

Projet Intégrateur
Groupe 1B



Cahier des Charges

Sommaire

- Introduction et présentation du jeu
- Les éléments clés sont bien adressés par le projet
- Choix des technologies utilisées
- Vision du rôle de chef de projet
- Tâches et équipes
- Diagramme de Gantt

Introduction et présentation du jeu

Carcassonne est un jeu de plateau pouvant se jouer de 2 à 5 joueurs et qui tire son nom de la célèbre ville française du Sud de la France reconnue pour sa cité médiévale. Le jeu est constitué de pions et de 84 petites plaquettes cartonnées, appelées tuiles. Ces tuiles peuvent représenter un ou plusieurs morceaux de terrain tels qu'un ou plusieurs chemins, un ou plusieurs morceaux de villes, une abbaye avec des parties de champs sur presque chaque tuile.

Déroulement d'une partie:

Chaque joueur possède 7 pions au début de la partie et une tuile choisie aléatoirement est posée sur le plateau. A chaque fois qu'un joueur joue :

- Il pioche une tuile et la pose sur le plateau,
- Il pose éventuellement un pion sur un morceau de terrain de la tuile,
- Il compte les points.

1^{ère} étape:

La tuile doit obligatoirement être placée de manière adjacente à au moins une autre tuile déjà posée en suivant certaines règles : par exemple, il ne faut pas mettre un morceau de ville à la suite d'un chemin. Cela permet de créer, prolonger ou terminer un chemin, créer, agrandir ou terminer une ville ou avoir les 8 tuiles nécessaires pour une abbaye. Si la pose n'est pas possible, alors le joueur remet la tuile en bas de la pioche et en pioche une nouvelle.

2^{ème} étape:

Lorsque le joueur pose un pion sur sa tuile, il en choisit l'emplacement pour conquérir une propriété (exemple : un chemin, une ville...). Si le joueur pose une tuile dans la continuité d'un chemin ou d'une ville alors qu'il y a déjà un pion d'un autre joueur, il ne pourra pas mettre son propre pion dessus.

3^{ème} étape:

Les points sont comptés de la manière suivante:

- Lorsqu'un chemin est complété (soit entre deux intersections, soit entre deux villes, soit entre une intersection et une ville), le joueur récupère son pion et compte un point par tuile qui constitue le chemin,
- Lorsqu'une ville est terminée, le joueur récupère son pion ainsi que deux points par tuile. Si une tuile a un écusson, alors il ajoute deux points supplémentaires.

Les abbayes sont des tuiles spéciales. Pour obtenir tous les points (9 points), il faut que l'abbaye soit entourée de huit tuiles. Cependant, lors de son tour, un joueur qui a un pion placé sur une abbaye, peut le récupérer même si les huit tuiles adjacentes ne sont pas toutes placées et ainsi récupérer le nombre de points correspondants aux tuiles placées avec la tuile abbaye.

Si plusieurs joueurs sont sur un même chemin ou une même ville, il y a deux possibilités :

- S'il y a le même nombre de pions, alors tous les joueurs auront le même nombre de points calculé comme ci-dessus,
- Si un joueur a plus de pions qu'un autre, alors ce dernier obtient tous les points.

En fin de partie:

La partie se termine lorsqu'il n'y a plus aucune tuile dans la pioche. A ce moment-là, chaque joueur compte le nombre de points qu'il obtient avec tous les pions qu'il a posés. Chaque chemin est toujours un point par tuile. Chaque ville non terminée est un point par tuile (sauf s'il y a une écusson, cela donne deux points supplémentaires). Le joueur ayant obtenu le plus de points gagne la partie.

Plateau du jeu:



Les différentes tuiles:



Une extension de jeu inclut 12 tuiles “rivières” dont une tuile “lac” et une tuile “source”. Elle permet d’embellir le plateau et de commencer la partie différemment. Lorsqu’elles sont incorporées au jeu, la partie commence avec la tuile “source”. Une pioche constituée seulement de tuiles “rivière” est créée avec la tuile “lac” placée en dernier. Chaque joueur pioche à tour de rôle dans cette pile pour constituer la rivière. Les règles ci-dessus s’appliquent dans cette version du jeu, c’est-à-dire si une tuile “rivière” possède un chemin, le joueur pourra poser un pion dessus. Aucun pion ne peut être posé sur la rivière en elle-même. Lorsque la dernière tuile “lac” est placée, le jeu commence normalement.

