

Projet - développement java : *un jeu de bataille navale*

Sujet

Vous êtes chargé du développement en java d'un jeu de bataille navale qui permet à un joueur humain d'affronter une intelligence artificielle. Pour ce projet, vous pouvez utiliser la structure qui vous est proposée en aide (diagramme de classe et prédéveloppements commentés associés). Rien ne vous empêche cependant de reprendre le développement depuis le début, et de proposer un modèle objet différent si vous pouvez le justifier.

Règles de jeu

Les bateaux

Chaque joueur dispose de deux bateaux occupant trois cases.

- Un bateau radar/tireur : ce bateau a la capacité de faire une saisie radar (le signal est dégradé proportionnellement à l'état du bateau), et de tirer une fois par tour. Couler ce bateau (toucher les trois cases) permet de gagner la partie.
- Un bateau brouilleur qui a la capacité de brouiller le signal radar, si le brouilleur associé à la case avant n'est pas endommagé.

Les espaces associés aux joueurs

Les bateaux se déplacent dans des environnements séparés, de 10 cases de largeur sur 5 cases de hauteur.

Actions de déplacement par tour : un bateau peut se déplacer d'une case en avant ou en arrière, ou effectuer une rotation de 90 degrés, au sein de son espace.

- Un bateau radar/tireur effectue une action aléatoire de déplacement par tour quelque soit son état.
- Un bateau brouilleur effectue deux actions de déplacement par tours, à moins que la case arrière soit touchée, ce qui immobilise le bateau.

Représentation graphique des espaces

On utilise une représentation en mode console de l'état du monde (5 lignes de dix caractères par espace).

- Représentation de l'espace du joueur IA, perçu par le radar du joueur humain (sur la base de l'environnement brouillé envoyé par le brouilleur de l'IA) : un caractère pour la perception d'un signal sur une case (X), un caractère lorsqu'aucune perception n'intervient(0).

- Représentation de l'espace du joueur humain : un caractère associé à l'eau (O), un caractère associé à un impact(I), un caractère associé à une case bateau(X), et les caractères spéciaux moteur(M) et brouilleur(B).

On obtient une représentation de la forme suivante :

Joueur IA :

```
000X000000000000
0000X0X00000X00
000000000XX0000
0000000000000000
0000000000000000
```

Joueur Humain :

```
0000M00000000000
0000X00000000000
0000B0000IXX000
0000000000000000
0000000000000000
```

Déroulement du jeu :

Les joueurs effectuent leurs actions de perception et de tir successivement, jusqu'à ce qu'un bateau radar/tireur soit coulé. À chaque coup, le joueur humain saisit les coordonnées de la case visée, l'IA sélectionne aléatoirement une case parmi l'ensemble des cases perçues par son radar (auparavant brouillées par le brouilleur de joueur humain).

Objectifs du projet

Ce projet a pour objectif de vous familiariser avec le développement sous l'IDE (interface de développement) NetBeans ou Eclipse au choix. Ainsi, vous aurez à vous documenter sur la création d'un projet, l'ajout de classes, le lancement de l'exécution, etc. Netbeans semble le plus intuitif, ainsi je vous conseille de privilégier cet IDE dans un premier temps.

D'autre part, vous aurez à respecter autant que possible les principes de modélisation objet décrits dans les supports de cours qui vous ont été fournis. Ce projet ne doit pas être développé en langage procédural, mais doit faire intervenir différents objets qui se répartissent les fonctionnalités à mettre en place. Pour cela, vous devez respecter le modèle objet qui vous est proposé dans ce sujet (quelles classes utiliser, quelles relations entre les classes, etc.). Rien ne vous empêche cependant de faire les aménagements que vous trouvez souhaitables de celui-ci si celles-ci sont justifiées.

Le projet sera évalué sur la base des aspects fonctionnels développés (respect des fonctionnalités décrites ci-dessus), et sur la qualité du développement objet que vous proposez.

