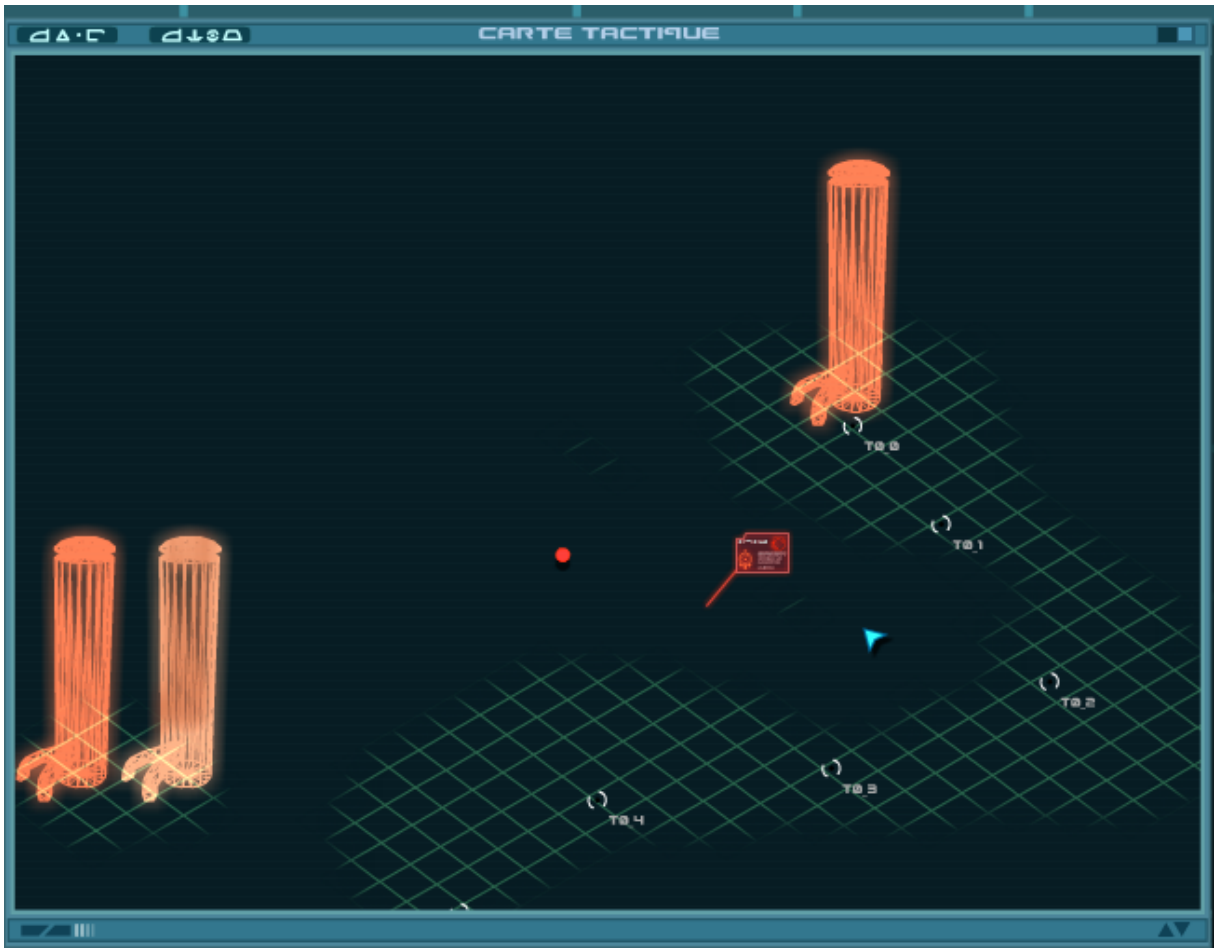
Université de technologie Belfort-Montbéliard

Éduram (Adaptation du jeu Pandémie)

LO43 Bases Fondamentales de la programmation orientée objet

Auteur :
GLANGINE Geoffrey
SENE Victor

Professeur :
Gechter Franck



18 décembre 2015

Table des matières

Première partie

Qu'est-ce que Eduram ?

