

Alexis MARCEL

Lilian LEBLANC

Justine PRULIERE



Agonline

Sujet S14: Piccadilly Game

Tuteur: Pierre-André Guenego

Table des matières

Introduction	3
Objet du document et présentation du projet	3
Présentation de l'équipe et rôles de chacun	3
Planning.....	3
Analyse.....	4
Découpage fonctionnel du projet et modèles UML utilisés.....	4
Fonctionnalités générales.....	4
Le chat	4
Accès aux salles de jeu.....	5
Choisir un type de jeu	6
Première fonctionnalité majeure: le quiz	7
Seconde fonctionnalité majeure: ProjectX	9
Troisième Fonctionnalité majeure: Survival	10
Ajout d'un QR code pour accéder aux salles	12
Évolutions depuis l'étude préalable.....	12
Évaluation des risques	12
Réalisation	14
Architecture logicielle.....	14
Tests de validation.....	15
Difficultés rencontrées	15
Conclusion	16
Liens vers les ressources complémentaires	16

Introduction

Objet du document et présentation du projet

Dans le cadre de nos études en IUT informatique, nous avons dû réaliser un projet tutoré tout au long de notre seconde année. Ce document fait office de rapport final et détaillera les différentes avancées et technologies mises en place tout au long du projet.

Notre groupe a dû réaliser un « Picadilly Game ». Ce sujet a pour objectif de créer un site web interactif permettant de lancer divers mini-jeux sur un écran principal (écran de type ordinateur ou plus grand de préférence) qui pourra être rejoint par n'importe qui via un code ou qr code. Chaque personne rejoignant le mini jeu pourra interagir sur l'écran principal avec son écran qui sera de préférence un écran de smartphone.

Présentation de l'équipe et rôles de chacun

Notre équipe est composée d'Alexis Marcel, Lilian Leblanc et Justine Pruliere. Alexis s'est principalement orienté vers le développement des interactions serveur/client avec notamment la création du quizz, d'un chat, la gestion de connexion aux salles de jeux, le front-end de l'application ainsi que la mise en place d'un jeu en HTML5. Il s'est également chargé de la correction et l'optimisation du code. Lilian s'est chargé de la mise en place du timer, de la sélection de réponses et de la vérification de la réponse dans le quizz, il a également créé la partie serveur et client, donné une couleur aux sprites des joueurs et mis en place un début de jeu pour ProjetX. Justine quant à elle a pris du retard sur le développement en apprenant les langages. Elle a cependant mis en place le QR code permettant d'accéder à une room de jeu. De plus, elle a participé à la finalisation du jeu ProjetX en collaboration avec Lilian comprenant la mise en place d'obstacles à éviter, d'un système de victoire et de la mise en place d'un joystick sur l'écran client.

Planning

Ce projet s'est déroulé en trois sprints: le premier aura permis à l'équipe de décider des meilleures technologies à utiliser, établir un plan de conception de la plateforme ainsi que la découverte des technologies utilisées en créant une première ébauche d'un chat et du quizz. Le second marquera la fin du développement du quizz et du début d'un jeu multijoueur en HTML. Durant le sprint final, le premier jeu en HTML « ProjetX » sera finalisé et un autre jeu, « Survival », sera mis en place. Dans cette dernière partie, une amélioration de l'expérience utilisateur sera mise en place à la suite de tests avec des personnes extérieurs aux projets. Notamment, une meilleure gestion de connexion à une salle de jeu avec l'ajout d'une file d'attente.

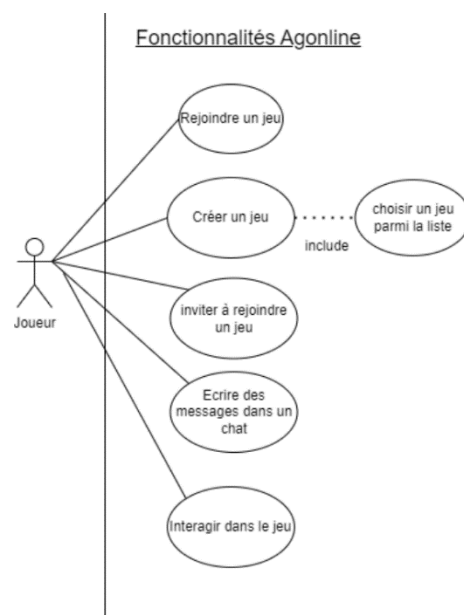
Analyse

Découpage fonctionnel du projet et modèles UML utilisés.

Comme évoqué plus haut, notre projet s'est découpé en trois sprints. Ceux-ci nous ont permis de mettre en place divers mini-jeux pour compléter notre « Picadilly Game ». Nous allons donc détailler les fonctionnalités que nous avons choisi de réaliser tout au long de celui-ci.