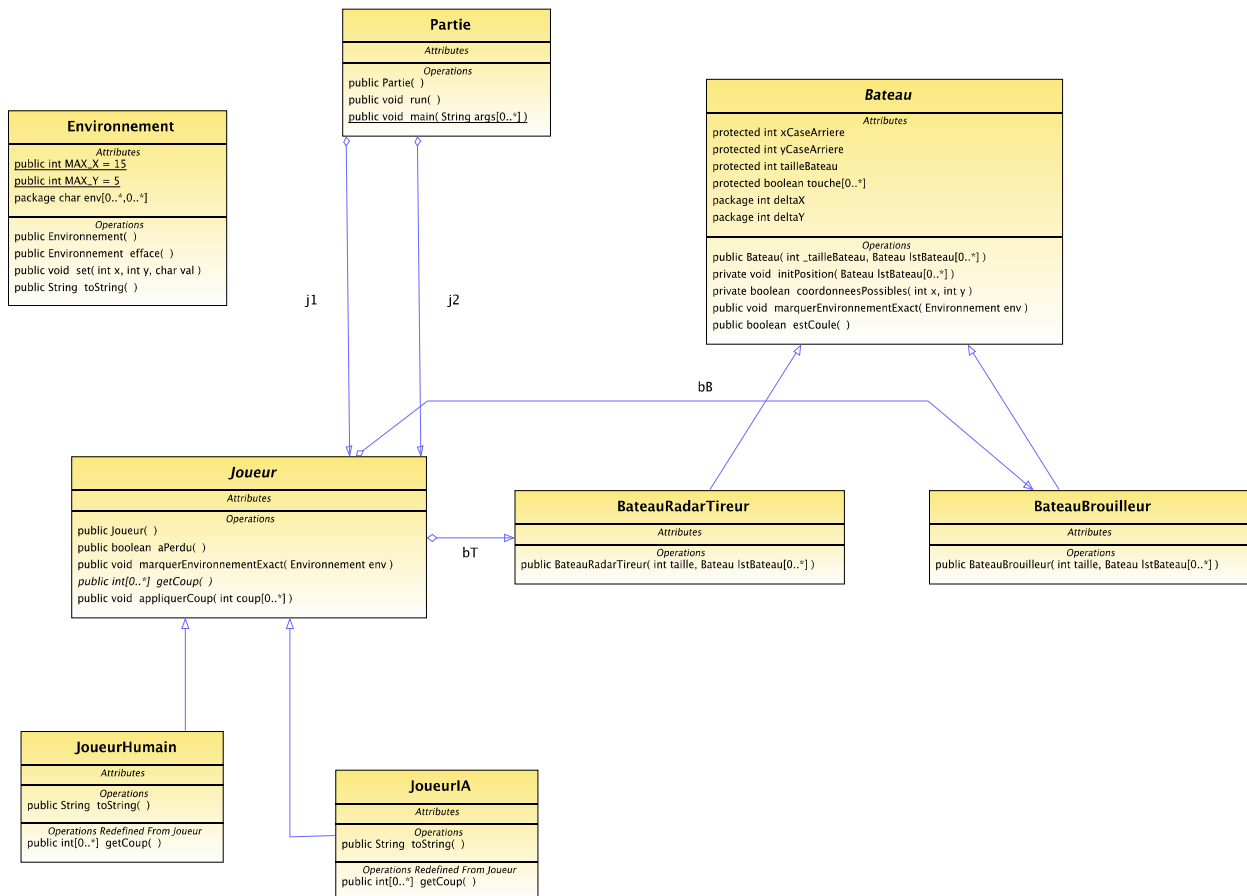
Rendus : le code informatique et un rapport explicatif de votre contribution.

Description des prédéveloppements à votre disposition

Le modèle

Les différents objets/classes que l'on définit nous permettent de distribuer les fonctionnalités entre objets. Chaque objet a un domaine de compétences, et collabore avec les autres objets du système. La collaboration est réalisée par appels de fonctions/procédures entre objets.

Un bateau est un objet de type Bateau, on distingue des bateaux radar/tireur et des bateaux brouilleurs. Les radars et les brouilleurs sont aussi des objets. On représente la partie comme un objet, les espaces de jeux (Environnement) comme des objets et les joueurs comme des objets. On obtient alors le diagramme de classes ci-dessous (se référer au cours) :



Un premier modèle de classes

Ce diagramme ne comprend pas les classes Radar et Brouilleur, qu'il vous faudra ajouter.

Ce qui est fait, .. et ce qu'il reste à faire

Les différentes classes du modèle de classes ci-dessus sont partiellement implémentées. Lorsque vous exécutez le main (point d'entrée du programme) de la classe Partie, deux joueurs sont créés (humain et IA), les bateaux de ces joueurs sont positionnés aléatoirement, un affichage simplifié des environnements des deux joueurs est présenté, le joueur humain saisit les coordonnées de son premier coup.