Eléments clés du projet

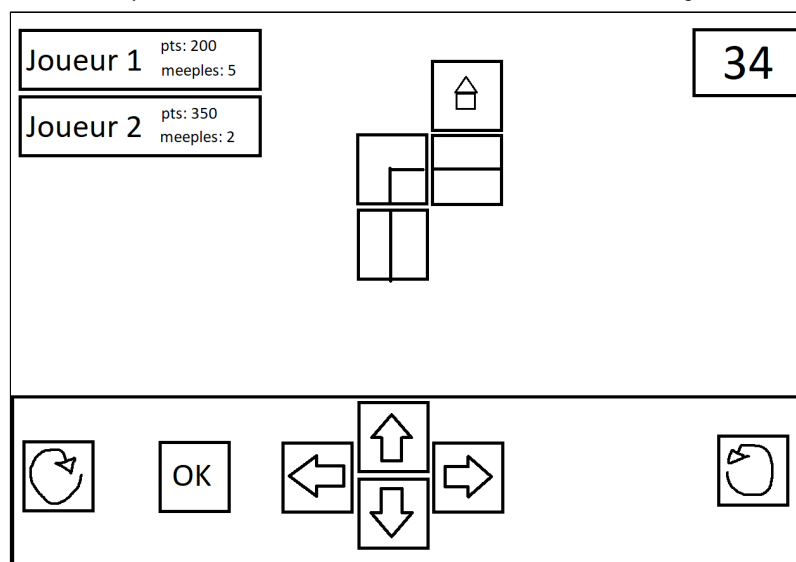
Front:

Le front s'occupe en majorité de l'esthétique du jeu et de son ergonomie. D'abord faire en sorte que les différents menus soient chacun assez clair et intuitif pour que n'importe quel utilisateur comprenne instinctivement notamment en ajoutant des options nécessaires pour adapter le confort de jeu de l'utilisateur.

De plus, le front s'occupe de créer les différents éléments du jeu comme les tuiles et les pions, car pour pouvoir utiliser Unity il faut dans un premier temps créer les modèles 3D.

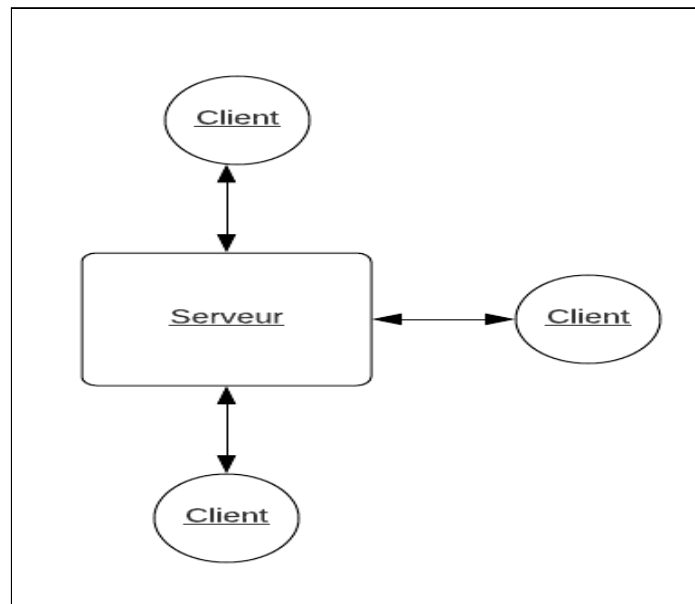
Si le temps le permet, nous pourrions ajouter des animations et améliorer la charte graphique, qui n'est pas obligatoire pour le bon fonctionnement de jeu mais qui ajoute un confort visuel ainsi qu'un plaisir de jeu pour l'utilisateur.

Maquette de l'interface utilisateur dans le jeu :



Réseau:

Le réseau est la base de notre jeu. Comme dit dans la description du projet, l'idée de départ est de créer un jeu multi-joueurs. Le but du réseau est très simple: pouvoir mettre en place un système permettant au joueur d'interagir avec d'autres. Pour que le jeu puisse fonctionner en multijoueur, nous avons à notre disposition une machine qui est fournie par la faculté sur laquelle on implémentera un serveur web. Elle sera configurée de manière à ce que les joueurs puissent se connecter dessus, et qu'ils puissent lancer, rejoindre et jouer leurs parties. Il faudrait donc que ces communications soient sécurisées et fiables. Comme le jeu est un tour par tour, il n'y aura pas trop d'informations à gérer simultanément. L'architecture ressemblera à ceci:



La décision sera maintenant de savoir quelle infrastructure de réseau utiliser. Ces choix seront détaillés dans la section suivante.