Fonctionnalités générales

Dans un premier temps, nous avons établi un diagramme des fonctionnalités générales que nous comptons mettre en place sur notre site:

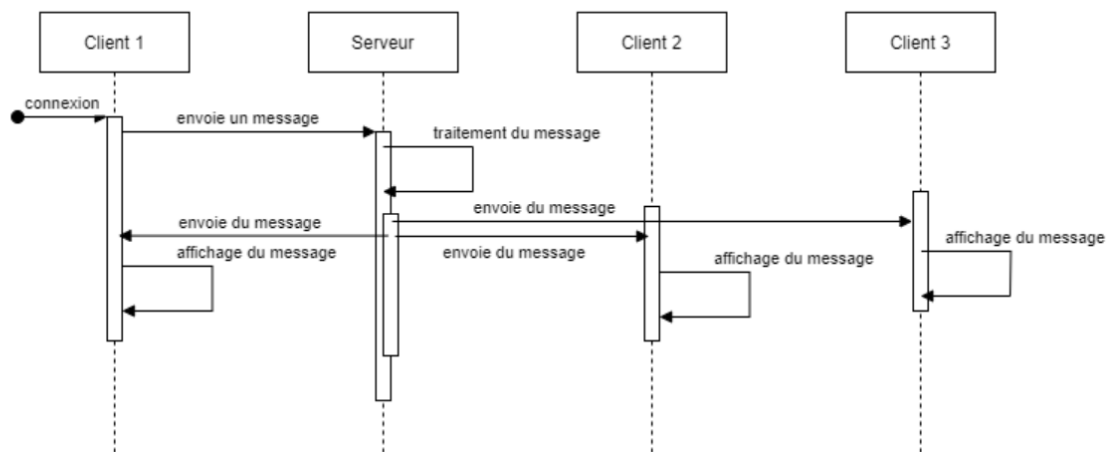


Ainsi, il est prévu que l'utilisateur puisse rejoindre ou créer un jeu, qu'il puisse inviter à rejoindre une partie qu'il aurait créée, écrire des messages dans le chat ou encore interagir dans le jeu (c'est-à-dire répondre aux questions ou faire bouger son personnage en fonction du jeu choisi).

Le chat

Avant de commencer à proposer des jeux, notre équipe a voulu s'entraîner à développer avec les technologies choisies et a donc décidé de réaliser un chat qui sera ensuite utilisé dans chaque jeu afin de permettre à tous les utilisateurs de communiquer entre eux. Cette fonctionnalité permet de rajouter une dynamique d'interaction entre les utilisateurs au-delà de la compétition dans le jeu.

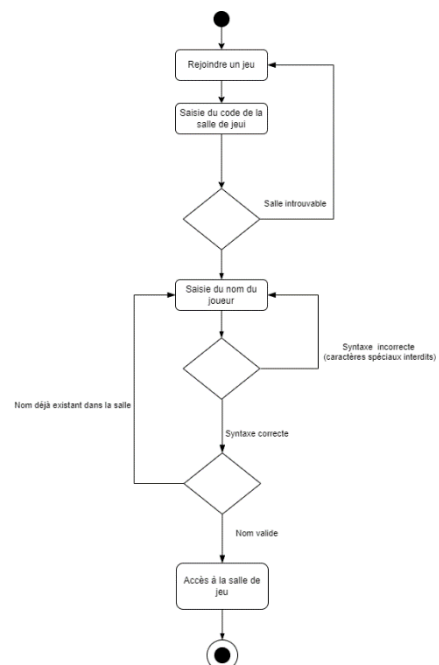
Envoi d'un message de Client 1 via le chat aux autres Clients



Un premier client envoie son message, qui sera reçu et traité par le serveur avant de le renvoyer à tous les participants qui l'afficheront. Un changement d'affichage est fait lorsque le maître du jeu envoie un message, le différenciant des autres utilisateurs.

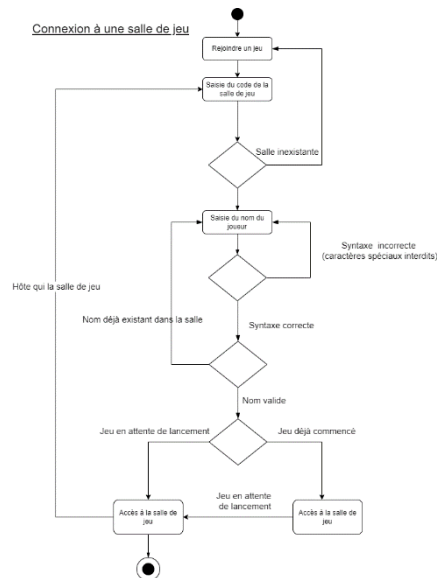
Accès aux salles de jeu

Une fois le chat en place et fonctionnel, il nous a fallu réfléchir à la mise en place des salles de jeu et de leur fonctionnement, pour nous aider nous avons donc réalisé quelques diagrammes.

