**MOBA With Unity**

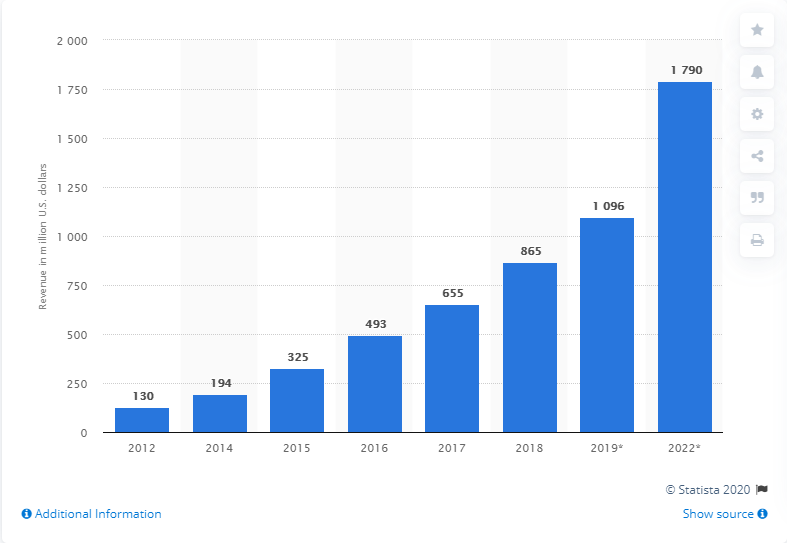
1. Expliqué clairement votre idée:

Mon idée est de faire un jeu sur Unity (C# avec le .NET framework). Le jeu sera de style MOBA (Multiplayer Online Battle Arena). C’est-à-dire, 2 équipes de 5 joueurs qui se battre contre pour emporter la victoire. Les joueurs se tuent, détruisent les objectifs adverses, tuent des monstres, affrontent des bosses, tout cela dans le but de détruire le ‘Nexus’. La première équipe qui détruit le ‘Nexus’ adverse, emporte la partie. C’est un jeu d’équipe, alors l’équipe s’entre aide pour arriver à ses buts ou même entreprendre diverses stratégies et tactiques pour démolir l’équipe adverse. Vous ramasser de l’argent en tuant des monstres, tuant l’équipe adverse, détruisant les objectifs ennemis. Avec cet argent, vous pouvez acheter des objets qui vous permettront d’être encore plus puissant! Il y a diverses personnages (champions) que vous pouvez choisir lors du début de la partie. Chaque personnage possède 4 habilitées uniques. Il est donc possible de choisir vos champions en équipe afin d’exécuter des synergies hors du communs! Mais à ne pas oublier, il faut aussi défendre ses propres objectifs.

1. Expliqué et prouvé la pertinence de l’idée

Cette idée est pertinente selon moi puisque tout d’abord, c’est un jeu. Nous sommes dans le programme de développement de jeux vidéo alors l’idée est en lien directe avec mon cheminement scolaire. Je crois que l’idée est aussi pertinente puisque le jeu sera réalisé sur Unity et nous avons eu 2 cours de Unity (Moteur de jeu I et Moteur de jeu II) durant nos 2 premières années dans ce programme. Le projet sera donc en lien directe avec ce que nous avons appris. Ensuite, Unity utilise le language de programmation C#. Ce sera pertinent puisque C# n’est pas seulement utilisé en programmation de jeux vidéo, mais aussi dans diverses compagnies pour différentes raisons comme la programmation d’applications, des statistiques, etc. L’idée est aussi pertinente selon moi puisque je n’ai jamais touché au multijoueur avec Unity. Nous n’avons pas touché le multijoueur avec l’école alors je trouve que c’est une bonne opportunité pour moi de m’y lancer et de l’apprendre. C’est très bon de savoir comment programmer du multijoueur car la majorité des jeux populaires actuels sur le marché sont multijoueur. Il est bien plus intéressant pour un joueur de jouer avec ses amis que toute seule. Il est donc évident que les jeux multi-joueurs ne cesseront pas d’exister et de dominer le marché des jeux vidéo. Unity permet aussi une exportation vers mobile (téléphone cellulaire) ce qui est très intéressant car la communauté de joueurs sur mobile est continuellement en croissance. Cela est due car le mobile est accessible pour pratiquement tout le monde, les enfants, les adultes, les personnes âgées, etc. L’idée est donc pertinente puisque cela me permettra de créer des jeux populaires qui accèdent à tout le monde. Finalement, je crois que l’idée est aussi pertinente car avec ce type de jeu, il est possible d’accéder au eSport (sport en ligne). Des joueurs passent leurs vies à jouer a des jeux pour devenir professionnel et y gagner leur vie. Si le jeu devient très populaire, alors il est possible que des joueurs y consacrent leurs vies, qu’ils y aillent des compétitions, etc. Ce qui est très rentables pour le créateur/la compagnie de jeux vidéo.