1 Un jeu au règles complexes

Eduram est un jeu basé sur le célèbre jeu Pandémie (Pandémic en version originale). Le principe du jeu est d'essayer d'empêcher des virus de se propager dans les villes du monde entier. C'est un jeu de coopération ou les joueurs jouent tous dans la même équipe contre le jeu. Chaque joueur possède des cartes avec des noms de ville et des couleurs car la carte est séparée en plusieurs continents et chaque continent possède une couleur. Les joueurs peuvent effectuer 4 actions par tour : se déplacer, créer un vaccin contre 5 cartes de la même couleur, enlever un virus sur la ville ou ils se trouvent actuellement, donner une carte à un ami qui se trouve sur la même ville qu'eux ou encore effectuer une action liée à leur spécialité. Chaque joueur possède un rôle qui lui procure une ou plusieurs des actions spéciales (exemple : le scientifique peut créer un vaccin pour seulement 4 cartes). Pour poursuivre son tour le joueur pioche des cartes joueurs qui vont lui permettre de créer les antidotes au virus, de se déplacer ou bien d'utiliser une capacité unique conférée par les cartes spéciales. A la fin du tour du joueur, un ensemble de carte sont piochées et engendre le développement d'un virus sur la ville donner par la carte. Cependant lorsque 3 virus sont présents sur cette ville et qu'un quatrième fait son apparition il y a alors une éclosions : chaque ville aux alentours est contaminé par le virus. Le jeu est perdu s'il y a eu plus de 8 éclosions, ou bien s'il n'y a plus de cartes dans la pioche des joueurs, ou encore si on ne peux plus placer de cubes de virus sur la carte. A contrario, la partie est gagné lorsque tous les vaccins ont été créés.

2 Nos choix d'adaptation du jeu

Nous avons décidé de refaire une version utbohémiene du jeu. Vous incarnez un des rôles nécessaires au bon fonctionnement du réseau de l'UTBM. Votre but est d'éradiquer les virus présent sur les équipements de réseau de l'UTBM. Vous avez à votre disposition les mots de passe nécessaire pour accéder aux équipements des salles. Votre rôles vous procure une compétence spécifiques à utiliser selon votre gré. En effet, nous avons choisi de représenter sur notre carte les bâtiments principaux de l'UTBM avec des salles. Un joueur situé sur une salle peut se déplacer dans les salles adjacentes pour un coût de 1'action.
Nous avons aussi décidé de remplacer les cartes de joueur par des mots de passes pour coller un peu mieux avec le secteur de l'informatique. Donc les joueurs pourront s'échanger des mots de passe comme les cartes dans le jeu original. Les joueurs pourront aussi créer des anti-virus pour supprimer des virus. La partie sera donc gagnée lorsque tous les anti-virus seront créés. Les conditions de défaites sont les mêmes que dans le pandémie original.

2.1 Implantation des rôles de joueurs

Nous avons décidé dans un premier temps de faire seulement 3 rôles :
— L'administrateur système : il peut se déplacer de bâtiments en bâtiments plutôt que de salles en salles.
— Le technicien : il peut enlever tous les virus de la salle ou il se trouve pour le coût d'une seule action.
— L'enseignant-chercheur : Peut créer un anti-virus avec seulement 4 mots de passes.

2.2 Modes de déplacement

Le jeu pandémie présentait plusieurs modes de déplacement (voiturier, vol nolisé et vol direct) Nous avons décidé de remplacer les moyens de transport par des types de communication réseau.

- SSH : représente le vol nolisé (on défausse le mot de passe de la salle où l'on se trouve pour se rendre dans n'importe quelle autre salle)
- Telnet : représente le vol direct (on défausse un mot de passe pour se rendre dans la salle associée)
- Console : représente le déplacement voiturier (d'une salle vers une salle adjacente à celle ci)