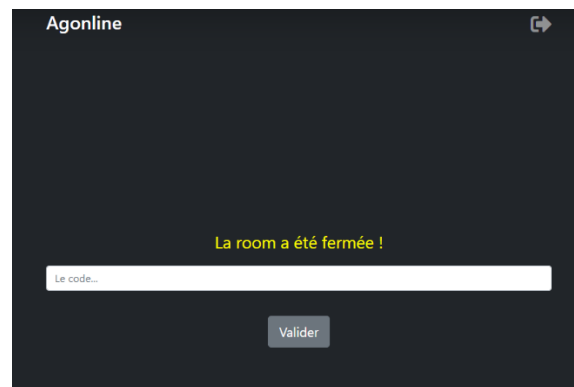


Ces diagrammes nous ont permis d'éclaircir les interactions client/serveur ainsi que les problèmes pouvant être rencontrés par l'utilisateur lors de son arrivée dans une salle de jeu.

Par la suite, nous avons amélioré notre système de connexion à une salle en ajoutant plusieurs fonctionnalités. Tout d'abord, si le jeu est en cours de partie, l'utilisateur sera amené dans une file d'attente et sera automatiquement redirigé dans la salle de jeu quand la partie sera finie. De plus, si l'hôte de la partie (maître du jeu) quitte la salle, alors tous les joueurs seront redirigés vers le menu de connexion à une salle.



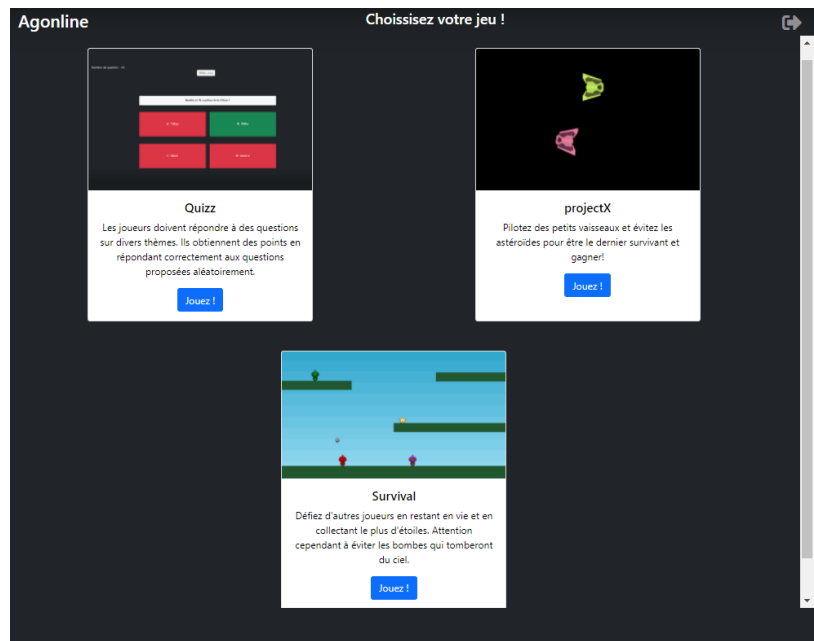
Placement en file d'attente si la partie est déjà lancée.



Fermeture d'une room après le départ du maitre du jeu.

Choisir un type de jeu

Nous avons créé plusieurs mini-jeux différents. L'utilisateur peut choisir de créer une salle correspondante à un jeu qu'il pourra choisir avant de créer la salle.

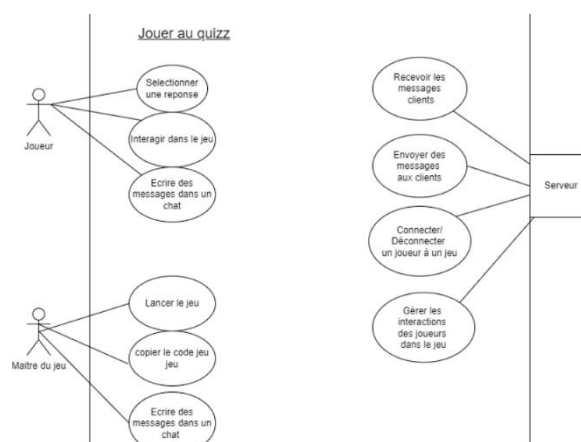


L'utilisateur rejoindra ensuite une salle qui correspondra à un type de jeu et sera redirigé vers le menu de joueurs associé au jeu.