Voici le graphique du marché eSport ($) de 2012 à 2022 en million de dollars USD :



Source : <https://www.statista.com/statistics/490522/global-esports-market-revenue/>

1. Lister toutes les fonctionnalités

-Menus (login, main menu, settings, match history, champions, etc.)

-Lobby (création de lobby, rechercher un lobby, etc.)

-Jouer avec vos amis ou des inconnus (multijoueur)

-Tuer des monstres (minions)

-Tuer des bosses en équipe (dragon, baron, etc.)

-Acheter ou vendre des objets (magasin)

-Champions (beaucoup de champions différents qui permettent de faire des synergies uniques)

-Objectifs (les détruire, les protéger, etc.)

-Tuer l’équipe adverses (les autres joueurs)

-Récompenses de fin de partie

1. Discuter de votre choix de technologie

Pour cette idée, il y avait deux choix de technologies qui l’emportaient largement sur les autres : Unity (C#) et Unreal Engine (C++). J’ai décidé d’aller avec Unity. Pourquoi? Il y a plusieurs raisons. Tout d’abord, j’ai pensé aux utilisations futures et pour moi, il est plus utile de perfectionner le language C# que C++. Il est très fort improbable que je travaille dans Unreal Engine plus tard ou même avec le language C++ en général. Il y a plus de chances que je travaille avec C# que ce soit avec Unity pour les jeux que de la programmation de logiciels. Le C# avait plus d’avantages à long terme pour moi. Ensuite, Unreal Engine (C++) fournisse peu de documentation, voir presque zéro contrairement à Unity dont tout est documenté et on peut se procurer de l’information sur n’importe quel sujet qui attrait Unity. Cela ma aidé à prendre ma décision puisque le multijoueur est un sujet dont je ne connais pratiquement rien donc nécessairement je vais avoir besoins de documentation pour me supporter lors du développement de ce projet. Si j’utilise Unreal qui n’en fournit presque pas, je vais rencontrer des problèmes dont je ne suis même pas sûr que je vais pouvoir résoudre sans documentation. C’est donc pourquoi j’ai choisi Unity car je suis sûr que, même si j’ai des difficultés tout au long du projet, je vais être capable de me renseigner et de trouver les informations nécessaires pour moi d’avancer dans le développement. Une autre raison pourquoi j’ai choisi Unity est aussi le temps de développement. Unreal Engine peut fabriquer des jeux beaucoup plus puissants et beaucoup plus beaux graphiquement que Unity, mais il est extrêmement long pour arriver à ces résultats. Contrairement à Unity, on peut commencer un projet et en quelques jours avoir quelques choses à montrer à ses pairs, ce qui peut être très encourageant lorsqu’on apprend ou qu’on touche à quelque chose de nouveau. Une autre raison pourquoi j’ai choisi Unity est aussi le temps alloué pour ce projet. Nous avons quelques mois seulement pour réaliser ce projet et comme mentionné, pour arriver à des résultats avec Unreal Engine (C++), c’est très long. J’ai donc peur de ne pas avoir assez de temps pour être en mesure de fournir un résultat qui me conviendra. Finalement, après quelques recherches sur les engins, Unity travaillent beaucoup sur l’engin pour améliorer la programmation multijoueur et s’assurer que leur engin est bon pour programmer des jeux multijoueur. Ils font des mises à jour de l’engin continuellement pour améliorer leurs systèmes, contrairement à Unreal Engine, on ne voit pas trop de progrès au niveau multijoueur. C’est donc pour toutes ces raisons que j’ai choisi Unity C#.

