

Jeu de tir au canon

On veut faire un jeu de tir au canon où on règle l'inclinaison du canon pour ajuster le tir et toucher une cible.

Vous écrirez les fichiers javascript dans un dossier `src`. Pour pouvoir utiliser les modules ES6 (sans les transformer par Webpack), il faut un serveur Web. Ouvrez une console Node, rendez vous dans votre dossier de travail et tapez `node filesaver.js` pour lancer le serveur. Vous chargerez la page dans le navigateur avec `localhost:8080//dist/canon.html`.

Canon

Écrivez et exportez dans le fichier `canon.js` la classe `Canon` représentant un canon : un canon est construit à partir de la vitesse `vitesseSortie` de sortie des boulets et des paramètres d'inclinaison `paramInclinaison` du canon, objet avec les propriétés `minimum`, `maximum`, `delta` représentant l'inclinaison minimum, maximum et le pas d'incrément, de décrémentation, l'inclinaison (entière en degré) initiale du canon sera l'inclinaison minimum ; la méthode `inclinaison()` renvoie l'inclinaison, les méthodes `leve()` et `baisse()` augmente et diminue l'inclinaison du pas sans sortir des limites ; la méthode `portee()` renvoie la distance maximale de tir = $vitesseSortie^2/g$ où $g=9,81 \text{ m/s}^2$ est l'accélération de la pesanteur, la méthode `tir()` renvoie la distance $vitesseSortie^2 \times \sin(2 \times inclinaison)/g$ à laquelle le boulet tiré tombera, enfin la méthode `etat()` renvoie l'état du canon, c.-à-d. un objet avec les propriétés `inclinaison` et `vitesse` contenant l'inclinaison du canon et la vitesse de sortie des boulets..

Gestion du canon

Récupérez le fichier HTML `canon.html` qui contient les éléments de l'interface pour tirer au canon et afficher les résultats. Récupérez aussi le fichier `eventemitter3.js` qui exporte par défaut la classe `EventEmitteur`, émetteur d'événements que l'on peut écouter.

Récupérez les fichiers `vue.js`, `modele.js` et `controleur.js` qui exporteront des méthodes de la Vue, du Modèle et du Contrôleur. La Vue contient des fonctions d'initialisation au chargement de la page et de ménage quand on quitte la page. Elle a aussi un émetteur d'événement interne qu'elle permet d'écouter par les méthodes `addListener` et `removeListener` qu'elle exporte. Le Modèle a lui aussi un émetteur local qu'on peut écouter. Le Contrôleur a des

fonctions d'initialisation au chargement de la page et de ménage quand on quitte la page. **Faites en sorte** que la page HTML charge le contrôleur.

Écrivez et exportez dans la vue la fonction `afficheCanon(etatCanon)` qui affiche l'état du canon en affichant l'inclinaison du canon au bon endroit. Dans le Modèle, ajoutez une fonction `nouveauCanon()` qui crée et renvoie un nouveau canon, ajoutez une variable qui contient un canon, initialisée avec un nouveau canon, et une fonction `etatCanon()` exportée qui renvoie l'état de ce canon. Faites en sorte avec le contrôleur que le canon soit affiché dans la vue au démarrage.

On veut changer l'inclinaison du canon avec les boutons + et -. La gestion des boutons de l'interface de la vue sera gérée par un seul gestionnaire d'événements. Écrivez ce gestionnaire `onInterface` (vide pour l'instant) et faites en sorte qu'il soit appelé quand on clique dans l'interface. Faites en sorte que lorsque le bouton + est cliqué, la vue émette l'événement `'inclinaison'` avec 1 pour valeur, et que lorsque le bouton - est cliqué, la vue émette l'événement `'inclinaison'` avec -1 pour valeur. Ajoutez et exportez dans le modèle les méthodes `leveCanon` et `baisseCanon` qui lève et baisse le canon, et font en sorte que le modèle émette l'événement `'canonChange'` avec l'état du canon comme valeur si l'inclinaison du canon a changée. **Faites en sorte** que le contrôleur écoute les changements d'inclinaison du canon demandés par la vue et les transmette au modèle. **Faites en sorte** que le contrôleur écoute les changements du canon et les transmette à la vue. **Testez. Modifiez** `nouveauCanon` pour que le modèle mette l'événement `'canonChange'` avec l'état du nouveau canon et simplifiez alors le contrôleur.

Gestion du tir

Quand on appuie sur le bouton `Tirer`, il faut tirer et afficher le résultat du tir, pour l'instant juste afficher la distance du tir. Ajoutez et exportez dans le modèle la fonction `tire` qui fait tirer le canon : il crée un objet contenant les résultats du tir, c.-à-d. un objet contenant la propriété `distance` contenant la distance du tir, l'ajoute en mémoire dans un tableau (pour faire plus tard des statistiques), et émet l'événement `'tir'` avec les résultats comme valeur.