Première fonctionnalité majeure: le quizz

La mise en place des salles étant maintenant claire, nous avons pu commencer le développement de notre premier jeu, afin de se familiariser encore avec les aspects serveur/client, nous avons choisi de nous orienter vers un quizz, qui nous semblait la meilleure option pour une prise en main tout en avançant dans nos fonctionnalités. L'avantage de débiter avec le quizz est que ce sont des interactions simples et peu nombreuses à l'inverse de créer un jeu visuel fonctionnant avec plusieurs requêtes par seconde.

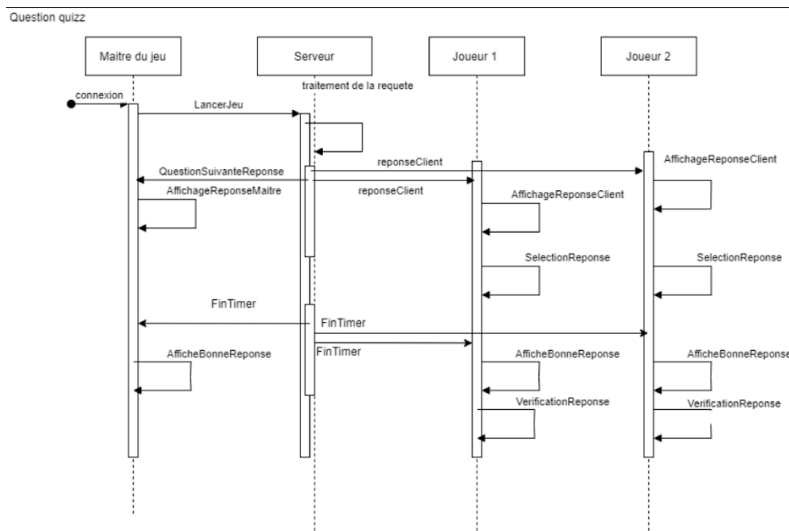
Afin de nous orienter, nous avons mis en place un premier cas d'utilisation pour notre quizz à venir.



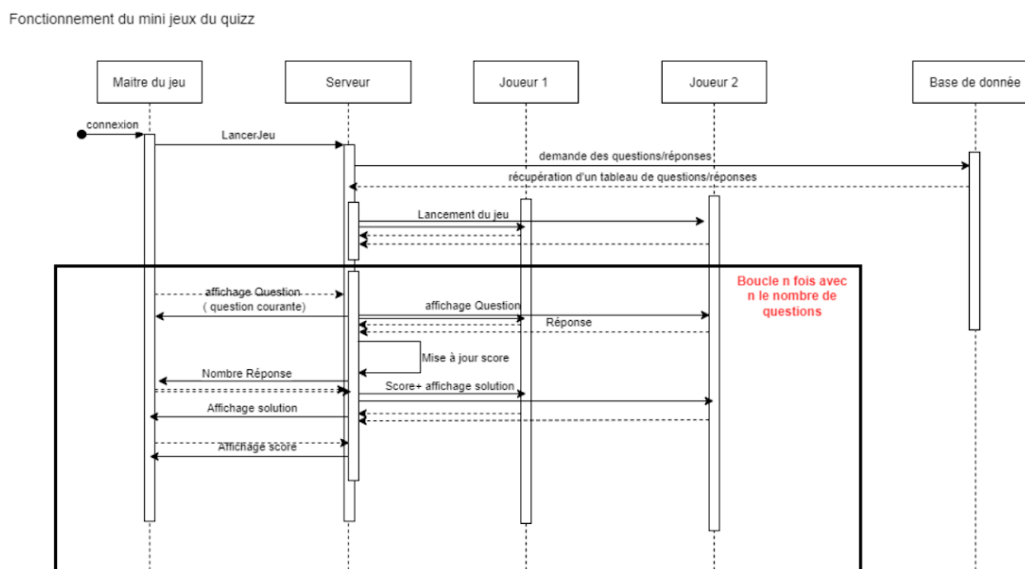
Ainsi, le client (le joueur) pourrait répondre aux questions en sélectionnant sa réponse et écrire dans le chat, pendant que le maître du jeu pourrait choisir de lancer la partie,

de partager le code de jeu afin de permettre à d'autres joueurs de rejoindre mais également d'écrire dans le chat. Pendant toute la durée de la partie, le serveur serait sollicité pour recevoir et afficher les messages des utilisateurs, gérer les connexions et déconnexions ainsi que gérer les interactions des joueurs, c'est-à-dire traiter les réponses reçues et mettre les scores à jour.

