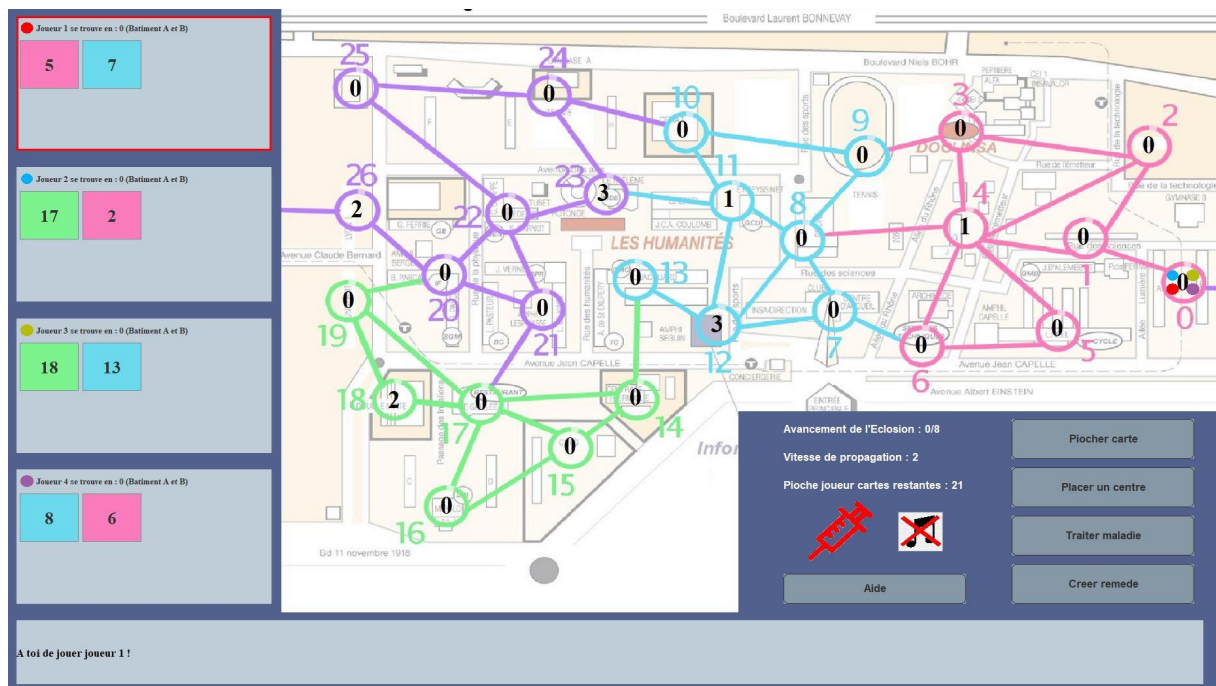


GIORDANA Maxime
GOUASMI Sarah
JOSEPH Mathéo
LAHOUSSE Coline

Groupes 53/56
Enseignantes référentes : CAZALENS Sylvie
MERVEILLE Odysée

Rapport d'informatique

Projet COV'INSA



Al Le cahier des charges de votre programme

But: Nous nous sommes fixé comme objectif de créer une version numérique du jeu de coopération de plateau pandemic. Etant donné la complexité du jeu, nous avons prévu de simplifier les règles du jeu, et d'adapter le plateau de jeu au contexte actuel et à notre campus.



Une maladie se propage sur le plateau de jeu représentant le campus de la doua et les 4 joueurs doivent lutter ensemble contre l'ordinateur pour éviter la pandémie.

Nous avons aussi pour ambition de réaliser un jeu équilibré. C'est-à-dire qu'il ne doit être ni trop difficile, ni trop facile de gagner.

Pour plus d'informations sur les règles précises, voir la *fenetreAide* qui s'affiche à l'appui du bouton aide présent lors du lancement du jeu. Nous avons également souhaité ajouter une ambiance sonore à notre jeu avec une musique de fond ainsi que des effets sonores.

