

Projet Web 2024 : petits trains



Ce projet est à faire en binôme. Les technologies attendues sont HTML, CSS et JavaScript; l'usage de jQuery est autorisé mais non indispensable. L'usage de PHP n'est pas autorisé. L'usage de copié-collé, plagiat, recours (même partiel) à des générateurs de code (« intelligences artificielles ») est proscrit, hors partie libre (V2).

L'objectif du projet est de programmer un simulateur de trains. L'internaute peut ajouter des trains, modifier le plateau du simulateur en cours de simulation, et observer le comportement des trains évoluant sur les rails.

⚠ DEUX versions à rendre

Votre archive finale devra être au format `.zip` ou `.tar.gz`, et devra contenir **deux** répertoires et un fichier :


1. un fichier `README.md` contenant
 - (a) vos noms et prénoms (ne pas oublier votre binôme),
 - (b) d'éventuels bugs ou non respect des consignes sur votre première partie,
 - (c) un très bref descriptif de ce que vous avez implémenté dans la deuxième partie, et
 - (d) les éventuelles aides extérieures pour la deuxième partie (source de morceau de code ou algorithme récupéré sur Internet, requêtes effectuées à des générateurs de code automatisés, etc.);
2. un répertoire `TrainV1` correspondant à la première partie, et contenant exclusivement le répertoire `images` et son contenu, le fichier `train.html`, et `css/train.css` (tous ces fichiers doivent être fournis sans aucune modification), ainsi que `js/train.js` (votre fichier);
3. un répertoire `TrainV2` correspondant à la seconde partie (beaucoup plus libre), et contenant ce que vous voulez. Ce second répertoire devra bien sûr contenir tous les fichiers (HTML, CSS, JavaScript, images...) nécessaires à la bonne exécution de votre programme dans sa version améliorée.

L'objectif de ce projet est de construire un simulateur ferroviaire, où des trains avancent le long de rails sur un plateau en deux dimensions. Les trains ne peuvent ni sortir du jeu, ni aller sur des cases incompatibles (sauf améliorations potentielles en deuxième partie); si un train

sort du jeu ou rencontre une case incompatible, il est immédiatement détruit (et disparaît donc instantanément). Les rails sont à *double voie* : les trains peuvent donc se croiser sans problème.


E X E R C I C E


Exercice 1 (Partie 1 : partie guidée).











 **Question 1 :** Récupérer l'archive sous Moodle. Vérifier que son contenu s'affiche correctement. Renommer `train-squelette.html` en `train.html`. Renommer `js/train-squelette.js` en `js/train.js` et modifier le lien depuis `train.html`.


JavaScript pur





Aucune modification des fichiers CSS et HTML fournis n'est autorisée au cours de la première partie.


 **Question 2 :** Après avoir lu tout le sujet, réfléchir aux classes, objets et variables globales dont vous aurez besoin : train, wagons, plateau de jeu, simulateur, etc.

 **Question 3 :** Gérer la partie construction du monde, à l'exception des trains :

- Sélectionner un des boutons (eau, forêt, rails... mais pas ceux des trains) doit permettre ensuite, par un clic sur une case du plateau, de changer cette case dans le mode (eau, forêt, rails...) sélectionné. Les boutons signifient dans l'ordre de gauche à droite :
 -  : forêt
 -  : eau
 -  : rail horizontal
 -  : rail vertical
 -  : rail allant de horizontal vers le haut (une description reformulée est indiquée en commentaires dans la classe `Type_de_case` dans le fichier `.js` fourni)
 -  : rail allant de vertical montant vers la droite
 -  : rail allant de horizontal vers le bas
 -  : rail allant de vertical descendant vers la droite
- Lorsqu'un bouton est sélectionné, il doit passer en désactivé (grâce à une propriété CSS exprimée en JavaScript) afin de montrer à l'internaute que ce bouton est sélectionné. Par exemple, sur la [figure 1a](#), le bouton  est sélectionné car désactivé. Sur la [figure 1b](#), c'est le bouton «  » qui est sélectionné.

