



DOCUMENT DE FIN D'ÉTUDES PRÉLIMINAIRES

03 OCTOBRE 2018

Jonathan CABEZAS p1513280

Alexandre HOET p1502007

Elodie MONTCARMEL

Louis POITEVIN p1410541

Gabriello ZAFIFOMENDRAHA p1311399



Intervenants : E. COQUERY, L. MEDINI

Année universitaire 2018-2019

Dernière modification : 03/10/2018

Auteurs : Jonathan CABEZAS, Alexandre HOET, Elodie MONTCARMEL,
Louis POITEVIN, Gabriello ZAFIFOMENDRAHA

Sommaire

Introduction	3
Cahier des charges	4
I. Présentation du projet	4
II. Fonctionnalités	5
Cas d'utilisation	8
Pistes d'architecture	16
Outils de gestion de projet et d'intégration continue	18
Risques identifiés	18

Introduction

Dans le cadre de l'unité d'enseignement MultiMIF du Master 1 Informatique à l'université Claude Bernard Lyon 1, nous sommes amenés à améliorer les fonctionnalités d'un jeu tout en assurant une bonne gestion de projet. Ce projet vise à se rapprocher d'une expérience de développement professionnel tout en mettant en pratique les techniques vues au cours de nos études. Le sujet du projet est le développement d'un jeu de courses de chevaux.

Nous allons tout d'abord faire une présentation du projet puis nous décrirons les fonctionnalités à implémenter. Ensuite, nous présenterons les différents cas d'utilisation de l'application, les pistes d'architectures, les outils de gestion de projet et d'intégration continue et enfin les risques identifiés.

Cahier des charges

I. Présentation du projet

- **Contexte**

Dans le cadre du projet MultiMIF nous devons améliorer le projet Poneymon commencé en Gestion de projet. Pour cela nous sommes partis du projet réalisé par Jonathan CABEZAS en Génie Logiciel qui nous servira de base et nous allons au fur et à mesure rajouter des fonctionnalités qui contribueront à l'amélioration du jeu.

- **Objectifs**

L'objectif de ce projet est de développer un jeu de courses de chevaux avec plusieurs modes de jeux et notamment les modes Solo et Multijoueurs qui est un mode en ligne.

L'utilisateur aura aussi la possibilité de configurer les paramètres du jeu.

De plus, une amélioration de l'IA existante sera présente afin de prendre en compte les parcours et les types de poneys.

- **Répartition des rôles**

Product Owner : Elodie MONTCARMEL

SCRUM Master : Jonathan CABEZAS

Tracker : Alexandre HOET

Responsable Qualité : Gabriello ZAFIFOMENDRAHA

Référents techniques :

Réseau : Alexandre HOET

JavaFX : Jonathan CABEZAS

Animation/Sprites : Louis POITEVIN

II. Fonctionnalités

- **Affichage du menu principal**

Au lancement de l'application, le menu principal apparaît. Il offre la possibilité de choisir le mode de jeu et de configurer les paramètres de l'application.

- **Mode Solo**

Dans ce mode de jeu, le joueur peut lancer une course contre des IA. L'utilisateur peut configurer la partie avant de la lancer. Il n'aura pas besoin d'une connexion internet pour jouer avec ce mode.

- **Mode Multijoueurs**

Dans ce mode de jeu, le joueur aura la possibilité de créer ou de rejoindre une partie en ligne. Il pourra alors jouer avec les joueurs ayant rejoint la même partie.

Si le joueur crée une partie, il devra la configurer puis il deviendra alors l'hôte de celle-ci et devra attendre que l'ensemble des joueurs rejoigne la partie.

Sinon le joueur rejoint une partie déjà configurée.

- **Réglage des paramètres**

L'utilisateur peut choisir de régler le son, la résolution graphique ainsi que les commandes du jeu.

- **Choix de plusieurs terrains possibles**

Avant de lancer la partie le joueur peut choisir le terrain sur lequel la course va avoir lieu. Chaque terrain possède des obstacles qui lui sont propres, et un nombre fixe de voies.

