paraît à l'abord être simple à prendre en main.

Un problème qui sera sûrement au cœur de notre projet de par l'aspect temps réel est la latence et plus particulièrement la différence de latence entre les joueurs. En effet cela donnera des avantages à ceux qui ont une meilleure connexion et cela n'est pas souhaitable. Nous étudierons cet aspect plus en profondeur dans la suite de notre projet mais actuellement la solution qui semble la plus simple à mettre en place et qui permettrait de réduire cette différence de latence serait la "latence négative".

Sujet: Piccadilly Game

Le but est de deviner les actions des joueurs avant qu'il ne les fasse. Cela va permettre de gagner de précieuses millisecondes et ainsi créer une illusion d'immédiateté ce qui rendra le jeu d'autant plus agréable à jouer et cela pénalisera moins les joueurs ayant une connexion plus faible que leur opposant.

Pour ce qui est de l'hébergement du serveur étant capable d'assumer la connexion de plusieurs joueurs simultanément, nous comptons pour commencer utiliser un Raspberry pi.



En effet je dis pour commencer car si on imagine que notre plateforme prenne de l'ampleur, un Raspberry ne suffira probablement pas et il faudra probablement faire héberger votre site sur une plateforme dédiée ou bien grâce à un Nas par exemple.



En ce qui concerne la partie Software, le serveur web semble être une bonne solution du fait de sa popularité. Apache est un logiciel de serveur web gratuit et open source et nous permettra de mettre facilement notre site en ligne et du fait de sa popularité les tutoriels ne manquent pas. Le rôle d'un serveur web consiste à servir des sites web sur internet. Pour atteindre cet objectif, il agit comme un intermédiaire entre le serveur et les machines des clients. Il extrait le contenu du serveur sur chaque requête d'utilisateur et le transmet au web.