 **Question 4 :** Gérer l'ajout des trains. Sélectionner un des boutons de création de trains doit permettre d'ajouter un train sur la case indiquée. La locomotive est ajoutée sur la case sélectionnée, les wagons à sa gauche; la direction du train est initialement vers la droite. Ne pas ajouter le train si tout ou partie du train est hors du plateau. Les boutons ont la signification suivante :

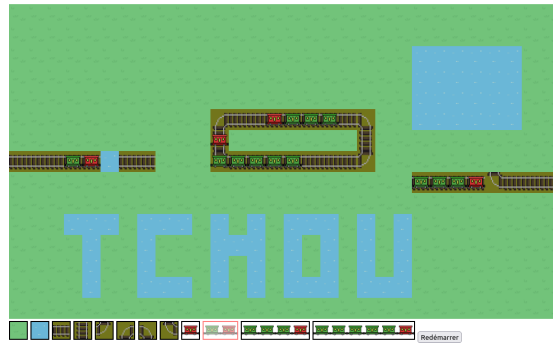
- bouton  : ajout d'une locomotive seule
- bouton  : ajout d'une locomotive et un wagon à sa gauche
- bouton  : ajout d'une locomotive et 3 wagons à sa gauche
- bouton  : ajout d'une locomotive et 5 wagons à sa gauche

 **Question 5 :** Améliorer la partie d'ajout des trains :

- impossible d'ajouter une locomotive ou un wagon sur une case hors rail horizontal (eau, forêt, ...) ni sur un train existant;
- le train est nécessairement horizontal lors de sa création.

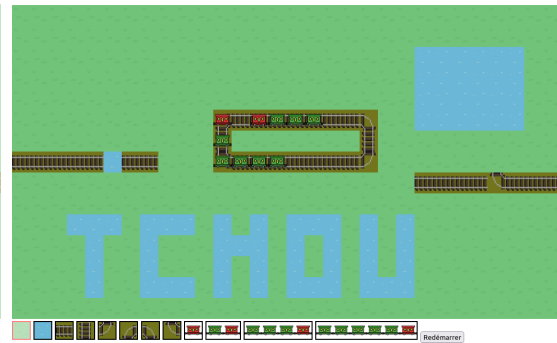
 **Question 6 :** Gérer l'avancement des trains (hors cases incompatibles) :

Simulateur de trains



(a) Temps t


Simulateur de trains




(b) Temps $t + 1$

FIGURE 1 – Simulateur

- ajouter un temporisateur (à l'aide de `setTimeout`);
- la vitesse du simulateur doit être de deux coups par seconde;
- à chaque coup, chaque train doit avancer dans sa direction;
- en cas de rail allant dans une autre direction, la direction de la locomotive doit être actualisée; les wagons suivent toujours la position du wagon (ou de la locomotive) les précédant.

 **Question 7 :** Gérer le bouton pause : lorsque l'on appuie dessus, le simulateur se met en pause. Lorsque l'on rappuie dessus, le simulateur repart. Lorsque le mode pause est activé, ce bouton doit s'appeler « Redémarrer »; sinon, le bouton s'appelle « Pause ».

 **Question 8 :** Prévoir la gestion des cases incompatibles lors de l'avancement des trains. Un train va exploser (en réalité : disparaître instantanément et entièrement du simulateur) si l'une des conditions suivantes est vérifiée :

- il sort en partie du plateau;
- il rencontre une case d'eau, de forêt, ou de rail dans une mauvaise direction.

La [figure 1a](#) montre un exemple de simulateur au temps t . Les locomotives sont affichées en rouge, et les wagons en vert. La [figure 1b](#) montre l'évolution de ce simulateur une unité de temps plus tard. Les deux trains sur la boucle centrale ont évolué d'une case dans leur direction respective. Le train tout à gauche du simulateur a disparu car il a rencontré une case d'eau. Le train tout à droite a disparu car il a rencontré un rail dans une mauvaise direction. Dans ces deux images, le simulateur est en mode pause puisque le bouton s'appelle « Redémarrer ».