1. Diagrammes

Pour une meilleure vision sur les diagrammes, ouvrir le dossier Diagrammes et vous pourrez consulter l’entièreté des diagrammes que ce soit en PNG ou sur le site :  
[www.draw.io](http://www.draw.io)   
Une fois sur le site, vous pouvez ouvrir les documents avec l’extension : .drawio

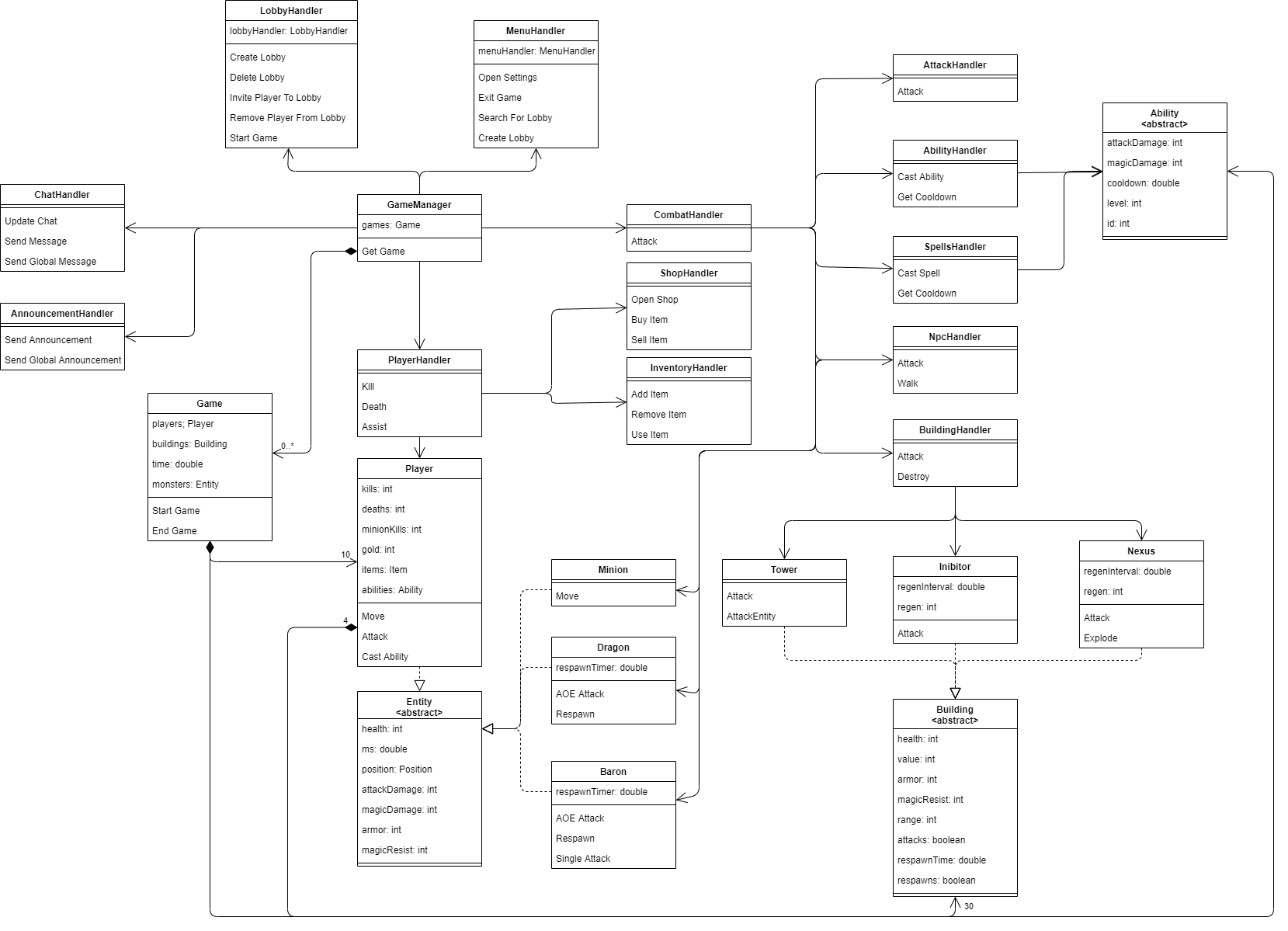
Diagramme de classes :