- **Choix du poney**

Le joueur peut choisir le poney avec lequel il souhaite jouer. Chaque poney a des caractéristiques propres lui permettant d'être avantage sur certains terrains.

Exemple de poneys :

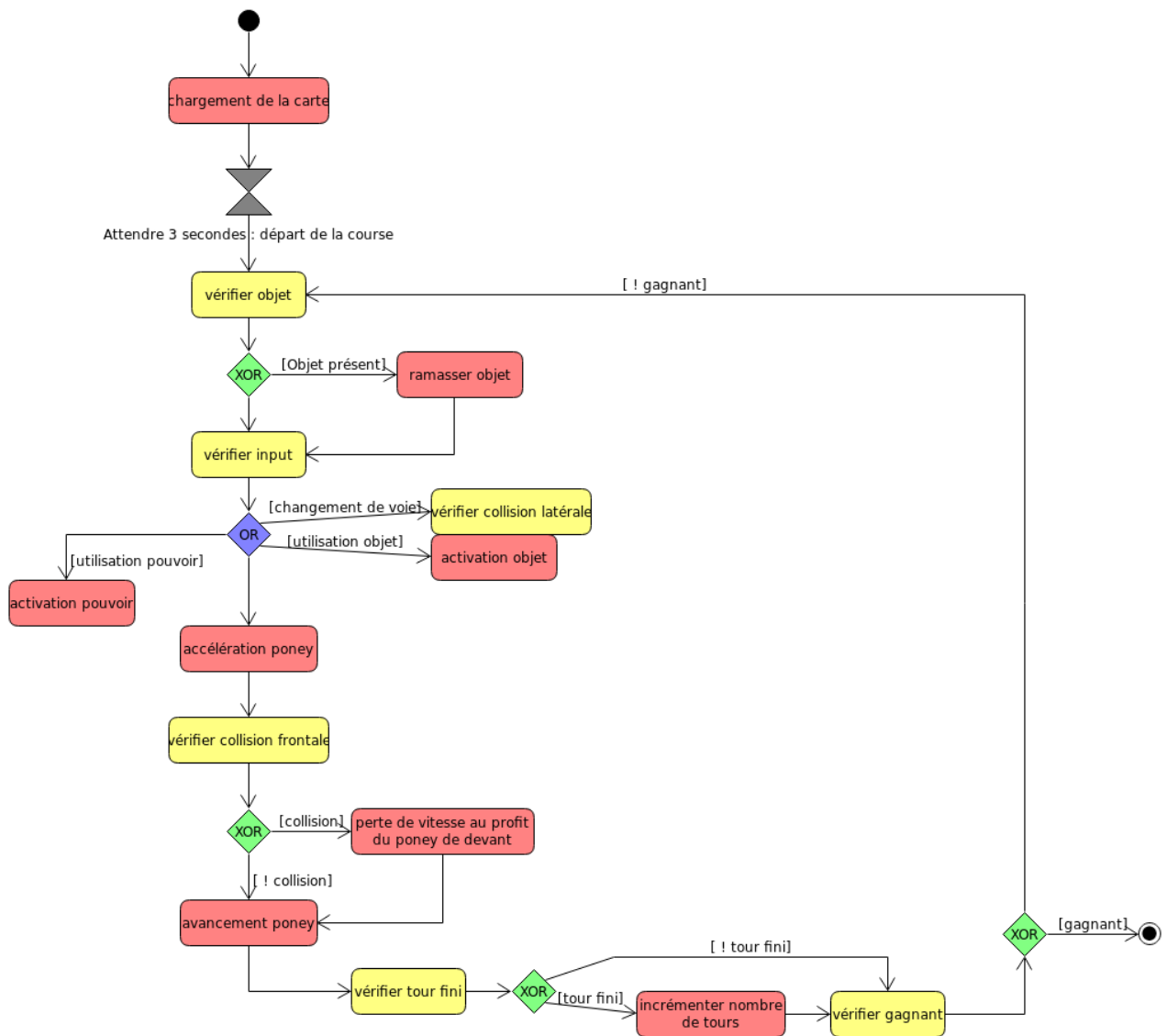
Les deux premiers poneys distincts correspondent à deux joueurs différents. Le troisième correspond à une amélioration du deuxième poney.