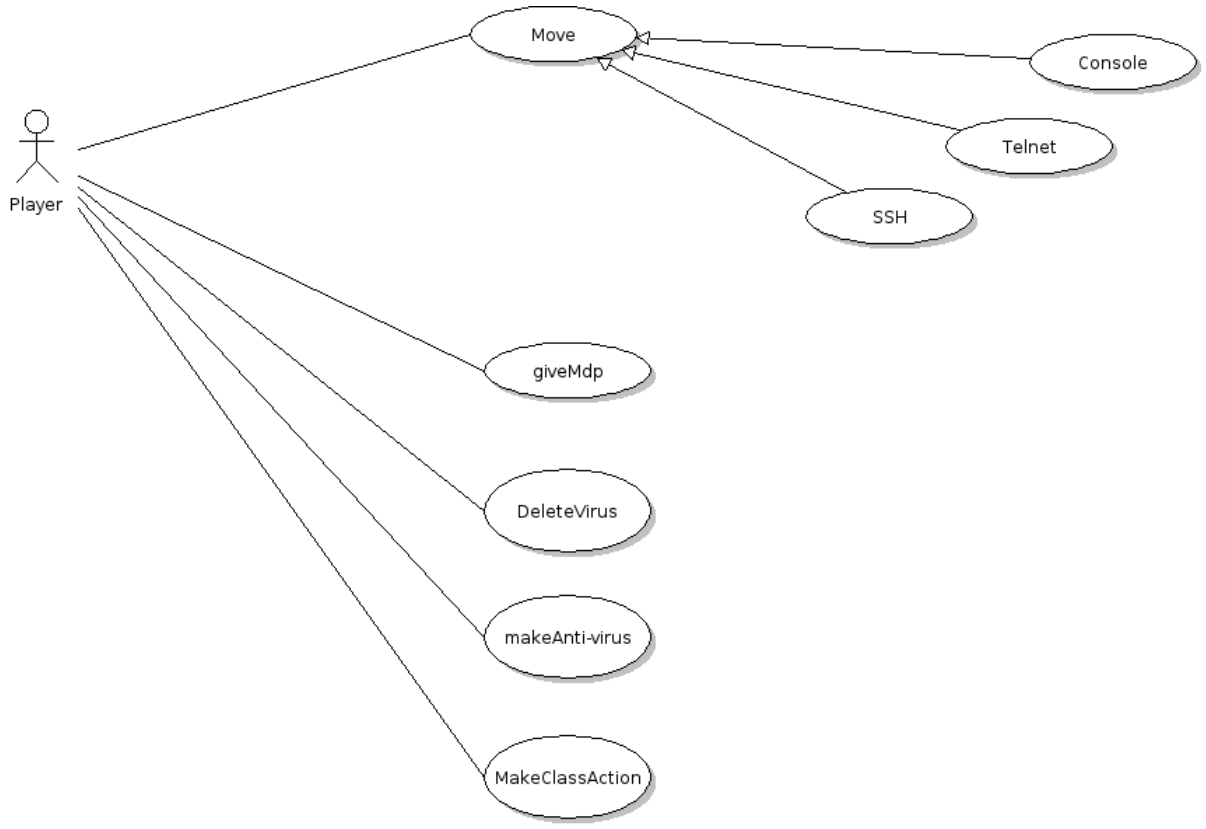
Deuxième partie
Choix de conceptions

3 Le code

Pour la programmation du projet, nous avons décidé d'utiliser le gestionnaire de version Git et d'utiliser un dépôt privé sur Github pour l'hébergement. Nous développons le projet en java et nous utilisons JavaFX pour l'interface graphique. Nous avons choisi javaFX car nous pensons que cette librairie graphique récente va petit à petit remplacer swing pour les GUI (Graphical User Interface)en JAVA. Pour des raisons de maintenance et par habitude, nous avons aussi décidé de mettre en place une architecture MVC (Modèle Vue Contrôleur) qui peu nous permettre de faire des changements de librairie graphique plutôt facilement (voir la section Amélioration). Le projet est développé en anglais de manière à ce que le code puisse être compris par une communauté qui n'est pas forcément francophone.