Diagramme de Use Cases :

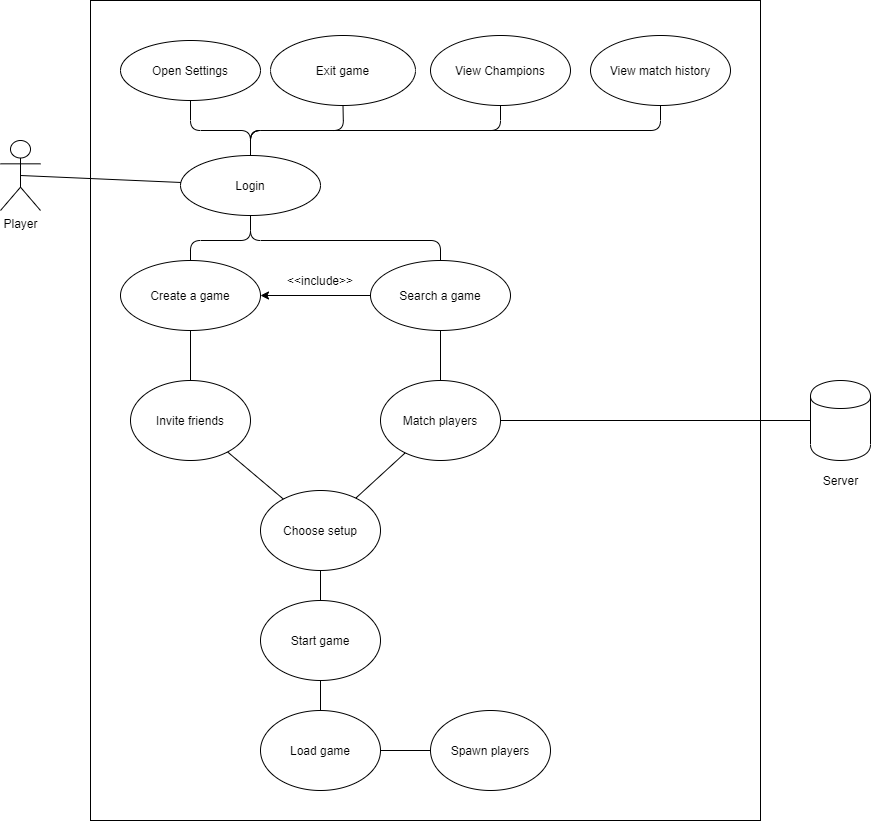


Diagramme Activité #1 :

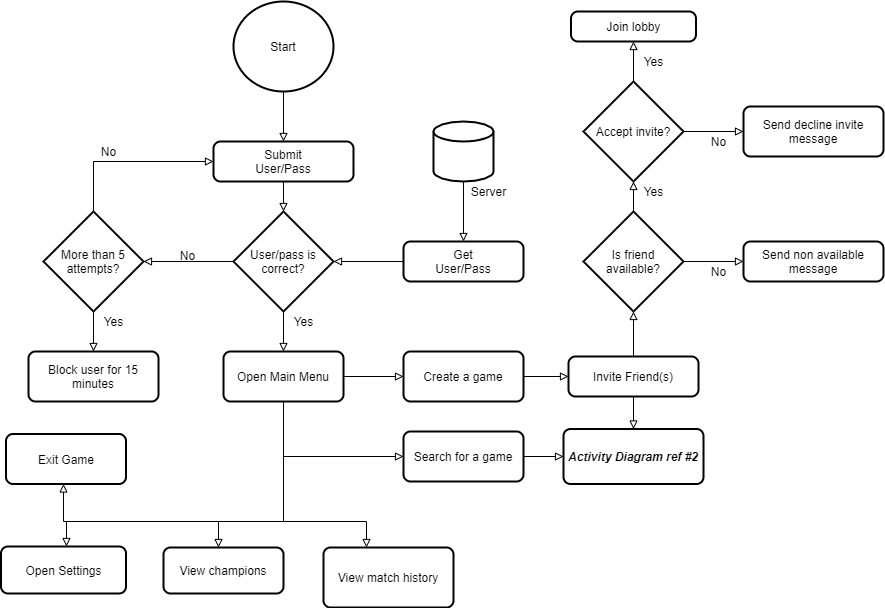


Diagramme Séquence #1 :