Nous avons ensuite proposé un premier diagramme de séquence afin de mieux visualiser comment structurer notre code.



Nous avons ensuite amélioré la version du quizz en ajoutant plusieurs fonctionnalités.



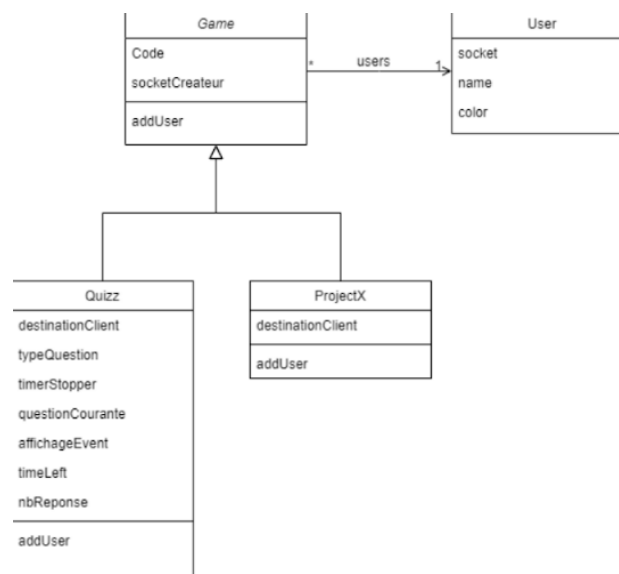
Le quiz final consiste donc au lancement du jeu par le créateur de la room (le maître du jeu), le serveur ira ensuite récupérer les questions/réponses dans la base de données avant de lancer le jeu pour les participants. Ensuite, une boucle répétée n fois (soit n le nombre de questions disponibles) se lance dans laquelle les questions sont affichées pour les joueurs. Après le temps écoulé, ou que tous les utilisateurs ont répondu, la solution s'affiche et les scores sont mis à jour en fonction des réponses données. Enfin vient l'affichage du score. La boucle recommence ainsi de suite en passant à la question suivante.

De plus, nous avons différents thèmes de questions qui peuvent être choisis au moment de la création de la salle de jeu, afin de permettre à l'utilisateur de pouvoir varier ses parties ou choisir son domaine de prédilection.



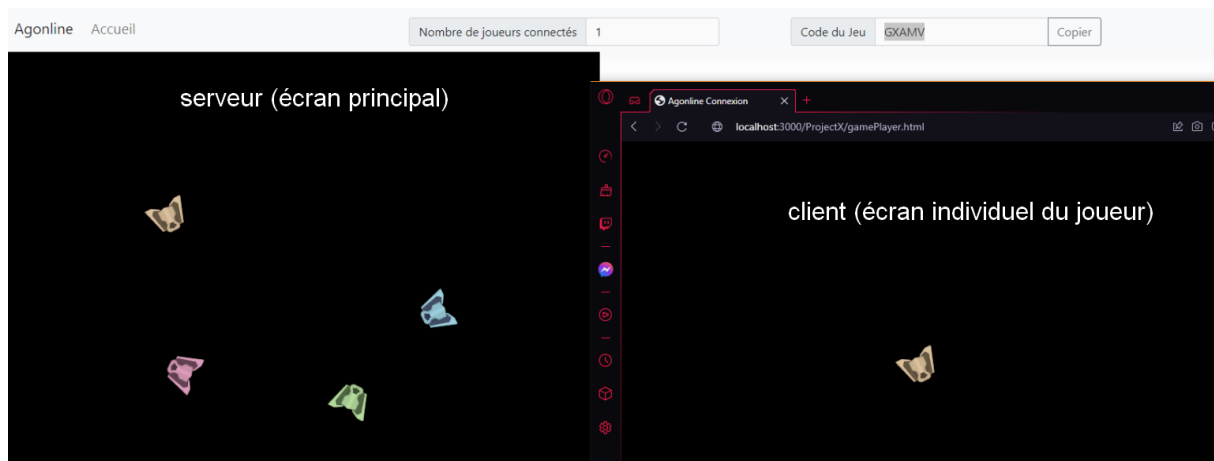
Seconde fonctionnalité majeure: ProjectX

Pendant qu'une partie du groupe finissait d'améliorer le prototype de quizz pour en faire une version finie, l'autre commençait la mise en place d'une nouvelle fonctionnalité. Nous avons donc ajouté un nouveau type de jeu à la version existante du serveur.



Nous nous sommes lancés vers un jeu demandant beaucoup plus d'interactions entre le serveur et les clients.