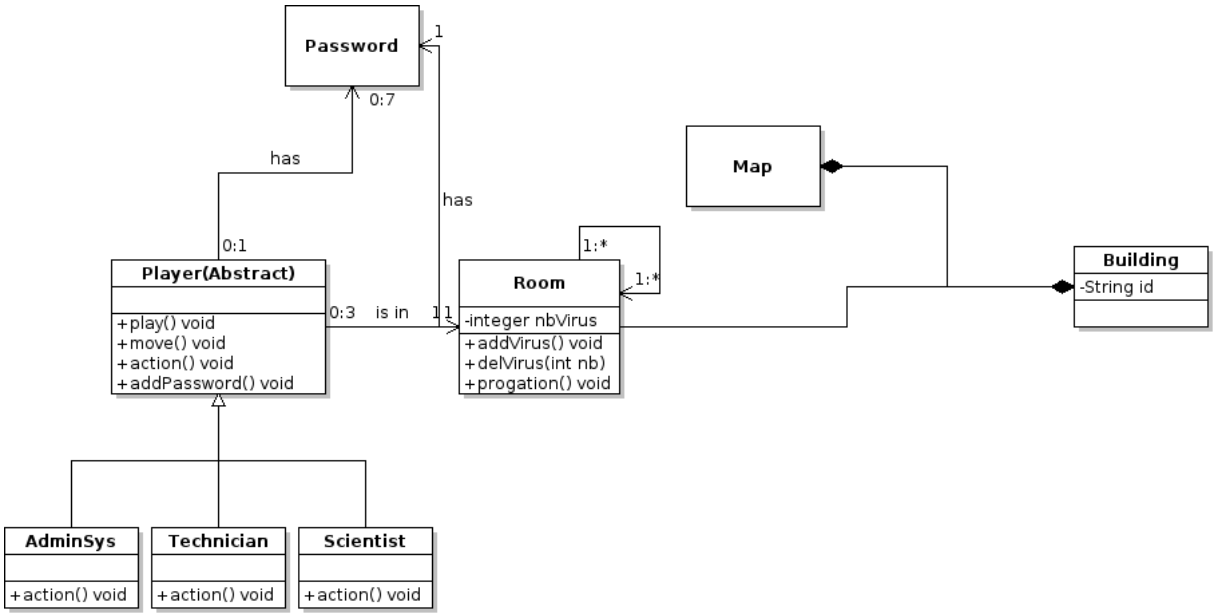
4 Analyse et conception

4.1 Digramme de cas d'utilisation



On peut voir dans ce diagramme que le joueur est le seul acteur dans ce jeu et qu'il à le choix de faire plusieurs actions telle que se déplacer, donner un mot de passe, supprimer des virus, créer un anti-virus ou encore effectuer son action de classe (Rôle).

4.2 Diagramme de classes



Le diagramme de classe précédent représente seulement les classes importantes au niveau algorithmique du jeux. On peut ainsi voir les classes de Joueurs , la carte ou plateau (map) qui est composée de Bâtiments eux mêmes composés de salles. On peut aussi voir la classe Password qui est une représentation des mot de passe d'accès au salles (la pile de carte joueur dans le jeu original).

4.3 Diagrammes de séquences

Nous avons décidé de réaliser plusieurs diagrammes de séquence pour représenter certaines phases importantes du jeu.

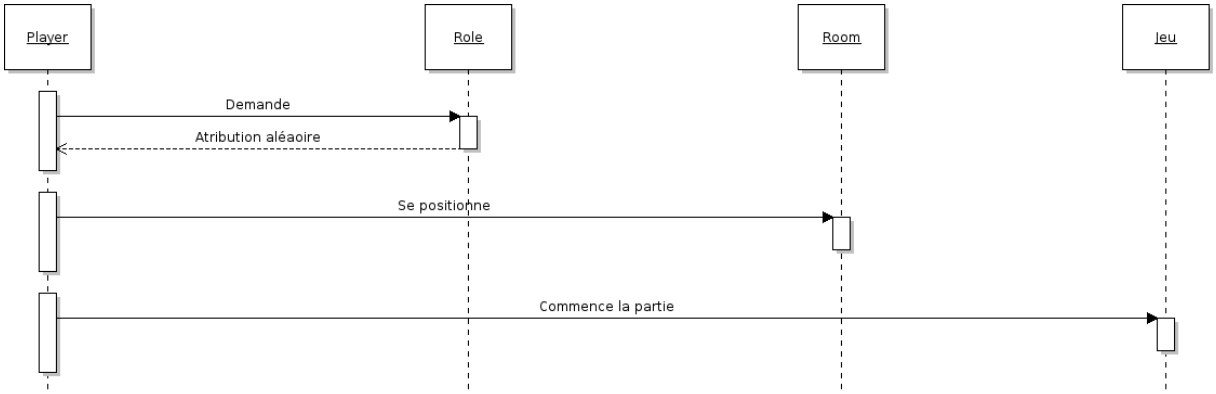


FIGURE 1 – Ce diagramme représente le début d'une partie, ou le joueur reçoit un rôle, est positionné dans la salle de départ et commence la partie.