Back:

Le back a pour objectif d'imaginer une architecture de classes et de sous-classes pour chaque objet qui sera utilisé lors d'une partie (pioche, tuiles, pions, chronomètres, points, mode...). Cette architecture devra respecter à la lettre les mécanismes du jeu (placement des tuiles, placement des pions, comptage des points). Notre travail sera aussi de faire la portabilité entre l'application logicielle et l'application web ou téléphone.

BDD:

La base de données nous servira essentiellement pour collecter les informations des joueurs (nom d'utilisateur, mot de passe, score, classement, liste d'amis). Si plusieurs joueurs jouent plusieurs parties ensemble, cela permettra de définir des statistiques entre ces joueurs.

L'accès à la base de données se fera uniquement en début et en fin de partie. En début pour vérifier si le joueur existe déjà dans la base de données et en fin de partie pour enregistrer ses résultats ainsi que son classement qui seront mis à jour à chaque partie.

Choix des technologies utilisées

Pour l'expérience utilisateur ainsi que la jouabilité (GamePlay), notre outil principal pour développer sera Unity afin de manipuler les modèles 3D. Unity a été choisi car il est l'un des moteurs qui permet de faire des jeux 3D. C'est pour cela que les moteurs GDevelop ou Allegro n'ont pas été retenus. D'autres moteurs de jeux comme RPG Maker ou Unreal étaient envisageables mais ne correspondaient pas à la création d'un jeu de plateau; nous les avons donc aussi écartés. À la fin, il nous restait le choix entre deux célèbres moteurs de jeux Unity et Godot, nous avons choisi Unity car Godot est un moteur prometteur mais trop jeune. On y trouve beaucoup moins de cours et de tutoriels, à l'inverse d'Unity qui possède une grande communauté.

Le langage de programmation retenu est le C# (C Sharp) car c'est le langage natif d'Unity. Le JavaScript était une seconde possibilité (surtout employé pour du langage web) mais nous préférons prendre le langage qui est le plus adapté pour ce type de projet et qui permet l'intégration des différentes phases de développement entre les différents groupes sans trop de difficulté.

Unity rend possible la création d'interfaces utilisateurs réactives qui permettent de varier entre différentes scènes et ne sont pas problématiques pour la portabilité du programme. De plus, l'interface sera "responsive" c'est-à-dire qu'elle s'adaptera en fonction de l'orientation du support utilisé ainsi que ses dimensions.

Pour le réseau, Unity sera aussi utilisé ce qui nous permettra d'écrire des scripts en C# comme dit précédemment. En effet, l'API (application programming interface) d'Unity contient beaucoup de solutions pour le netcode du jeu, dont MLAPI, une API open source spécialement conçue pour les jeux multijoueurs en-ligne. Cette API nous permettra d'opérer sur des parties hébergées par le client, d'envoyer et de recevoir des données entre les clients et le serveur et gérer tout le reste. Nous utiliserons aussi Apache 2 pour le serveur web. Dans le cas où l'on devrait communiquer avec la base de données, on utilisera le langage MySQL.

Vision du rôle de chef de projet

Le rôle principal du chef de projet est d'une part, de fédérer, d'organiser le travail de chacun et de motiver les membres pour que la globalité du travail soit faite dans les temps et d'autre part, que le produit fini réponde parfaitement au cahier des charges.

Premièrement, il est là pour donner une ligne directrice à suivre tout au long du projet pour que chaque sous-groupes et membres soient coordonnés et que chacun ait un travail clairement défini à produire tout au long de la création du projet sans qu'il y ait des personnes avec une surcharge de travail et d'autres avec trop peu de tâches à faire. De plus, le chef de projet se doit d'accompagner les membres de l'équipe et leur travail. Il doit suivre régulièrement le travail de chacun, apportant ainsi une vision d'ensemble pour pouvoir prévoir les éventuels retards ou dysfonctionnements dûs à un travail asynchrone ou à des mauvais choix quelconques.

Secondement, il doit apporter une cohésion et une bonne entente dans l'équipe pour que chacun se sente à sa place, que chacun soit à son juste poste et que tous donnent le meilleur d'eux-même. Il a aussi un rôle de médiateur pour tout type de problèmes et il se doit d'être présent pour chacun s'il le faut. Il se doit aussi d'être bienveillant pour qu'il y ait une bonne dynamique de groupe et son comportement doit inciter une confiance pour que chacun puisse venir le voir pour tout problème. De plus, il devra asseoir une certaine autorité notamment au regard des choix importants à faire tout au long du projet et pour le succès de ce dernier. Cela a pour but qu'il y ait une bonne et équitable répartition du travail et qu'il n'y ait pas d'inégalité et de retard lorsqu'il y a un désaccord entre certaines personnes. Finalement, le mot-clé est la motivation car son attitude doit donner envie aux membres de se surpasser même lorsqu'il y a des problèmes au sein de l'équipe. Il doit réussir à être stable émotionnellement et à encourager lorsqu'il le faut pour que le projet soit réussi.