- **Déroulement d'une partie**

Lorsque l'utilisateur a choisi le mode de jeu auquel il souhaite jouer, il a alors la possibilité de configurer sa partie et de choisir son poney. Ensuite, la course commence. Le joueur voit la course en « vue de dessus ». Il a la possibilité de déplacer son poney d'une voie à l'autre afin d'éviter des obstacles, de récupérer des objets ou encore dans le but de gêner ses adversaires.

Le joueur n'a pas besoin de faire accélérer son poney, celui-ci accélère tout seul au fur et à mesure de la course. Quand le joueur rentre en collision avec un poney devant lui, il lui transmet de sa vitesse et s'en trouve ralenti. Il n'est pas possible de ralentir via les commandes du jeu. Plus on va vite, plus il est compliqué d'éviter les obstacles.



- **Ramasser des objets et des bonus/malus**

Les objets et les bonus/malus peuvent être ramassés pendant la course. Les objets ramassés pourront être utilisés plus tard alors que les bonus/malus ont un effet immédiat.

- **Affichage du classement**

A la fin de la partie, le jeu affiche le classement final de la course.

Cas d'utilisation

- **Affichage du menu principal**

Objectif : Le joueur accède au menu principal du jeu

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur lance l'application

Précondition : Le joueur possède le jeu

Garanties minimales : /

Garanties en cas de succès : Le joueur accède au menu

Scénario nominal :

1. Le joueur lance l'application
2. Le joueur voit s'afficher le menu principal comprenant le choix du mode de jeu (partie en mode solo ou en mode Multijoueurs) ainsi que l'accès aux paramètres et un bouton quitter

- **Lancer une partie en mode solo**

Objectif : Le joueur lance une partie en mode solo

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur lance un jeu

Précondition : /

Garanties minimales : Le joueur accède à une partie en solo

Garanties en cas de succès : Le joueur accède à une partie en solo

Scénario nominal :

1. Le joueur lance l'application et atterrit sur le menu principal
2. Il sélectionne le mode solo

3. Le joueur configure sa partie (nombre de tours, nombre d'IA, etc...)
- 4.. Le joueur lance la partie

- **Créer une partie en mode Multijoueurs**

Utilisateur :

Objectif : Le joueur crée un lobby Multijoueurs

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur lance le jeu

Précondition : Le joueur a accès à Internet

Garanties minimales : Le joueur crée un lobby, le configure et attend d'autres joueurs

Garanties en cas de succès : Le joueur accède à une partie avec plusieurs joueurs humains

Scénario nominal :

1. Dans le menu principal, le joueur clique sur partie Multijoueurs
- 2.. Le joueur a le choix entre rejoindre un lobby et créer un lobby, il choisit d'en créer un
3. L'hôte configure les différentes options de la partie : nombre de joueurs, joueurs humains et IA, objets et bonus disponibles, tickrate serveur, etc et choisit son poney
- 4.. L'hôte appuie sur "Lancer la partie" quand tous les joueurs sont prêts

Système :

Objectif : Le joueur crée un lobby Multijoueurs

Portée : Poneymon

Niveau : Système

Acteur principal : Le joueur

Acteur secondaire :

Déclencheur : Le joueur lance le jeu

Précondition : Le joueur a accès à Internet

Garanties minimales : Le joueur crée un lobby, le configure et attend d'autres joueurs

Garanties en cas de succès : Le joueur accède à une partie avec plusieurs joueurs humains

Scénario nominal :

1. Dans le menu principal, le joueur clique sur partie Multijoueurs
2. Le joueur a le choix entre rejoindre un lobby et créer un lobby, il choisit d'en créer un.
3. Le système initialise un lobby et considère le joueur comme hôte
4. Le système affiche l'adresse IP au joueur hôte
5. Le joueur hôte envoie l'adresse IP aux autres joueurs pour qu'ils puissent le rejoindre