Le premier jeu ajouté se nomme « ProjectX », chaque client possède un vaisseau qu'il peut contrôler depuis son écran et qui s'affiche sur l'écran principal avec tous les autres.



Captures d'écran du jeu ProjetX.

Le principe du jeu est d'éviter des astéroïdes qui éliminent le joueur lorsqu'il le touche, le dernier participant qui ne s'est pas fait toucher sera le gagnant.

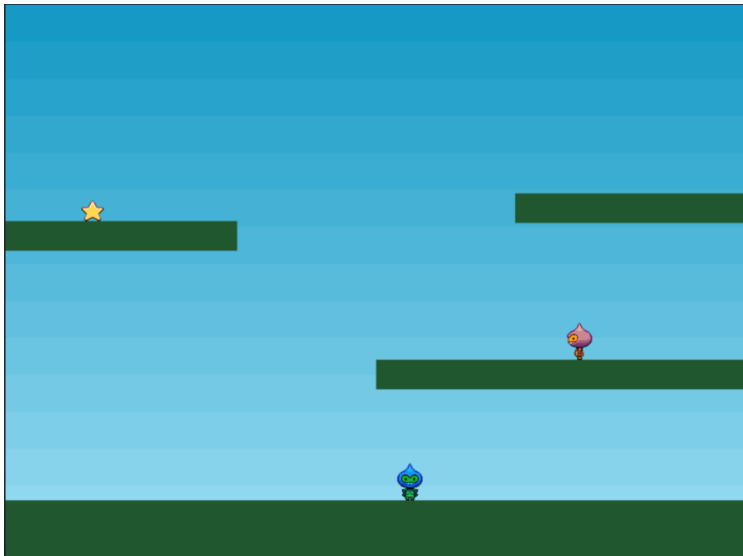


Partie de ProjetX avec les astéroïdes.

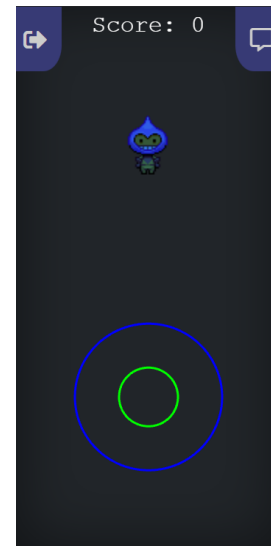
Au niveau des interactions, le maître du jeu crée dans un premier temps la partie, qu'il ne pourra lancer qu'une fois qu'au moins deux joueurs auront rejoint. Pendant ce temps, les clients (joueurs) attendent le début de la partie. Une fois le jeu lancé, l'affichage s'actualise et le jeu apparaît comme représenté au-dessus. Les astéroïdes commencent à apparaître, un certain nombre toutes les 3 secondes, que les joueurs doivent éviter. Si un joueur est touché, il perd: son vaisseau devient rouge et il ne peut plus bouger. Lorsqu'il ne reste plus qu'une seule personne en vie, un message s'affiche indiquant le pseudo du gagnant.

Troisième Fonctionnalité majeure: Survival

Le troisième jeu quant à lui se nomme « Survival », c'est un jeu dans lequel les joueurs doivent diriger un personnage en lui faisant esquiver les bombes. Pour améliorer son score, le joueur devra rester le plus longtemps en vie et ramasser un maximum d'étoiles qui rapportent des points bonus.



Capture d'écran de l'écran principal du jeu Survival.

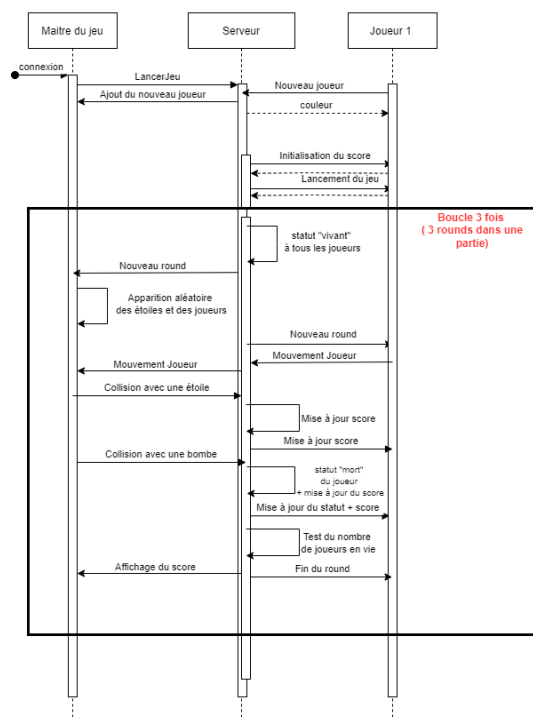


Capture d'écran de l'écran client du jeu Survival.