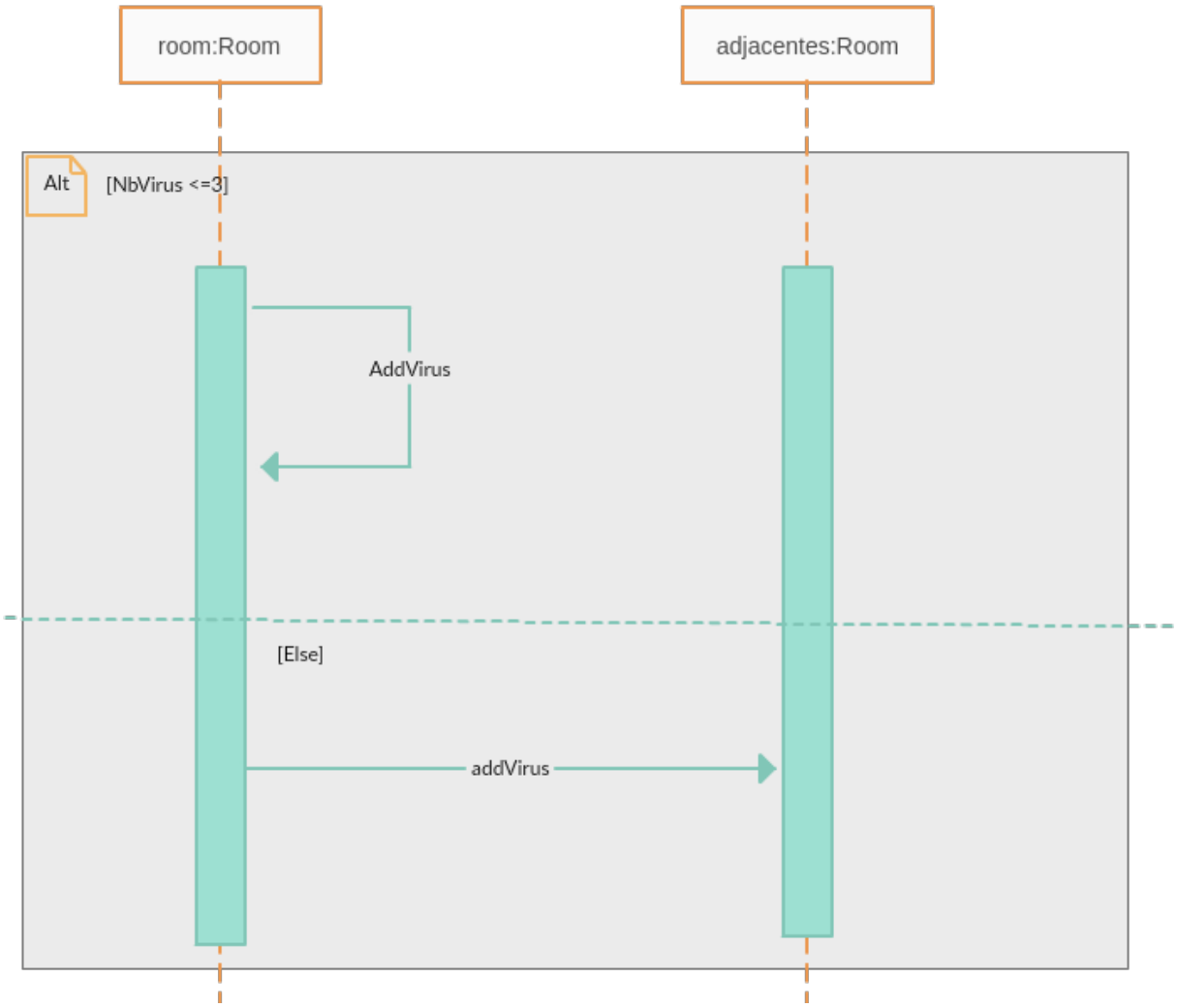


FIGURE 2 – Le diagramme ci-dessus représente une éclosion de virus. En effet dans le jeu si il y a déjà 3 virus dans une salle, on ne peut pas rajouter de virus dans cette salle. Par contre un virus est ajouté à toute les salles adjacentes à cette salle.