B/ La description du problème posé

Nous avons envisagé différents problèmes relatifs au codage de notre projet. L'un des premiers qu'il a fallu résoudre a été le choix de la méthode de définition des lieux et des informations les concernant. Pour cela, nous avons réalisé un fichier texte regroupant toutes ces informations, et la classe *Monde* vient lire ce fichier et créer un tableau de lieux qui lui correspond.

De plus, il nous a fallu créer un plateau de jeu dont la taille s'adapte proportionnellement à la taille des écrans de chacun. En utilisant les outils appris au S3, nous avons créé une IHM pour une résolution adaptée à tous les écrans utilisés, par une simple relation de proportionnalité.

Puis, nous nous sommes posés la question du moyen utilisé pour donner toutes les informations nécessaires sur le plateau sans surcharger celui-ci. Nous avons donc opté pour une association de chiffres, de couleurs et de symboles. Par exemple, la quantité de pions maladies sur une case est représenté par un chiffre, qui change de couleur quand cette case est touchée par une éclosion de maladie. Les pions des joueurs sont représentés par les points de couleur et les centres de maladies par des symboles blancs.

Quant à la partie algorithmique du code, il a fallu trouver un moyen de connaître le nombre d'arêtes minimum qui sépare les lieux, afin de connaître le nombre d'actions qu'il coûtera à un joueur pour se rendre d'un lieu à un autre à l'aide du déplacement normal.

Également, la gestion de la propagation de la maladie était un problème à résoudre, car il fallait trouver une méthode permettant d'identifier les lieux touchés en cas d'éclosion (pour plus d'informations sur les règles, voir la fenêtre d'aide dans le jeu).

C/ Le principe de l'algorithme

Déplacement : On crée pour commencer une liste de lieux à traiter (T), et une liste de lieux non traités (NT). On ajoute le lieu de départ dans NT. Un lieu est non traité lorsque l'on a pas encore parcouru ses voisins.

En commençant par le lieu de départ du joueur, on parcourt la map par cercle de voisins en stockant les voisins atteints dans NT, jusqu'à atteindre le lieu souhaité, quand tous les voisins d'un lieu sont parcourus il est ajouté à T. On initialise au début de cet algorithme une variable qui a pour valeur le nombre d'action restante au joueur. A chaque nouveau cercle de voisins parcouru on décrémente cette variable si elle prend une valeur négative avant que lieu souhaité soit atteint, alors le joueur ne

peut pas s'y rendre, sinon alors la variable correspond au nombre d'action restante après le déplacement.

Propagation des maladies: On crée une liste des lieux touchés par la propagation. On y ajoute le lieu pour lequel on souhaite au départ ajouter une maladie. Puis on parcourt cette liste, pour chaque lieu qui y est présent: si son nombre de maladie est inférieur à 3 alors on lui ajoute une maladie, sinon on ajoute ses voisins qui ne s'y trouvent pas encore à la liste .

Gestion de la quantité d'actions par joueur: le joueur possède un attribut *nbAction*, modifié à chacune de ses actions et réinitialisé à chaque début de tour par la méthode *tourSuivant*.