- **Rejoindre une partie en mode Multijoueurs**

Utilisateur :

Objectif : Le joueur rejoint un lobby Multijoueurs

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur lance le jeu

Précondition : Le joueur a accès à Internet

Garanties minimales : Le joueur accède à un lobby et attend d'autres joueurs

Garanties en cas de succès : Le joueur accède à une partie avec plusieurs joueurs humains

Scénario nominal :

1. Dans le menu principal, le joueur clique sur partie Multijoueurs
2. Le joueur a le choix entre rejoindre un lobby et créer un lobby, il choisit d'en rejoindre un
3. Un écran l'invite à rentrer l'adresse IP de l'hôte hébergeant le lobby
4. Le joueur choisit son poney et clique sur "Prêt" quand il est prêt
5. Le joueur attend que l'hôte lance la partie

Extensions :

3a. Le joueur ne connaît pas l'adresse IP, il se renseigne ou abandonne l'action

Système :

Objectif : Le joueur rejoint un lobby Multijoueurs

Portée : Poneymon

Niveau : Système

Acteur principal : Le joueur

Déclencheur : Le joueur lance le jeu

Précondition : Le joueur a accès à Internet

Garanties minimales : Le joueur accède à un lobby et attend d'autres joueurs

Garanties en cas de succès : Le joueur accède à une partie avec plusieurs joueurs humains

Scénario nominal :

1. Dans le menu principal, le joueur clique sur partie Multijoueurs
2. Le joueur a le choix entre rejoindre un lobby et créer un lobby, il choisit d'en rejoindre un
3. Un écran l'invite à rentrer l'adresse IP de l'hôte hébergeant le lobby
4. Le joueur demande à l'hôte son adresse IP
5. Le joueur rentre l'adresse IP reçue
6. Le système ouvre une connexion vers l'adresse IP
7. Le système récupère les informations du lobby auprès de l'hôte

Extensions :

6a. La connexion échoue, il faut trouver la cause de l'échec

• Le joueur (client) se déconnecte

Utilisateur :

Objectif : Le joueur se déconnecte

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur perd la connexion

Précondition : Une partie est en cours

Garanties minimales : Le jeu continu sans le joueur déconnecté

Garanties en cas de succès : Le jeu se poursuit avec le joueur déconnecté

Scénario nominal :

1. Le joueur choisit une partie Multijoueurs dans le menu principal
2. Il rejoint un lobby
3. Il attend que l'ensemble des joueurs soient prêt
4. La partie se lance
5. Le joueur se déconnecte
6. Le joueur peut tenter de revenir dans la partie

Extensions :

- 6.a : Le joueur quitte définitivement la partie

Système :

- Objectif : Déconnexion d'un joueur client pendant une partie
- Portée : Poneymon
- Niveau : Système
- Acteur principal : Poneymon
- Acteur secondaire : Le joueur
- Déclencheur : Le joueur perd la connexion pendant une partie
- Précondition : Une partie est en cours
- Garanties minimales : Le personnage du joueur déconnecté "joue" encore
- Garanties en cas de succès : Le joueur se reconnecte et le jeu continue normalement
- Scénario nominal :
 - 1. Pendant une partie le joueur perd sa connexion
 - 2. Le système du joueur et de l'hôte constate la perte de connexion via un timeout (délai d'attente de réponse de l'interlocuteur dépassé)

- 3. L'Hôte et le client conservent les données de la partie et continuent de la faire tourner comme si le joueur était encore connecté
- 4. Le client parvient à se reconnecter à l'hôte, le dialogue entre les deux interlocuteurs reprend
- Extensions :
 - 4a. Le client ne parvient pas à se connecter en moins de x Secondes, il est renvoyé à l'écran principal et est remplacé par une IA dans la partie
- **Le joueur (hôte) se déconnecte pendant la partie**

Utilisateur :

Objectif : Déconnexion d'un joueur hôte pendant une partie

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur perd la connexion pendant une partie

Précondition : Le joueur a commencé une partie Multijoueurs

Garanties minimales : La partie s'arrête

Garanties en cas de succès : Le joueur hôte se déconnecte mais le jeu continu

Scénario nominal :

