

Projet de Langages Web

L3 Informatique

2017–2018

À déposer sur UniversiTICE au plus tard pour le **dimanche 1^{er} avril 2018**.
Groupe de deux étudiants maximum.

Le projet consiste à réaliser un jeu où le joueur doit défendre une base des attaques ennemies en construisant des canons pour éliminer les assaillants.

1 Principe du jeu

Le jeu se joue sur une grille rectangulaire. À intervalles réguliers, des ennemis traversent la carte pour chercher à atteindre la base du joueur (pour simplifier, on supposera qu'ils suivent toujours le même chemin). Pour les éliminer, le joueur peut placer des canons sur les cases de la grille (en dehors de l'itinéraire des ennemis). Quand un ennemi est tué, le joueur gagne des crédits. S'il a suffisamment de crédits, il peut acheter de nouveaux canons pour se défendre face aux ennemis qui sont de plus en plus nombreux et résistants au fil du temps.

2 Mise en œuvre

L'écran d'accueil permet de choisir une carte parmi celles que le jeu propose. Les cartes seront stockées côté serveur dans des fichiers ou dans une base de données. Les différentes cartes pourront se distinguer par leurs dimensions ou par l'itinéraire emprunté par les ennemis.

Une fois que le joueur a choisi une carte, celle-ci est affichée. Les ennemis commencent à arriver en suivant l'itinéraire défini par la carte. L'affichage doit mettre ce dernier en évidence (par exemple avec une couleur de fond spéciale).

À tout moment, le joueur peut sélectionner un type de canon parmi ceux qui sont accessibles compte tenu des crédits dont il dispose et il peut le placer sur la carte. Quand un ennemi est à portée d'un canon, celui-ci lui tire dessus et l'ennemi perd des points de vie. Il n'est pas nécessaire que les canons tirent des projectiles mais l'affichage doit montrer la perte de points de vie des ennemis (par exemple avec une barre de santé qui diminue). Si l'ennemi n'a plus de points de vie, il meurt et le joueur gagne des crédits.

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).

Au moins 2 types d'ennemis apparaîtront dans le jeu. Chaque type d'ennemi aura un nombre de points de vie et une vitesse de déplacement.

De même, il sera possible de placer au moins 2 types de canons, qui auront chacun un prix d'achat, une puissance (nombre de points de vie ôtés à l'ennemi), une vitesse de tir, une fréquence de tir (temps entre deux tirs successifs) et une portée (distance au-delà de laquelle le canon n'atteint pas les ennemis).

Le langage utilisé côté serveur sera PHP.

Vous pouvez utiliser un framework côté client ou côté serveur mais dans ce cas :

- le code fourni par le framework ne doit pas représenter la quasi-totalité de votre projet ;
- le rapport devra justifier le choix du framework utilisé, présenter l'architecture et la philosophie de celui-ci et permettre d'identifier clairement les fichiers écrits 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 :

- une limitation du nombre d'ennemis qui peuvent atteindre la base (si plus de N ennemis arrivent jusqu'à la base, le joueur a perdu) ;
- des canons qui ralentissent les ennemis sans les blesser ;
- des cartes où les ennemis peuvent suivre plusieurs itinéraires, voire se déplacer n'importe où dans la carte (mais le joueur ne doit pas pouvoir les empêcher d'atteindre sa base en plaçant ses canons) ;
- la possibilité de sauvegarder la partie en cours sur le serveur (le joueur devra alors créer un compte sur le site ou se connecter) et de la continuer plus tard ;
- ou toute autre fonctionnalité que vous jugerez intéressante.

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. Si vous avez travaillé en binôme, vos deux noms doivent apparaître sur le rapport.

Comme il ne sera 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. En particulier, 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. Sinon, on pourra définir les paramètres de connexion dans un fichier PHP unique et inclure dans l'archive un fichier SQL permettant de créer et de remplir les tables utiles au programme. Ce fichier ne devra pas créer la base de données ou obliger à lui donner un nom spécifique (pas de `create database`, de `use...`).