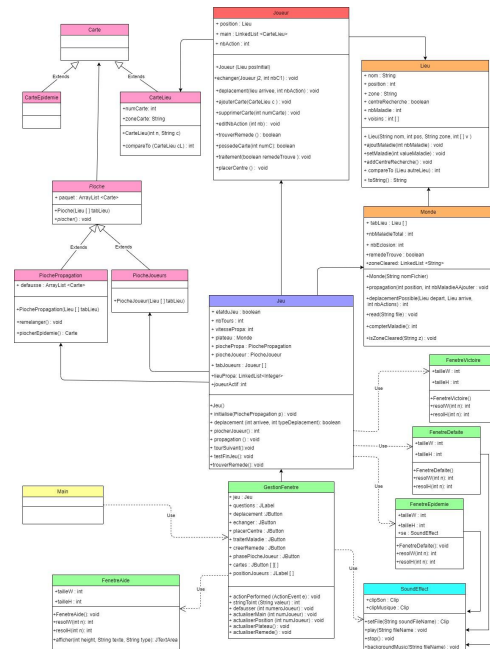
Gestion de la main des joueurs: Lorsque la taille de la main d'un joueur dépasse 5 cartes, *GestionFenetre* utilise la méthode *defausser*. Cette méthode prend en entrée le numéro du joueur et lui demande quelle carte il veut défausser via un *JOptionPane*, tant que la taille de sa main est supérieure à 5. Les cartes choisies sont défaussées, et la main actualisée.

Création de la pioche Joueur: A partir de l'attribut *paquet* de sa classe mère *Pioche*, nous utilisons la méthode *subList* afin de diviser cette pioche en 3 parties et ajouter une carte Epidemie dans les 2 premiers sous-paquets, dans le but d'équilibrer la pioche.

D/ La bibliographie

- PANDEMIC - Le Tutoriel. (2017, 20 octobre). [Vidéo]. YouTube.
<https://www.youtube.com/watch?v=dOvD9lDfZuY>
- Coder en Java - CodeurJava. (2021). CodeurJava. <http://www.codeurjava.com/>
- Stack Overflow - Where Developers Learn, Share, & Build Careers. (2021). Stack Overflow. <https://stackoverflow.com/>
- Adding sound to our game. Java Game Programming for Beginners 7. (2012, août 21). [Vidéo]. YouTube. <https://www.youtube.com/watch?v=kbzT3miTWqI>
- Diagrammes de classes UML, les bases. dans Moodle INSA Lyon, FIMI, Informatique, PC ressources communes, Documents principaux.
<https://moodle.insa-lyon.fr/course/view.php?id=2615>

E/ La structuration des données



https://drive.google.com/file/d/1_4SV-1glStueEuXyzRCG-TfBm_BnzdCr/view?usp=sharing

Notre programme s'exécute à partir de la classe *Main*. Cette classe lance la classe *GestionFenetre* qui gère l'IHM principale du jeu et utilise les classes *FenetreAide* et *SoundEffect*. *FenetreAide* présente les règles du jeu lorsque que l'utilisateur clique sur le bouton créé par *GestionFenetre*. *SoundEffect* est la classe qui gère la musique et les bruitages du jeu.

GestionFenetre a un attribut de type *Jeu*; *Jeu* contient l'algorithme principal du fonctionnement de notre jeu. Cette classe initialise le plateau de jeu, gère l'avancement de la partie, et teste les conditions de victoire/de fin de jeu/d'événements. En effet, elle utilise les classes *FenetreVictoire*, *FenetreDefaite* et *FenetreEpidemie*, qui gèrent respectivement l'affichage en cas d'une victoire, d'une défaite ou du déclenchement d'une épidémie.

L'interface *Carte* a 2 classes filles: *CarteEpidemie* et *CarteLieu*. Ces classes caractérisent les cartes du jeu, qui constituent les mains des joueurs et les pioches. La classe *Pioche* est une classe abstraite qui contient la méthode *piocher*, et a pour attribut une ArrayList d'éléments de type *Carte*. Elle a pour classes filles *PiochePropagation* et *PiocheJoueurs*, qui permettent respectivement de gérer les cartes propagation et les cartes joueurs. La classe *PiochePropagation* contient également des méthodes permettant de gérer les mélanges dûs aux épidémies lors de la partie. La classe *Jeu* a des attributs de type *PiochePropagation* et *PiocheJoueur*.

