

Projet Trajectoires

1 Présentation générale du projet

Le projet consiste à modéliser en XML, puis en C#, un ensemble de particules bougeant de manière aléatoire dans un environnement de simulation, et à enregistrer leurs trajectoires. La simulation sera initiée grâce à un nombre de particules et une définition des caractéristiques de l'environnement. Elle générera les particules. Lors de son arrêt elle enregistrera d'une part les trajectoires individuelles des particules dans `Trajectories.xml`, mais également les positions des particules dans `Particles.xml`. La simulation de particules pourra éventuellement être affichée de façon dynamique sous la forme d'une application MonoGame. De plus, les enregistrements des trajectoires feront l'objet d'une transformation XSLT vers une image SVG qui permettra leur affichage dans un navigateur (HTML).

Ce travail vous donnera l'occasion d'intégrer vos connaissances en XML, XML Schema, XSLT avec l'usage de C#, de ses parsers et de la sérialisation.

2 Réalisation

2.1 Modélisation XML Schema

Vous allez devoir modéliser plusieurs types dont la plupart des instances XML vous sont fournies. Les schémas XML devront valider ces instances sans modification.

Il est recommandé de passer par une étape de modélisation UMC (plant UML).

Les types à modéliser sont :

- Coordinate
- BoundingBox
- Environnement
- Particle
- Particles
- Trajectory
- Trajectories
- Simulation

Notez qu'un `Environnement` est une `BoundingBox`, au même titre qu'une `Particle`. Bien entendu, `Particles` contient une liste de `Particle`, `Trajectories` contient une liste de `Trajectory`, une `Simulation` contient un `Environnement`

2.2 Feuilles de transformation XSLT

Ecrire une feuille de transformation qui génère une figure au format SVG ([Documentation SVG](#)) comme suit. La figure enregistrée dans un fichier `trajectoires.svg` contient :

- un rectangle coloré qui représente l'environnement
- un ensemble de lignes (`polyline`) qui représentent chacune une trajectoire.

Il est recommandé d'écrire un SVG test à la main et de l'afficher en HTML, comme cela a été fait en cours.

Ecrire une deuxième feuille qui génère directement une page HTML qui affiche cette figure.

2.3 C# - Validation de schéma et feuilles XSLT

Ecrivez une classe C# nommée `Simulation` qui valide les schemas de la simulation, des particules et des trajectoires.

Ecrivez dans cette même classe une méthode qui applique sur les données de trajectoires la feuille de transformation XSLT générant du SVG.

Idem avec une méthode générant le HTML correspondant.

2.4 C# - Sérialisation

Ecrivez les classes suivantes de telle manière à ce qu'elles soient capable de sérialiser les types XML correspondants.

- `BoundingBox`
- `Environnement`
- `Particle`
- `Particles`
- `Trajectory`
- `Trajectories`
- `Simulation`

Le type `Coordinate` n'a pas besoin d'être codé en C# car il est substitué par le type existant `Vector2`.