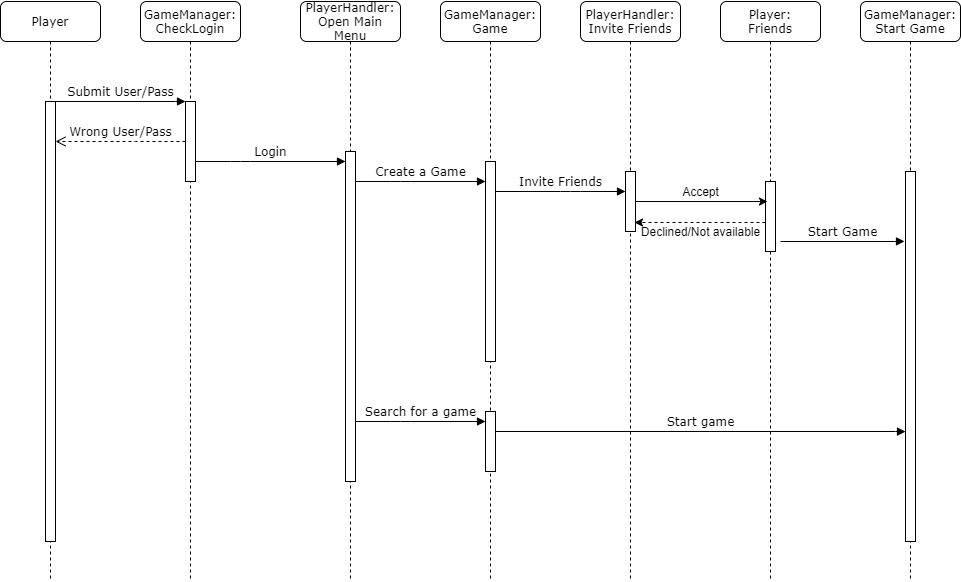


Diagramme Activité #2 :

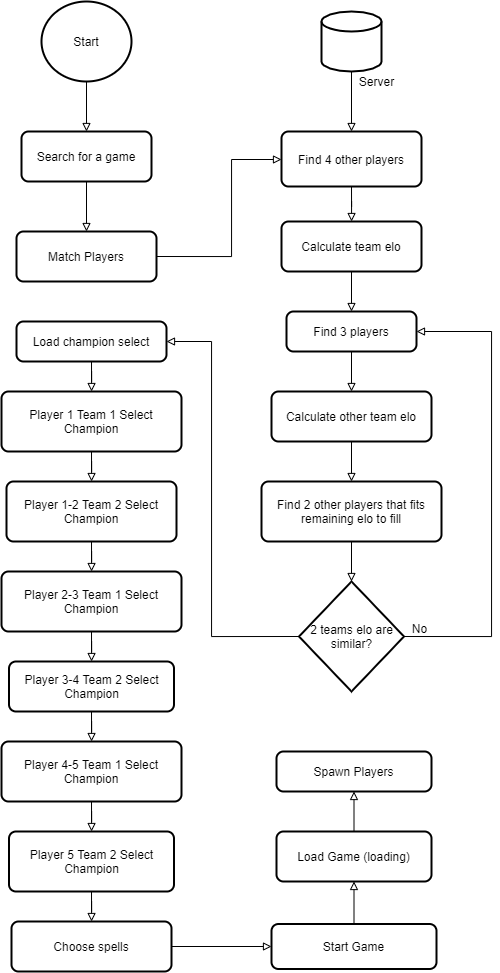


Diagramme Séquence #2 :