L'écran du client contient un joystick qui lui permet de déplacer son personnage, ainsi que la prévisualisation de son personnage qui lui permettra de mieux s'identifier sur l'écran principal. L'avantage du joystick est qu'il s'adapte à tous les supports et qu'il est facile d'utilisation.

Côté écran principal, les personnages ont des collisions avec les autres joueurs mais également avec les plateformes et bordures de la carte. Ils auront également des interactions avec les étoiles qu'ils pourront ramassés et les bombes qui les fera "mourir". Ce jeu est représenté par le diagramme de séquence suivant.

Fonctionnement du mini jeux du survival



Le maître du jeu se connecte et crée la room, une fois que tous les joueurs ont rejoint (sur le diagramme, seul un joueur est représenté, mais il peut y en avoir autant que voulu) il peut lancer la partie. Le serveur lancera alors le jeu pour tous les participants en leur mettant le statut vivant puis le round démarrera (3 rounds donc 3 boucles) durant lequel les joueurs pourront ramasser des étoiles, devront éviter les bombes afin d'augmenter leur score. Si un joueur touche une bombe, il ne peut alors plus se déplacer et sa couleur change. Le serveur lui attribuera le statut de "mort". Lorsqu'il ne reste plus qu'un joueur en vie, le round s'arrête en affichant le nom du vainqueur du jeu et le tableau des scores finaux est affiché, la boucle reprend ensuite jusqu'à la fin du nombre de rounds maximum.

Le score est calculé en fonction de 3 variables :

- Le temps passé en vie
- Le nombre d'étoile ramassées
- Si le joueur est le dernier en vie ou non

Ajout d'un QR code pour accéder aux salles

Enfin, il est possible de rejoindre les salles de jeu à l'aide d'un code présent en haut de la page. Comme notre site est fait pour être rejoint depuis n'importe quel support, nous avons pensé que mettre en place un QR code qui donne un accès direct à la salle voulue serait plus pratique pour les utilisateurs de smartphones.



Évolutions depuis l'étude préalable

Notre vision et les fonctionnalités souhaitées n'ont pas changé depuis l'étude de décembre. Nous avons pu mettre en place toutes les fonctionnalités que nous souhaitions et avons évoquées. Le principal changement aura été la refonte graphique du site avec Bootstrap afin de le rendre responsive pour tous les supports. Ainsi que le choix du support d'hébergement du site.

Évaluation des risques

Le risque principal de ce projet serait la latence, en effet, celui-ci étant basé sur un concept de multi joueurs massif, si des joueurs se retrouvent pénalisés à cause de leur connexion, alors cela affectera leur expérience de jeu voire le rendra injouable pour certains. Il faudrait donc prévoir ceci afin de rendre le jeu praticable pour tous joueurs quel que soit la qualité de leur connexion.

Le second risque serait que le produit final ne plaise pas au public cible, il faudrait alors revoir les jeux proposés afin de pouvoir répondre à leurs attentes. De plus, il faudrait pouvoir proposer un nombre de jeux suffisants pour satisfaire les envies de chaque utilisateur, s'ils se retrouvent sans jeux répondant à leur envie, cela pourrait nuire à leur expérience. Il faudra également s'assurer de proposer des jeux originaux ne ressemblant pas trop à l'existant afin d'offrir une nouvelle expérience à l'utilisateur.

Enfin, il faudra assurer le bon fonctionnement du serveur afin que celui-ci puisse supporter le nombre de joueurs variables sans risquer de crash.

Réalisation

Architecture logicielle.

Notre projet étant un site internet, nous l'avons donc développé en HTML5 et utilisé le Framework Bootstrap afin d'avoir un CSS responsive pour tous les supports.

La gestion des interactions client/serveur quant à elle a été réalisée avec le Framework NodeJS qui était pour nous la meilleure option car nous avons déjà appris l'utilisation de javascript durant nos cours précédents, nous ne partions donc pas de zéro pour l'apprentissage de l'utilisation de ce Framework. 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 bibliothèques 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. Son utilisation nous a donc permis de mettre en place des interactions fonctionnelles et rapides.

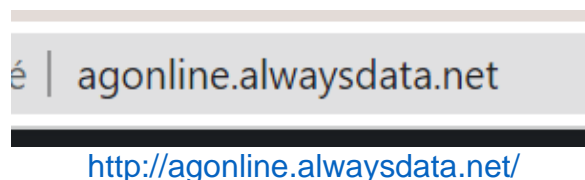
Nous avons intégrés à NodeJs plusieurs bibliothèques tel que socket-io permettant la communication entre le serveur et les clients en temps réel indispensable pour la création de notre application. De plus nous avons utilisé express nous permettant de créer un serveur fonctionnel.