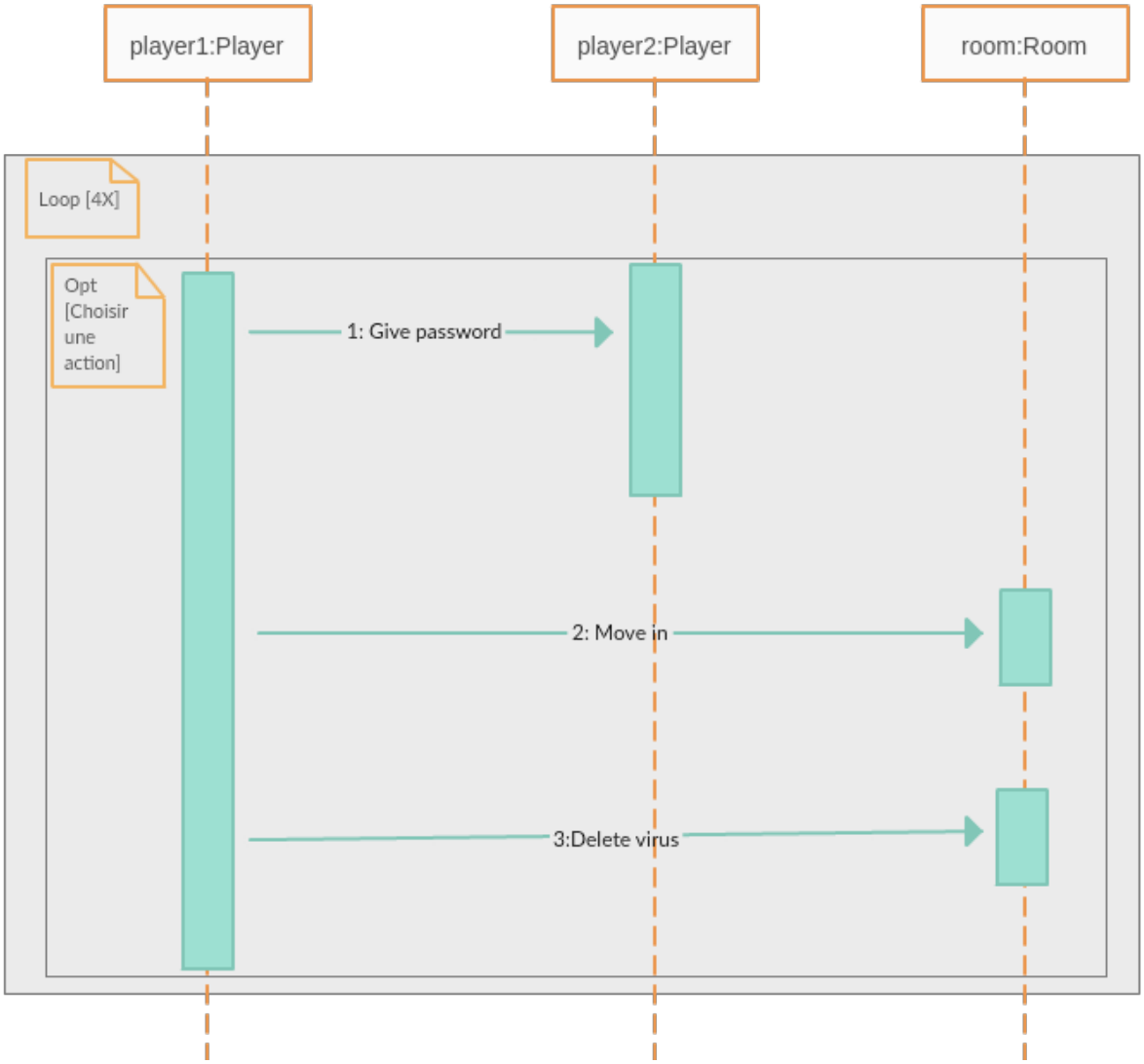


FIGURE 3 – Le diagramme ci-dessus représente le tour d'un joueur. Un joueur peut faire 4 actions par tour au choix parmi le déplacement, le don de mot de passe ou encore la suppression d'un virus.

Troisième partie

Améliorations possibles

En développant ce projet nous avons pensé à plusieurs idées d'améliorations. En effet on pensait pouvoir mettre plusieurs types de virus avec des comportements différents, mais ça posait plusieurs problèmes au niveau équilibrage du jeu Et nous aurions passé plus de temps à réfléchir à la façon d'équilibrer les virus plutôt qu'à les implémenter.

Il peut aussi être possible de rajouter un nouveau rôle de joueur qui serait "Hacker" en effet ce nouveau joueur ne jouerait pas en coopération avec les autres joueurs mais bel et bien contre eux et avec "le jeu" Cette amélioration aurait été très amusante à mettre en place, mais elle aurait complètement bouleversé le gameplay et nous aurions du revoir une grande partie de l'implémention.

