

1. Description du projet

Dans le cadre du module M2201, vous êtes invité(e) à réaliser un projet dont la réalisation s'étendra sur plusieurs semaines. Ce projet a pour objectif de vous faire utiliser les notions et algorithmes vus jusqu'à maintenant sur les graphes.

Ce projet consiste en la réalisation, par groupes de trois ou quatre, d'un ensemble d'IA pour un jeu de stratégie qui vous est fourni. Le langage utilisé sera le Java et un rapport est demandé au format PDF. Pour réaliser ce projet, vous disposez de quatre séances de TP. Durant ces séances, la présence est obligatoire et vous pourrez poser des questions aux différents chargés de TP.

Vous devrez envoyer par mail une archive au format 7z (uniquement) contenant l'intégralité du jeu et de votre code. Un seul mail est demandé par groupe et il est impératif que vos archives **ne contiennent pas de fichier exécutable et soit au format 7z** (sans quoi vos mails seront refusés par le serveur de l'IUT). Ce mail devra être envoyé avant le **vendredi 13 mai 23h59** à l'adresse suivante :

matthieu.simonet@iut-dijon.u-bourgogne.fr.

Le sujet du mail devra impérativement débiter par **[S2_projetGraphe_X]** où X devra être remplacé par votre demi-groupe de TP (A1,A2,...). Le mail devra contenir de façon explicite les noms des différents membres du groupe.

2. Travail attendu

2.1. Principe du jeu

SpaceConquest est un jeu à deux joueurs : les licornes et les zombies. Chacun des deux joueurs dispose d'un vaisseau pouvant se déplacer sur une carte à cases hexagonales en respectant les règles suivantes :

- Chaque vaisseau dispose chaque tour de 2 points d'action (PA).
- Se déplacer sur une case vide voisine de la case actuelle coûte toujours 1 PA.
- Aucun des deux vaisseaux ne peut se déplacer sur une case occupée par une étoile.
- Le vaisseau des licornes étant plus massif que celui des zombies, se déplacer sur une case d'astéroïde lui coûte 2 PA.

L'objectif des licornes est de regagner leur planète d'origine et celui des zombies est de rattraper le vaisseau des licornes.

2.2. Construction des graphes

Question 1 : Création du graphe vide

Dans la classe carte, créez une méthode *getGrapheGrille* qui renvoie le graphe modélisant la grille hexagonale (sans contrainte). Les sommets de la grille seront numérotés ligne par ligne en partant du coin haut gauche de la carte. Dans ce graphe, deux sommets seront reliés si et seulement si les cases auxquelles ils correspondent sont voisines. Testez votre méthode sur une carte de taille 2 en faisant afficher la matrice d'adjacence.

Question 2 : Création du graphe pour les zombies

Dans la classe carte, créez une méthode *getGrapheZombie* qui utilise la méthode *getGrapheGrille* pour générer un graphe de la carte puis qui modifie ce graphe afin de traduire le fait que

le vaisseau des zombies ne peut pas se déplacer sur une étoile. (Il est fortement recommandé de créer une méthode *isolerSommet(int i)* dans la classe *graphe* dont vous trouverez la fonction)

Question 3 : Création du graphe pour les licornes

Dans la classe *carte*, créez une méthode *getGrapheLicorne* pour générer un graphe de la carte traduisant (en plus des contraintes précédentes) le fait que rentrer sur une case d'astéroïdes coûte 2 PA au vaisseau des licornes.

2.3. Gestion du mode manuel

Question 4 : Remise à "zéro" des couleurs de la carte

Dans la classe *carte*, créez une méthode *effacerColoration* qui colore toutes les cases de la carte en blanc.

Question 5 : Couleurs et distances

Dans la classe *carte*, créez une méthode *colorationMouvements* qui prend en attribut un couple *c* représentant les coordonnées d'une case et un graphe *g* modélisant les contraintes de déplacements et qui colorent :

- en vert : les cases nécessitant 1PA pour s'y rendre,
- en jaune : les cases nécessitant 2PA pour s'y rendre.

Question 6 : Mise en place de la coloration

Dans la classe *carte*, en vous inspirant de ce qui existe, modifiez la méthode *selectionCase* pour qu'elle affiche la coloration définie dans la question précédente lors de la sélection d'un vaisseau du joueur en cours et efface toute coloration si le vaisseau est sélectionné de nouveau.

Question 7 : Mise en place de la coloration

Dans la classe *carte* modifiez de nouveau la méthode *selectionCase* pour qu'elle permette de plus le déplacement du vaisseau du joueur si celui-ci à été sélectionné au préalable et si le joueur clique sur une case valide.

2.4. Gestion du mode automatique

Question 8 : Mouvement des licornes

Les licornes veulent rentrer chez elles le plus vite possible. Modifiez la méthode *tourDesLicornes* de la classe *TimerPartie* pour implémenter le mouvement automatique des licornes. On coloriera en vert les cases parcourues par le vaisseau des licornes à chaque tour.

Question 9 : Mouvement des zombies

Les zombies sont affamés, à chaque tour, il suivent le plus court chemin vers le vaisseau des licornes. Modifiez la méthode *tourDesZombies* de la classe *TimerPartie* pour implémenter le mouvement automatique des zombies. On coloriera en jaune les cases parcourues par le vaisseau zombie à chaque tour.

2.5. Gestion des shadoks

Les shadoks disposent d'une planète d'origine : la planète shadok, et d'un vaisseau : la fusée shadok. Cette dernière n'est pas équipée de commande directionnelle (petite erreur de conception) et se déplace aléatoirement autour de la planète shadok. Cependant, à cause de l'attraction shadokienne, la fusée ne peut se déplacer à plus de 3 cases de la planète shadok (case de la

planète non comprise).

Question 10 : Gestion des shadoks

Dans le mode automatique, rajoutez un troisième joueur : les shadoks. Modifiez les classes Partie et TimerPartie pour inclure les nouveaux éléments de jeu et gérer les mouvements de la fusée shadoks. On coloriera en rouge les cases parcourues à chaque tour par la fusée shadok.

2.6. Rencontre avec les Gibis

Les licornes sont rentrées en contact il y a de nombreuses années avec le peuple technologiquement avancé des Gibis qui les ont mis en garde contre le caractère imprévisible de l'exploration spatiale shadok. Le vaisseau des licornes est depuis équipé d'un module l'empêchant de s'arrêter sur une case située à moins de 2 cases de celle de la fusée shadok.

Question 11 : Optimisation Gibi

Modifiez votre code afin de prendre en compte cette nouvelle invention.

Les Gibis ont aussi fait quelques remarques sur le module de navigation des licornes. Ils ont remarqué deux limites à ce module.

Question 12 : Navigation suicidaire

Actuellement rien n'empêche le vaisseau licorne de se jeter sur le vaisseau zombie, corrigez cette erreur de conception du module de navigation (manuelle et automatique).

Question 13 : Une vision d'avenir

Dans un futur proche, les licornes devraient être capables de coloniser de nouveaux mondes. Modifier votre programme pour rendre ceci possible et faire que le module de navigation automatique des licornes les dirige vers la planète la plus proche de leur position.

2.7. Rapport

Pour chacune des questions précédentes, expliquez le fonctionnement de votre programme en justifiant entre autre le choix des algorithmes utilisées.