En somme, le chef de projet détient la vision globale du travail à fournir. Il se doit d'organiser le rôle de chacun des membres au service du projet et de donner des objectifs réguliers à ses équipes. Il se doit aussi de coordonner et d'accompagner les différentes équipes tout au long du processus de création du produit.

Tâches et équipes

Domaine / Tâches	Prénom
Front / Design	Anthony, Matthieu, Marc, Martin, (Achraf)
Back / BDD	Michel, Florentin, Achraf, Anas, Guillem, Déborah, (Martin), (Anthony), (Godwin), (Marc)
Réseau	Omar, Louis, Godwin

Les personnes entre parenthèses sont vouées à être déplacées d'équipe si jamais il y a des retards dans certaines équipes pour pouvoir équilibrer les charges de travail.

Front:

Les tâches importantes du front sont la conception des différents éléments du jeu:

- Créer un menu ergonomique et clair pour l'utilisateur.
- Créer des animations pour rendre le jeu fluide et plus agréable à jouer.
- Faire l'interface entre le jeu et le joueur.
- Créer les différents éléments du jeu (pions, tuiles...)

Back:

La répartition des tâches n'est pas encore définitive mais nous savons ce que nous allons faire, c'est-à-dire le déroulement d'une partie, avec le respect des règles. Le déroulement se fait comme ci-dessous):

- Mise en place de la pioche composée de la totalité des tuiles
- Lancement d'un chronomètre au début du tour de chaque joueur
- Pioche d'une tuile
- Pose de la tuile obligatoire:
 - Rotation de la tuile si nécessaire
 - Respect des caractéristiques spécifiques à chaque tuile (chemin/ville, sur le haut/bas/côté de la tuile)
 - S'il n'y a pas la possibilité de poser la tuile, elle est remise en bas de la pioche. Le chronomètre est réinitialisé et le joueur pioche à nouveau
 - Si le joueur ne joue pas dans les temps (fin du chronomètre), la tuile est posée automatiquement et c'est au joueur suivant de jouer
- Pose d'un pion, si possible
- Comptabilité des points, récupération du pion et changement de joueur

BDD:

Nous allons suivre le modèle classique pour la création d'une base de données tels que :

- Identifier le besoin
- Définir le modèle de base de données
 - Modèle entité-association
 - Modèle relationnel
- Choisir le système de gestion de base de données (SGBD)
 - MySql
- Prévoir son infrastructure (création des tables, relations, contraintes)
- Optimiser la base de données
- Suivre et maintenir la plateforme
- Anticiper l'évolution de la plateforme.

Réseau:

La répartition des tâches est inconnue pour le moment. Puisque notre équipe est assez petite et que chaque tâche est dépendante de la précédente, nous travaillerons ensemble sur chaque fonctionnalité décrite ci-dessous:

- Configurer le serveur web (installer sql, ssh, apache...)
- Permettre à un client de se connecter grâce à son pseudo et mot de passe.
- Possibilité de créer un lobby.
- Possibilité de rejoindre un lobby.
- Créer une structure de données pour les packet
- Envoyer/recevoir des demandes d'amis.
- Communiquer avec le serveur pendant une partie pour transmettre les actions du joueurs.

Diagramme de Gantt

Le diagramme de Gantt produira deux livrables avant la version finale situés au premier et deuxième tiers de la durée totale du projet. L'objectif est d'une part, de témoigner au client de l'avancée du projet et d'autre part, d'avoir des versions intermédiaires. Le diagramme de Gantt est ci-dessous.

Tâches	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
Réseau											
Configurer le serveur web											
Permettre à un Client de se connecter grâce à son pseudo/mdp											
Pouvoir créer un lobby											
Pouvoir rejoindre un lobby											
Envoyer/recevoir des demandes d'amis											
Communiquer avec le serveur pendant une partie pour transmettre les actions du											
Front											
Création des modèles 3D											
Création de l'interface utilisateur (menus, plateau de jeux)											
Création des animations des modèles 3D											
Création des fonctionnalités de l'interface utilisateur											
Back / BDD											
Documentation et conception											
Implémentation et intégration du premier prototypage											
Correctif et polissgae puis intégration du deuxième prototypage											
Adaption pour client web ou mobile											