Module Nouvelles Technologies du Web

TME - Développement avec Play! Framework

I°) Objectifs

L'application à réaliser sera un petit jeu de Risk. Votre application pour envoyer des troupes attaquer l'application d'un autre étudiant. En parallèle vous pourrez subir des attaques. Enfin vous afficherez des modestes statistiques de vos actions.

II°) Installation

Télécharger la version 2.2 de Play! sur le site. Respectez les étapes d'installation vu en cours.

III°) Arbitre

Dans cette partie nous allons écrire un controlleur permettant d'enregistrer les planètes des étudiants désirant livrer bataille. Il s'agit d'un "cache de client".

Créer une classe "ArbitreController" héritant de Controller.

Cette classe possède 3 méthodes :

1. subscribe : Ajoute le client faisant la requête HTTP. Si celui-ci est déjà présent, il est écrasé. La méthode host de request() devrait vous aider.
2. unsubscribe : Supprime le client faisant la requête HTTP. Si celui n'était pas présent, cettte méthode retourne une erreur 404
3. getPlanetes : Renvoie un tableau JSON contenant la liste des clients enregistrés.

Mettez à jour le fichier routes. Toutes ses requêtes sont de type GET (donc utilisable via un navigateur)

Valider votre travail

IV°) Joueur

Créer une classe PlayerController héritant de Controller

Le joueur possède actuellement deux méthodes :

1. attack(host, unit) : Envoie le nombre d'unités spécifiées attaqué l'host spécifié. Il faut donc supprimer les unités envoyées. Si le nombre d'unités à envoyer est plus grand que le nombre d'unités que l'on possède, un badRequest est renvoyé, et rien n'est effectué. Le résultat obtenu est soit un code 200, on a tué des soldats ennemis, soit la mort de l'ennemi.

2. defend(unit) : Est appelée par un attaquant. Cette méthode réduit le nombre d'unités que l'on possède par le nombre spécifié. Si il reste au moins une unités, un ok est renvoyé. Sinon, on renvoie un forbidden et on met notre nombre d'unités à 500.

Mettez à jour le fichier route. Ces méthodes sont de type GET. Valider votre travail.

V°) Statistiques

Créer une classe statistiques comprenant 4 entiers représentant les nombres de victoires, attaques, défense, défaites.

Ajouter une statistique dans PlayerController et mettez à jour les méthodes d'attaque et de défense pour prendre en compte les statistiques.

Créer un fichier stats.scala.html dans le package views. Ce template affiche les statistiques qu'on lui donne en paramètre.

Enfin, créer une méthode stats dans PlayerController qui fait afficher le template stats.scala.html et qui lui donne les statistiques à afficher.

Mettez à jour le fichier route. Ces méthodes sont de type GET. Valider votre travail.