Notation

Cette première partie est notée sur 12 points (sur un total de 20). Par conséquent, les points restants seront attribués en fonction de votre créativité (fonctionnalités, ergonomie, graphisme, animations...), comme décrit dans la seconde partie ci-dessous.



E X E R C I C E

Exercice 2 (Partie 2 : créativité).

 **Question 1 :** Améliorer votre projet de la façon qui vous plaît. La notation dépendra de votre créativité, de l'ajout de fonctionnalités, du graphisme, de l'ergonomie...

Toute modification des fichiers HTML et CSS fournis est *autorisée* dans cette partie.

Quelques idées en vrac ci-dessous, dont aucune n'est obligatoire; toute autre idée est bienvenue...!

- Améliorer les graphismes (dont les *sprites*).
- Modifier/améliorer l'ergonomie, en améliorant la disposition des boutons dans la page.
- Améliorer les images des boutons; penser à l'opacité ou à toute autre solution CSS pour indiquer quand ils sont désactivés.
- Soigner les effets CSS : transparence, animation...
- Ajouter des sons, par exemple un « tchou » aléatoire, un son de crissement de roues dans les virages, etc.
- Supposer que les rails sont à voie unique, et donc que des collisions peuvent se produire. Si deux trains entrent en collision, tous deux disparaissent instantanément.
- Lorsque l'on clique sur un train, celui-ci se retourne et change de direction.
- La vitesse des trains peut être paramétrée, globalement, ou à leur création (auquel cas des trains seront plus rapides que d'autres).
- Les trains sont amphibies et peuvent rouler sur l'eau (en ligne droite), mais bien entendu pas dans les forêts.
- En cas d'obstacle de type forêt, eau ou bord du plateau, les trains n'explosent pas, mais se retournent et repartent dans la direction opposée.
- Présence de personnes aléatoires sur les voies : lorsqu'un train heurte une personne, il explose et la personne disparaît aussi. Prévoir en option un bouton spécial pour l'internaute permettant d'arrêter tous les trains avant de heurter ces personnes.
- Présence d'un monstre qui se déplace de façon aléatoire sur le plateau, et qui peut percuter les trains, occasionnant leur explosion. En option : le monstre ne peut pas se déplacer sur l'eau. En option : au lieu d'une marche aléatoire, le monstre va chercher à se rapprocher du train le plus proche de lui.
- Montrer graphiquement l'explosion des trains, plutôt que de les faire disparaître instantanément (par exemple avec la lecture d'un gif animé, qui va ensuite disparaître). Un son peut aussi être lu, ou bien un message d'erreur apparaître sur la page.
- En cas d'explosion des trains, au lieu de le faire disparaître, une « épave de train » (c'est-à-dire une série de ruines) va apparaître. Cette épave constituera un obstacle infranchissable pour les autres trains. Les épaves ne disparaissent qu'après avoir été remplacées manuellement par un autre type de case.
- Prévoir des cases « aiguillages ». Ajouts supplémentaires : un système de signaux ferroviaires pour aiguiller les trains dans la bonne direction; ces signaux pourront être actionnés manuellement par l'internaute, notamment pour éviter des collisions.
- Ajouter des TGV, qui avancent deux fois plus rapidement que les autres. En option : un TGV est capable de freiner de lui-même pour ne pas percuter le train devant.
- Ajouter des RER B, qui s'arrêtent aléatoirement, c'est-à-dire à chaque unité de temps, avancent ou n'avancent pas.
- Ajouter des gares, dans lesquelles les trains vont s'arrêter une unité de temps.
- Ajouter des gares de triage : lorsqu'un train passe par une gare de triage, un wagon lui est ajouté.
- Un monde (configuration initiale du plateau) peut être choisi au début du jeu parmi une liste prédéfinie de mondes. Ne pas hésiter à varier les thèmes (train du désert, train du pôle nord, train sur Mars...).
- Le monde courant peut être exporté vers un fichier texte, et ensuite réimporté.