- 1.. Pendant une partie le joueur hôte est déconnecté
- 2.. Un texte est affiché pour prévenir de la migration de l'hôte et la partie est mise en pause
- 3.. Le texte de migration d'hôte disparaît et le joueur reprend le contrôle de son personnage

Extensions :

- 3a. Le joueur est renvoyé dans le menu principal avec un texte qui dit que la connexion a été perdue complètement

- **Le joueur modifie les paramètres de l'application**

Objectif : Le joueur modifie les paramètres de l'application

Portée : Poneymon

Niveau : Utilisateur

Acteur principal : Le joueur

Déclencheur : Le joueur lance l'application

Précondition : /

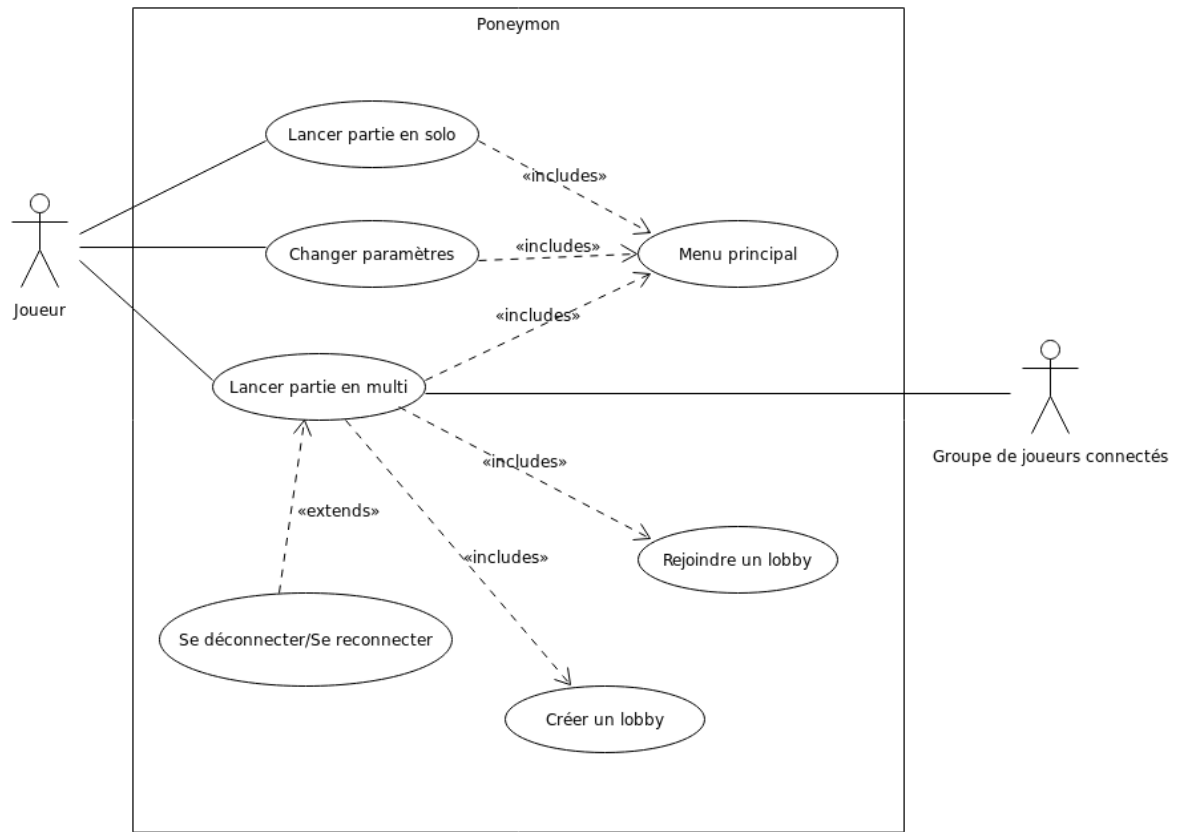
Garanties minimales : /

Garanties en cas de succès : Les changements effectués par l'utilisateur sont sauvegardés

