

Projet de Langages Web

L3 Informatique

2016–2017

À déposer sur la plate-forme UniversiTICE au plus tard pour le **dimanche 2 avril 2017**.
Groupe de deux étudiants maximum.

Le projet consiste à réaliser un jeu de conquête spatiale qui se joue par tour.

1 Principe du jeu

Le jeu se joue sur une grille représentant la galaxie. Certaines cases de la grille comportent une planète qui peut accueillir des colons. Une planète est soit neutre, c'est-à-dire qu'elle est peuplée mais par aucun des joueurs, soit conquise c'est-à-dire peuplée par un des joueurs. Chaque planète possède une propriété de croissance démographique, comprise entre 1 et 10, qui à chaque tour vient s'ajouter à sa population pour les planètes colonisées (la population des planètes neutres reste constante).

Au début de la partie, chaque joueur dispose d'une planète avec une population initiale. À partir de chaque planète qu'il possède, le joueur peut affréter une flotte en prenant une portion de la population actuelle de la planète et envoyer cette flotte vers une autre planète de la carte. Le temps nécessaire à cette flotte pour atteindre sa cible (c'est à dire le nombre de tours) est fonction de la distance de la cible (nombre de cases). Lorsqu'une flotte atteint sa cible, plusieurs cas peuvent se présenter :

1. la planète appartient déjà au joueur : dans ce cas les colons de la flotte viennent s'ajouter à la population de la planète ;
2. la planète appartient à l'adversaire : c'est la guerre ! Le nombre de colons est comparé à la population de la planète. Celui qui a le plus grand nombre remporte la victoire et la différence entre nombre de colons et population locale devient la nouvelle population de la planète ;
3. la planète est neutre : dans ce cas la bataille se joue contre les indigènes de la même manière.

Lorsque la population d'une planète tombe à 0, elle devient neutre. Les planètes inhabitées ne lancent pas d'attaque.

Chaque tour se déroule de la manière suivante :

1. Les populations de chaque planète sont mises à jour.
2. Les positions des flottes sont mises à jour.
3. Les batailles des flottes qui sont arrivées à destination sont résolues, le statut des planètes est mis à jour et le joueur est informé du résultat par un message du genre « la planète X a tenu contre l'attaque de JoueurY », « la planète X est tombée entre les mains de JoueurY », ...

4. Le joueur donne ses ordres aux flottes de ses planètes où la population est strictement supérieure à 1. Lorsqu'il a fini de donner ses instructions, il met fin à son tour.

Un joueur a perdu lorsqu'il ne possède plus de planète et qu'il n'a plus de flotte en transit. Le gagnant est le dernier en jeu.

Pour donner ses instructions, le joueur clique sur la planète source puis la planète cible, saisit le nombre de colons à envoyer, puis valide ou annule l'ordre. Il peut recommencer l'opération tant qu'il lui reste des populations strictement supérieures à 1.

Les informations suivantes devront être affichées sur la carte :

- les flottes en transit ;
 - le nombre de colons et la destination de chaque flotte du joueur (mais pas ceux des flottes adverses) ;
 - la population des planètes conquises par un joueur (mais pas celle des planètes neutres).
- Le jeu proposera des grilles prédéfinies (stockées dans des fichiers ou une base de données) comportant :

- des dimensions variables ;
- des planètes placées sur la grille, nommées et affectées d'une croissance démographique ;
- une liste ordonnée des planètes pour le placement des joueurs en fonction de leur nombre.

2 Mise en œuvre

La page d'accueil permet de choisir la carte et le nombre de joueurs. Pour rester simple, on propose de n'implémenter qu'un mode de partie à un seul joueur humain avec une intelligence artificielle rudimentaire pour les autres joueurs.

Une fois que la partie commence, la carte (une grille rectangulaire, qui pourra par exemple être représentée par un tableau HTML) est affichée. Le joueur saisit l'ensemble de ses ordres, chacun de la forme (source, destination, quantité de colons). Lorsqu'il a fini, il clique sur un bouton pour terminer son tour et ses ordres sont traités par le programme qui résout les conflits. La carte se met à jour.

2.1 Contraintes

Le jeu proprement dit sera écrit en JavaScript ; on ne communiquera avec le serveur qu'au début de la partie (pour charger la carte).

Le langage utilisé côté serveur sera PHP.

Vous pouvez utiliser un framework côté client ou côté serveur mais dans ce cas, il faudra bien expliquer dans votre rapport l'architecture de votre code et notamment quels fichiers sont fournis par le framework et lesquels ont été écrit par vos soins.

La présentation sera définie dans des fichiers CSS et en aucun cas avec l'attribut `style` des éléments HTML.

2.2 Améliorations possibles

Si tout ce qui précède fonctionne bien, on pourra ajouter une ou plusieurs des fonctionnalités suivantes :

- un éditeur de carte ;
- une intelligence artificielle améliorée (dans ce cas, il faudra expliquer dans le rapport comment elle fonctionne) ;
- la possibilité de sauvegarder la partie en cours et de la continuer plus tard.

3 Contenu de l'archive à envoyer

Vous enverrez une archive au format zip contenant tous vos fichiers sources. Vous réaliserez également un rapport explicatif bref présentant toutes les fonctionnalités de votre programme ainsi que les différents problèmes rencontrés et les solutions mises en œuvre.

Comme il ne sera peut-être pas possible d'organiser des soutenances, il est important (ce sera pris en compte dans la note) :

- que votre code soit lisible et commenté ;
- que le rapport permette de comprendre facilement quelles fonctionnalités (obligatoires ou facultatives) ont été réalisées ;
- et que votre projet soit facile à déployer. Par exemple, si votre application utilise une base de données, il est recommandé de fournir une interface d'administration en PHP qui permettra de saisir le nom de la base de données à utiliser, l'identifiant et le mot de passe du compte de connexion à la base et éventuellement un préfixe pour les noms des tables que votre application va y créer.