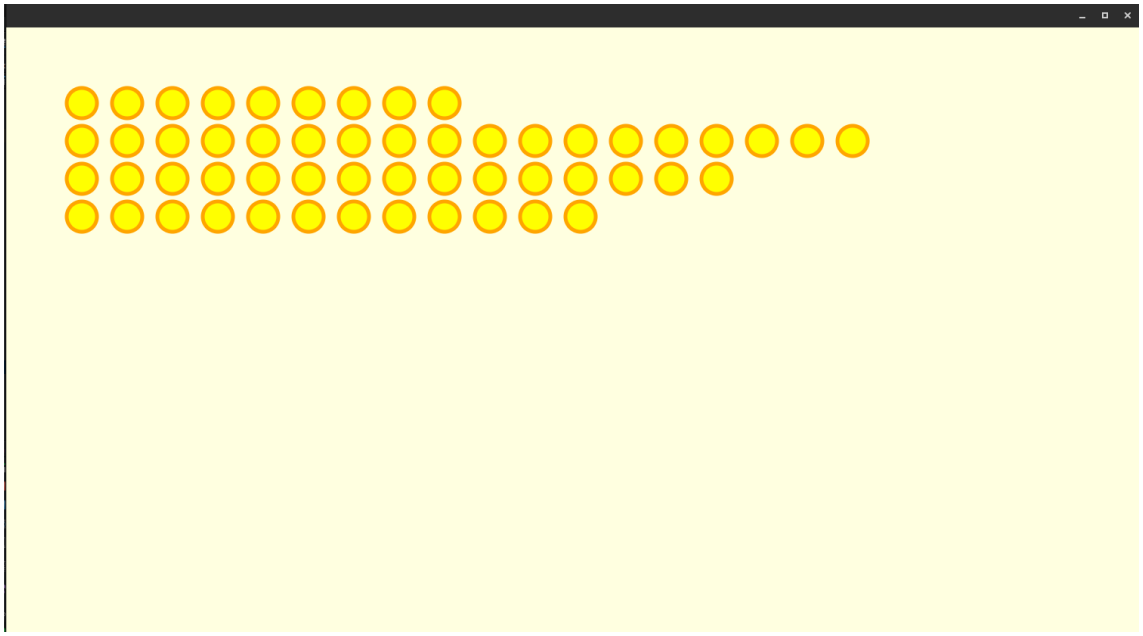
La mise en place du jeu en réseau aurait pu être faisable. En plus grâce à la mise en place du modèle MVC que l'on à fait il n'aurait pas été très compliqué de faire la mise en réseau du jeu. Mais pour faire un jeu en ligne cela nécessitait la création d'un serveur et d'un protocole pour les échanges de données. Le client aurait été complètement modifié dans la mesure ou il aurait simplement envoyé des requêtes TCP au serveur et interprété les résultats reçus pour simplement les afficher. Toutes les vérifications doivent se faire coté serveur et le client n'est en fait qu'un afficheur (Si on commence à contrôler des données sensibles dans le client, les joueurs peuvent trop facilement tricher en modifiant le code du client.). Cette partie reste, pour nous encore un espoir et sera peut être implémentée car elle nous semble la plus intéressante au niveau programmation. Mais elle nécessite toute fois un gros refactoring de code.

Et enfin nous avions aussi pensé à porter notre jeu pour android. Grâce au modèle MVC, il peut être très simple de créer l'application android. Il nous suffirait seulement de remplacer JavaFX.

Quatrième partie

L'implémentation

Une fois l'étape de conception du projet finalisé, nous avons pu commencer l'implémentation collaborative grâce à Git. Notre objectif de rendre un projet fonctionnel et propre au niveau du code. Dans l'état actuel des choses nous avons une interface graphique affichant un groupe de building contenant des "rooms".



Il nous reste donc à gérer les interactions avec le joueur et rendre l'interface plus jolie. Au final ce projet nous aura permit de découvrir une nouvelle librairie pour interface graphique et d'améliorer notre connaissance de la programmation orientée objet. Il va de soit que le projet va encore beaucoup évoluer d'ici le rendu et nous tiendrons à jour la documentation pour permettre une meilleur compréhension de notre démarche.