

MAR TP

Les systèmes à bases de particules

A l'adresse suivante : <http://www.irisa.fr/mimetic/Fabrice.Lamarche/MAR/2015>, vous trouverez le fichier `particles.zip` contenant le matériel pour la gestion des systèmes à base de particules. Téléchargez ce fichier et décompressez le dans le répertoire racine de votre TP.

Dans le répertoire `assets`, un sous répertoire `particles` doit apparaître. Ce répertoire contient un ensemble de textures que vous pourrez utiliser durant le TP pour associer des représentations graphiques à vos particules.

Trois fichiers sont ajoutés dans votre répertoire racine :

- `MathExt.js` : un fichier contenant une classe gérant des intervalles
- `Interpolator.js` : un fichier contenant des interpolateurs linéaires qui pourront être utilisés pour configurer le système de particules.
- `ParticleSystem.js` : un fichier contenant le gestionnaire de systèmes à bases de particules.

Le système à base de particules est géré par la classe *`ParticleSystem.Engine_Class`*.

Pour la connexion au graphe de scène de `three.js`, cette classe possède un attribut *`particleSystem`* qui doit être ajouté à votre graphe de scène afin d'effectuer le rendu du système à base de particules.

Cette classe met à votre disposition une méthode *`animate(dt)`*, permettant d'animer le système. Il met aussi à votre disposition une méthode *`addEmitter`* permettant d'ajouter un émetteur de particules. Il est à noter que la seule une classe d'émetteur de particules vous est fournie : *`ParticleSystem.ConeEmitter_Class`*. Enfin, la méthode *`addModifier`* vous permet d'ajouter des éléments modifiant les caractéristiques des particules afin de gérer l'animation. C'est à l'aide de ces derniers que vous pouvez créer votre propre système. Les modificateurs fournis sont les suivants :

- *`ParticleSystem.LifeTimeModifier_Class`* : met à jour la durée de vie d'une la particule. Il faut obligatoirement ajouter ce modificateur si vous voulez un systèmes fonctionnel gérant la vie et la mort des particules.
- *`ParticleSystem.PositionModifier_EulerIntegration_Class`* : change la position d'une particule en fonction de sa vitesse et des forces qui lui sont appliquées.
- *`ParticleSystem.PositionModifer_PlaneLimit_Class`* : empêche les particules de traverser un plan.
- *`ParticleSystem.PositionModifer_PlaneBounce_Class`* : fait rebondir les particules sur un plan.
- *`ParticleSystem.ForceModifier_ResetForce_Class`* : remet à zéro les forces appliquées sur les particules. Si vous utiliser la physique pour gérer l'animation des particules, il faut absolument ajouter ce modificateur avant d'ajouter des modificateurs de force. Cet modificateur est automatiquement ajouté lors de la création d'un systèmes à base de particules, vous n'avez donc pas à vous en soucier.
- *`ParticleSystem.ForceModifier_Attractor_Class`* : un attracteur ponctuel.
- *`ParticleSystem.ForceModifier_Repeller_Class`* : un répulseur ponctuel.

- `ParticleSystem.ForceModifier_AttractorSegment_Class` : un attracteur ayant la forme d'un segment.
- `ParticleSystem.ForceModifier_RepellerSegment_Class` : un répulseur ayant la forme d'un segment
- `ParticleSystem.ForceModifier_Weight_Class` : ajoute une force correspondant au poids de la particule.
- `ParticleSystem.OpacityModifier_TimeToDeath_Class` : modification de l'opacité de la particule en fonction d'un interpolateur paramétré par le ratio entre le temps de vie de la particule et sa durée de vie.
- `ParticleSystem.SizeModifier_TimeToDeath_Class` : modification de la taille de la particule en fonction d'un interpolateur paramétré par le ratio entre le temps de vie de la particule et sa durée de vie.
- `ParticleSystem.SizeModifier_TimeToDeathFactor_Class` : applique un facteur d'échelle à la particule en fonction d'un interpolateur paramétré par le ratio entre le temps de vie de la particule et sa durée de vie.
- `ParticleSystem.ColorModifier_TimeToDeath_Class` : modifie linéairement la couleur de la particule en fonction du le ratio entre le temps de vie de la particule et sa durée de vie.

Question 1 : Créez un système à base de particules ayant les caractéristiques suivantes :

- Nombre maximum de particules : 10000
- Fichier de texture : `assets/particules.particle.png`
- Mode de blending : `THREE.AdditiveBlending`

Question 2 : Ajoutez lui un émetteur de forme conique avec émettant 1000 particules à la seconde et émettant ces dernières vers le haut. Les particules émises devront avoir les caractéristiques suivantes

- Vitesse dans l'intervalle [5 ;10]
- Masse dans l'intervalle [0.1 ;0.3]
- Taille dans l'intervalle [0.1 ;1.0]
- Durée de vie dans l'intervalle [1.0 ;7.0]

Question 3 : Ajoutez les modificateurs suivants (dans l'ordre indiqué) :

- `ParticleSystem.LifeTimeModifier_Class`
- `ParticleSystem.PositionModifier_EulerIntegration_Class`

Observez le résultat : les particules partent en hauteur et ne retombent jamais.

Question 4 : Ajoutez une instance de la classe `ParticleSystem.ForceModifier_Weight_Class` juste avant le modificateur réalisant l'intégration d'euler. Vous pourrez alors observer que les particules retombent, car elles sont soumises à leur propre poids.

Question 5 : Ajoutez un modificateur de transparence afin que la particule soit opaque à sa naissance et transparente lorsqu'elle meurt.

Question 6 : Ajouter un modificateur de manière à ce que la particule vire au rouge foncé à l'approche de sa mort.

Question 7 : Accrochez la représentation graphique de votre système à base de particules à un axe en rotation. Que pouvez-vous observer ?

Question 8 : Afin de remédier au problème de la question 7, créez une copie de la classe d'émetteur en cône et ajoutez une fonctionnalité permettant de faire dépendre la position et l'orientation de l'émetteur d'une transformation associée à un nœud du graphe de scène. Remplacez l'émetteur de votre système de particules par cet émetteur. Refaites la question 7 en remplaçant la représentation graphique de votre système à base de particules à la racine du graphe de scène, tout en « attachant » votre émetteur à l'extrémité de l'axe en rotation.

Question 9 : Complétez le prototype de votre hélicoptère avec le système de particules de manière à ce que des particules soient émises à l'arrière des turbines latérales. Vous modifierez aussi les caractéristiques du système de manière à améliorer le rendu graphique.