Scénario nominal :

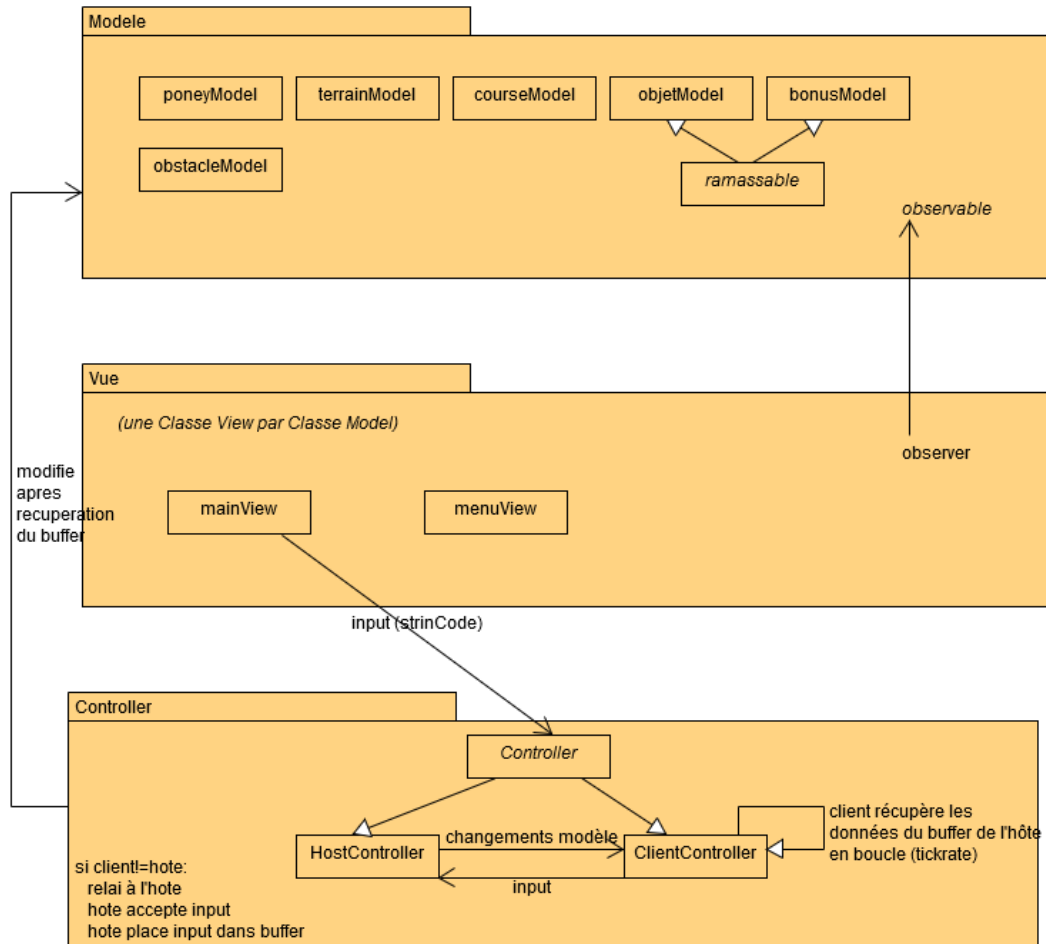
1. Le joueur lance l'application
2. Le joueur choisi "Modification de paramètres" dans le menu principal
3. Le joueur peut configurer les paramètres graphiques, de sons, et de commandes