Jeu a pour attribut un tableau 1D d'éléments de type *Joueur*. La classe *Joueur* contient en attributs sa position (de type *Lieu*), sa main(LinkedList de *CarteLieu*), et son nombre d'actions à un moment donné, ainsi que les méthodes permettant de gérer l'intégralité des actions possibles d'un joueur lors d'une partie.

L'attribut *plateau* de la classe *Jeu* est de type *Monde*; cette classe gère les actions passives du plateau (propagation de la maladie par exemple) et contient les caractéristiques du plateau à un moment donné. La classe *Monde* a donc un tableau de *Lieu* comme attribut.

La classe *Lieu* caractérise chaque case du plateau de jeu (la zone, le nombre de maladies sur cette case, ses voisins, présence de centre de recherches ou non...). Elle contient les méthodes relatives aux changements à effectuer sur les cases lors de l'avancement du jeu (ajout de maladie par le jeu, ou placement d'un centre de recherche par un joueur).

F/ Suggestions d'améliorations de votre projet, bugs connus

Concernant les bug que nous avons décelé dans le jeu, nous avons réussi à tous les éliminer. Toutefois, il y a des aspects du jeu que nous pourrions améliorer.

D'une part, il y a des améliorations que nous pourrions réaliser assez facilement si nous avons plus de temps. Par exemple, nous pourrions augmenter le nombre de lieux et créer différentes maladies pour se rapprocher des règles du vrai jeu Pandemic. De plus, nous pourrions aussi améliorer l'interface pour rendre le plateau de jeu plus lisible et plus intuitif. Finalement, ajouter les "rôles" aux personnages est aussi une modification envisageable bien qu'un peu plus complexe.

D'autre part, nous avons aussi imaginé une amélioration bien plus complexe mais qui rendrait le jeu bien plus proche du vrai jeu pandemic. En effet, nous pourrions réaliser un jeu en ligne où les joueurs ne peuvent pas voir les cartes des trois autres joueurs. De même, nous pourrions réaliser une fenêtre en début de partie où on choisit la difficulté de la partie que l'on veut mener.

G/ Carnet de route et échéancier décrivant l'évolution de votre travail au fil des semaines

Semaine 1 01/03/21	Semaine 2 08/03/21	Semaine 3 15/03/21	Semaine 4 29/03/21	Semaine 5 22/03/21(autonomie)	Semaine 6 05/04/21 (férié)	Semaine 7 12/04/21 (congé)	Semaine 8 19/04/21 (congé)	Semaine 9 26/04/21	Semaine 10 03/05/21
Discussion autour du choix du projet									
	Recherches autour du jeu pandemic, Appropriation des règles, éventuelles modifications des règles								
	Premier Diagramme UML								
		Répartition des classes entre les membres de l'équipe							
		Programmation des classes en parallèle							
					Premier Bilan sur le jeu				
					Correction des bugs, améliorations éventuelles des algorithmes, IHM plus esthétique, ajout de musique si possible				
		Mise à jour du diagramme UML au long de l'avancement du jeu et de la rédaction du code							
						Rédaction du rapport			

Echéancier de notre projet

Lors de la première séance, nous sommes tous arrivés avec des idées de projets différents mais nous avons en commun de vouloir créer une version numérique d'un jeu de plateau. Dans le contexte actuel, il nous a paru judicieux de modéliser le jeu de plateau *Pandemic*, et ce sur le campus de la Doua. Nous avons donc développé cette idée lors de la deuxième et de la troisième séance en définissant le cahier des charges que notre jeu devait respecter, et en décidant des simplifications que nous allions apporter au jeu dans le but de conserver un jeu équilibré et simple d'utilisation. Nous nous sommes ensuite répartis les différentes classes, et lors des séances suivantes nous faisons un point pour savoir où chacun en était et pour conclure sur l'avancement du projet.

H/ Le pourcentage d'implication de chaque membre de l'équipe.

Coline Lahousse	27%
Sarah Gouasmi	23%
Mathéo Joseph	27%
Maxime Giordana	23%