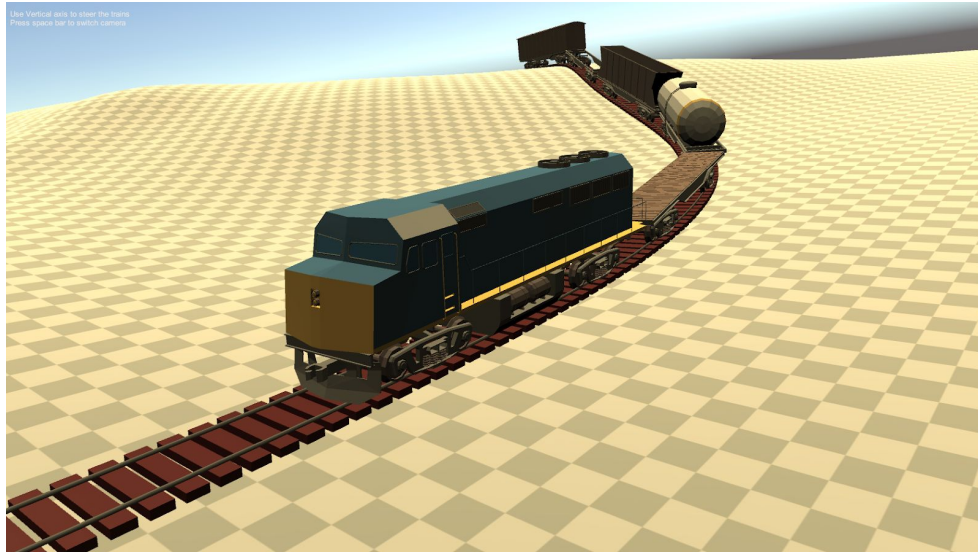


TP: Système de déplacement d'un train sur voie ferrée sous Unity3D



Le 16 juin 2020 de 9h à 17h.

Le projet est à réaliser en binôme.

Les objectifs du TP sont les suivants:

- Découvrir les courbes spline définies à partir de courbes de Bézier cubiques,
- Optimiser un algorithme de recherche sur une courbe spline,
- Développer dans Unity3D un système de déplacement d'un train sur voie ferrée:
 - Intégrer un package de génération et gestion de voie ferrée intitulé *SimpleRailwaySystem*.
 - Modéliser géométriquement le comportement en position et orientation d'un train sur des rails
 - Intégrer et configurer des assets graphiques du commerce représentant des locomotives et des wagons (origine: Asset Store)
 - Développer deux scripts: *TrainUnit.cs* et *Train.cs*, qui définissent respectivement le comportement d'un élément roulant (locomotive ou wagon) et d'un train complet constitué de plusieurs éléments roulants
- Utiliser les boîtes à outils Mathf, Vector3 et Quaternion

Un build est disponible qui rend compte des attentes.

L'archive du projet Unity3D, débarrassé du répertoire Library, doit être livrée avant 17h. Le nom de l'archive doit suivre la nomenclature suivante:

M1_2019_2020_TP_Unity3D_Train_<NomNom>.zip

L'évaluation de ce TP prendra en compte la qualité de vos réflexions, et donc la pertinence de vos questions.

Ordre conseillé des apprentissages et développements:

- Lancer le build et analyser le comportement cinématique attendu
- Créer un nouveau projet Unity3D (version supérieure ou égale à 2019.3.6f1)
- Importer les packages *SimpleRailwaySystem* et *LowPolyTrains*
- Lire la documentation du *SimpleRailwaySystem* et bien comprendre son fonctionnement. Attention, le code n'est quasiment pas du tout commenté, il va vous falloir faire un peu de *reverse engineering*.
- Lire de la documentation sur les courbes de Bézier, qui sont la base des courbes splines sur lesquelles s'appuie le *SimpleRailwaySystem*:
 - https://en.wikipedia.org/wiki/B%C3%A9zier_curve
 - a priori vous ne devriez pas avoir à développer, pour ce TP, quoi que ce soit en lien direct avec ces courbes
- **OBLIGATOIRE: Optimiser la méthode *GetPositionNormalTangent* du script *MySpline.cs*.** (rappelez-vous le cours sur l'interpolation linéaire de courbes 2D et les optimisations apportées à l'algorithme de recherche d'un point sur la courbe)
- Se documenter sur le comportement cinématique d'un véhicule équipé de deux bogies et se déplaçant sur voie ferrée:
 - <https://fr.wikipedia.org/wiki/Bogie>
- Réfléchir à la hiérarchie d'un GameObject représentant un tel véhicule comprenant:
 - Deux bogies
 - Deux points d'attache du véhicule, un à l'avant, un à l'arrière
 - Une voiture (la cabine habitable du véhicule)
- Importer le package *LowPolyTrains* et l'analyser. Les éléments graphiques du package peuvent-ils être utilisés pour atteindre les objectifs du TP ?
- Créer et configurer des prefabs des différents véhicules roulants que vous comptez utiliser (*a minima* une locomotive et un wagon)
- Développer les deux scripts *TrainUnit.cs* et *Train.cs*, en respectant quelques contraintes
 - Un bogie est toujours aligné selon les rails
 - La distance entre les deux bogies d'un même véhicule est constante
 - La voiture du véhicule doit être alignée selon l'axe formé par les deux bogies, et son orientation autour de cet axe doit "suivre" l'orientation moyenne calculée à partir de l'orientation des bogies.
 - De plus, la voiture d'un véhicule doit osciller légèrement pour ajouter du dynamisme
 - Les roues du train tournent à une vitesse de rotation proportionnelle à sa vitesse de translation et ne glissent pas par rapport aux rails
 - Vous élaborerez vos propres règles pour définir la position relative d'un véhicule par rapport au suivant. Dans le build qui vous a été fourni, la règle choisie, si elle permet de conserver une distance relativement constante entre deux véhicules,

ne permet pas d'assurer une connexion mécanique optimale entre les véhicules. Dans la vraie vie, le système est complexe car il autorise une certaine liberté de mouvement relatif entre deux véhicules connectés, ainsi qu'un amortissement mécanique des chocs. Vouloir implémenter informatiquement un tel système réel serait possible mais dépasse les objectifs de ce TP. En gros vous n'avez pas le temps de trouver une solution idéale, mais un bon compromis entre réalisme et complexité du système peut être trouvé. Soyez imaginatifs.