- Diagramme de cas d'utilisation



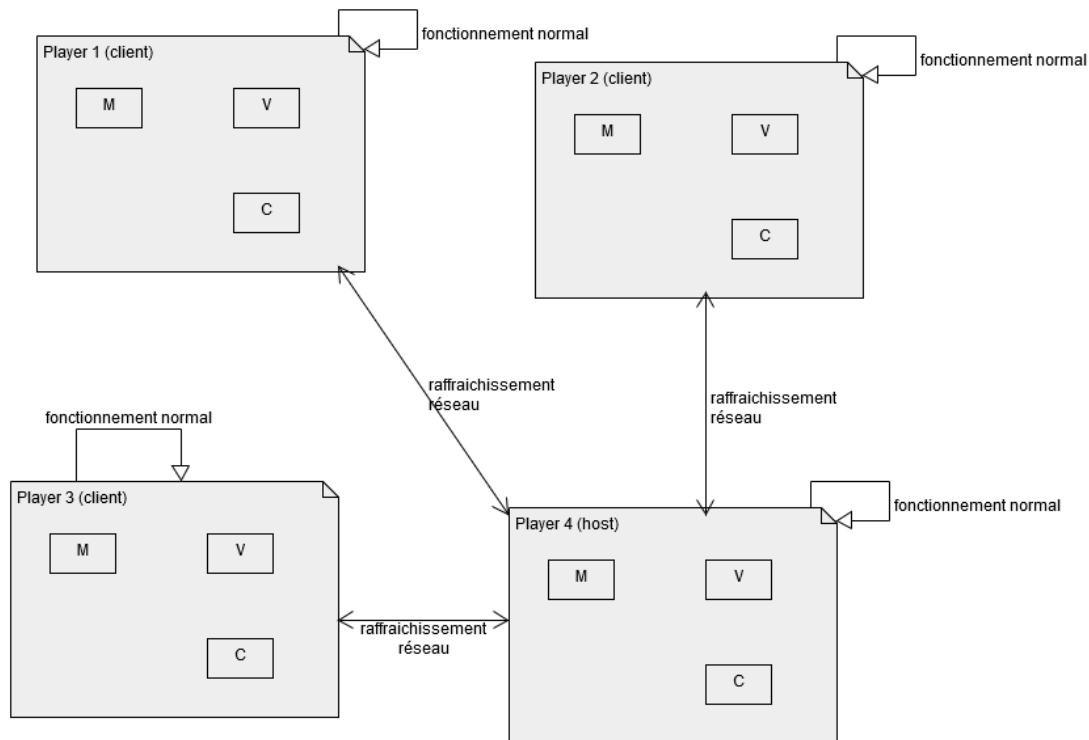
Pistes d'architecture

- **Modèle-Vue-Contrôleur**



Nous souhaitons structurer notre code en implémentant le MVC. Chaque objet aura un modèle et une vue associés. La vue récupèrera les données du modèle puis en fonction des actions de l'utilisateur, le contrôleur mettra à jour les modèles.

- Réseau



Pour jouer en mode Multijoueurs, il faut établir un système reliant les différents joueurs. Le joueur qui crée la partie devient l'hôte de celle-ci et reçoit les adresses IP de tous les autres joueurs de la partie. Les autres joueurs sont donc clients et reçoivent simplement l'adresse IP du joueur hôte.

Outils de gestion de projet et d'intégration continue

- **Maven**

Maven est un outil de gestion et d'automatisation de production des projets logiciels que nous avons eu l'occasion d'utiliser dans l'UE gestion de projet et génie logiciel. C'est pour cela que nous avons décidés de le réutiliser maintenant que nous connaissons son utilité et que nous le maîtrisons.

- **Sonarqube**

Sonarqube est un outil d'audit de code. Il permet ainsi de mesurer la qualité du code produit. Il fournit des rapports sur plusieurs mesures comme la duplication de code, le niveau de documentation, la détection de bugs potentiels, l'évaluation des tests unitaires sur le code mais également une analyse sur le design et l'architecture d'une application. Les intervenants de l'UE nous ont conseillé de l'utiliser lors de la présentation des projets.

Risques identifiés

- Le manque de temps pour implémenter toutes les fonctionnalités
- L'instabilité du programme suite à l'implémentation de plusieurs fonctionnalités majeures
- L'instabilité du modèle suite à l'implémentation du mode Multijoueurs
- Les compétences manquantes au sein du groupe (par exemple sur le graphisme)
- Une mauvaise estimation du temps nécessaire pour une tâche SCRUM
- Des problèmes de compatibilité entre plateformes