Vous pouvez commencer par examiner le comportement de la fonction `run` de partie, qui décrit le déroulement de haut niveau (saisie des coup, affichages, etc.). Vous pourrez alors proposer l'implémentation des fonctionnalités manquantes :

Étape 1 : Un jeu naïf

- exécution du coup d'un joueur sur un autre joueur : un joueur reçoit un coup, et le redistribue sur ses bateaux. Les bateaux, s'ils sont concernés par les coordonnées du coup enregistrent les débats.
- Écrire la fonction qui revoit vrai lorsqu'un bateau est coulé
- Écrire la génération des coups par le joueur AI
 - principe : sur la base d'un environnement perçu (cases 'X' et 'O'), l'IA sélectionne aléatoirement une case 'X' (les coordonnées de la case composent le coup). Modifier la signature de la fonction `getCoup` du joueur IA, en lui fournissant un environnement en entrée.
 - Remarque : examinez le fonctionnement de la fonction aléatoire (`Math.random()`) dans le constructeur de bateau.
- Écrire les processus de déplacement des bateaux. Appeler ces processus entre les coups des joueurs depuis la fonction `run` de la partie.

À ce niveau de développement, le programme est fonctionnel, mais l'IA et le joueur humain perçoivent la configuration adverse exacte.

Étape 2 : A vos manettes ..

- Développer les classes Radar et Brouilleur. Le radar prendra en paramètre un environnement (case 'X' et 'O'), et déformera le signal en fonction de son état de fonctionnement. Vous proposerez une fonction de déformation dont l'efficacité sera proportionnelle à l'état du bateau associé. Le brouilleur fonctionnera de la même façon (mais cette fois, la qualité du brouillage sera nulle lorsque celui-ci sera endommagé). Si vous pensez que les fonctionnalités de ces objets sont proches, essayez de proposer un modèle de classes qui permet de factoriser celles-ci.
- Intégrer la nouvelle visualisation dans le développement global (perception exacte de l'environnement du joueur humain (avec affichage des cases spéciales), perception bruitée de l'environnement de l'IA).
- Compléter le code de façon générale.

Méthode

À chaque étape de votre développement, demandez-vous si vous respectez les points suivants, garants de la qualité de votre modèle objet :

- « *Le code que je place ici est-il bien de la responsabilité de l'objet associé à la classe courante ?* »

Exemple : une erreur classique serait de calculer l'impact d'un coup sur un bateau dans la classe Joueur. Le bateau est responsable de lui-même. C'est lui qui doit réaliser ce type d'opération. Il faut donc préférer redistribuer le coup sur les bateaux, par appels de fonctions. Le joueur réceptionne le coup par appel de fonction, et le redistribue par appels de fonction sur chacun de ses bateaux.

- « *Puis-je factoriser la fonctionnalité que je développe ?* »

Exemple : Le processus que vous développez dans BateauBrouilleur est-il spécifique au BateauBrouilleur. Ce processus est utile au bateau TireurRadar ? Si oui, envisagez de le placer plus haut dans la hiérarchie de classes, dans la classe Bateau.

Utilisation de NetBeans

Configuration de votre projet (exemple sous NetBeans)

Création d'un nouveau projet

File->New Project : sélectionner *Java Application*

Si vous sélectionnez la création de la classe main, alors vous disposez du point d'entrée de votre programme : la fonction `public static void main(String[] args)`. Lorsque vous appuyez sur run, celle-ci est exécutée. Ajouter `System.out.print("coucou");` dans le corps de la fonction afin de vérifier que le message « coucou » s'inscrit bien dans la console.

Vous pouvez alors ajouter des classes (Clic droit sur votre projet dans l'arborescence de gauche puis new->java class).

Ouverture d'un projet existant (tel que celui joint à ce sujet)

File-> Open Project

Trucs et astuces

- Lorsque NetBeans (ou Eclipse) souligne en rouge un mot du code, il est possible que celui-ci ne soit pas reconnu. Il est alors nécessaire d'importer (entête de fichier) une classe afin de le reconnaître. Vous pouvez utiliser Ctrl + Espace en vous plaçant sur le mot, il vous sera alors proposé les ajouts de packages qui seront automatiquement insérés en entête de fichier.

- Si vous souhaitez des informations complémentaires sur les classes standards de java, référez-vous à la documentation en ligne : <http://java.sun.com/javase/6/docs/api/>

- Shift + Alt + F : Formatage automatique du code sélectionné sous NetBeans