L'utilisation de nodemon a également été très utile. En effet, cette bibliothèque permet de redémarrer le serveur et de le mettre à jour à chaque changement. Cela nous a fait gagner beaucoup de temps pendant le développement.

Ensuite, pour les jeux en HTML, nous avons choisi d'utiliser la bibliothèque Phaser qui est un moteur de jeu, ce qui nous permet de proposer des jeux plus fluides. Toute la documentation étant disponible en ligne, elle était facile à prendre en main, de plus, d'autres groupes ayant travaillé avec, nous avons pu en apprendre plus ou leur demander en cas de problèmes.

Pour la base de données, nous avons utilisé Mysql, un puissant système de gestion de base de données. De par nos précédents projets nous avons déjà utilisé Mysql, ce qui nous a permis de pouvoir créer une base de données simplement.

Pour l'hébergement, nous pensions d'abord utiliser un Raspberry Pi, mais nous nous sommes finalement orientés vers l'utilisation du site « alwaysdata », un site qui nous permet d'héberger notre jeu gratuitement (tant qu'il ne dépasse pas une certaine taille).



Tests de validation.

Fonctionnalité	Démarche	Résultat attendu	Fonctionnel?
Chat room	Ouvrir l'onglet du chat. Envoyer des messages qui seront affichés pour tous les participants de la salle.	Les messages s'affichent correctement pour tout le monde.	oui
Rejoindre salle de jeu	Entrer le code de la salle voulue. Choisir un pseudo.	La personne est ajoutée à la salle.	oui
Quiz	Les joueurs sont tous présents. Le maître du jeu lance la partie.	Les questions s'affichent. Les joueurs peuvent y répondre. Le score s'actualise et s'affiche à la fin du timer.	oui
Jeux HTML	Les joueurs sont tous présents. Le maître du jeu lance la partie.	Tous les joueurs sont affichés sur l'écran principal. Chaque joueur peut déplacer son personnage et peut aussi perdre. Les scores se mettent à jour et s'affichent à la fin du round.	oui
QR code	Le maître du jeu peut afficher un qr code qui permettra aux joueurs de rejoindre sa salle de jeu.	Le qr code redirige à la bonne salle.	oui

Difficultés rencontrées

La difficulté principale aura été la prise en main des nouveaux Framework que nous avons utilisés. De plus, la gestion de plusieurs interactions avec des salles différentes et des jeux différents a été un réel défi de conception et de développement. Le défi a été de créer un serveur unique pour différents jeux qui seront instanciés plusieurs fois tout en gardant une architecture claire et optimisée.

La seconde difficulté aura été la répartition et la synchronisation du travail au début du projet. Puis, avec l'utilisation de Trello et plus de communication, nous avons réussi à nous organiser. L'organisation de travail par réunion régulière a été un point clé de la réussite de notre projet.

Conclusion

Pour conclure, ce projet nous aura permis d'avoir un aperçu de ce que pourrait être la vie en entreprise dans une équipe affectée à un projet. Nous avons pu en apprendre plus sur le travail de groupe et nous avons pu découvrir de nouvelles technologies et langages de programmation afin d'enrichir nos connaissances. Ce projet nous a vraiment fait prendre conscience des possibilités infinies de l'informatique en collaboration avec la communauté.

Nous sommes satisfaits du rendu final et d'avoir pu proposer différents jeux fonctionnels. Il serait cependant possible d'améliorer ou de compléter certains points que nous n'aurions peut-être pas eu le temps d'aborder sur la durée du projet. Nous ne nous opposerions pas à la reprise de notre projet, mais toutes les fonctionnalités majeures (interactions client/serveur, gestion de différents jeux, gestion d'exceptions...) ayant déjà été réalisées, cette reprise apporterait peu à ses nouveaux contributeurs.

Cependant, si le projet venait à être repris, une amélioration possible serait la possibilité de mettre une salle de jeu publique et qu'elle s'affiche dans un menu prévu pour que n'importe quel utilisateur puisse la rejoindre, celle-ci peut être très intéressante.

Enfin, ce projet aura été le travail le plus complet que nous avons dû réaliser durant notre formation, nous permettant de découvrir de nombreux aspects (que ce soit du code ou du travail en groupe) qui nous seront utiles pour le futur.

Liens vers les ressources complémentaires

GitHub:

https://github.com/univ-lorraine-iut-charlemagne/S3A_S14_LEBLANC_MARCEL_PRULIERE

Site hébergé:

<http://agonline.alwaysdata.net/>