

# Gestion de scene pour le moteur 2D

Pierre Turpin

6 juin 2013

## Table des matières

<b>1</b>	<b>Position</b>	<b>3</b>
1.1	Informations . . . . .	3
1.2	Actions . . . . .	3
1.3	Logiciel . . . . .	3
<b>2</b>	<b>Node</b>	<b>3</b>
2.1	Informations . . . . .	4
2.2	Actions . . . . .	4
2.3	Logiciel . . . . .	4
<b>3</b>	<b>Node de caméra</b>	<b>4</b>
3.1	Informations . . . . .	5
3.2	Actions . . . . .	5
3.3	Logiciel . . . . .	5
<b>4</b>	<b>Node de fenetre de vision</b>	<b>5</b>
4.1	Informations . . . . .	5
4.2	Actions . . . . .	5
4.3	Logiciel . . . . .	5
<b>5</b>	<b>Node de lumière</b>	<b>6</b>
5.1	Informations . . . . .	6
5.2	Actions . . . . .	7
5.3	Logiciel . . . . .	7
<b>6</b>	<b>Node de mur de lumière</b>	<b>7</b>
6.1	Informations . . . . .	7
6.2	Actions . . . . .	7
6.3	Logiciel . . . . .	7

<b>7</b>	<b>Node graphique</b>	<b>8</b>
7.1	Informations . . . . .	8
7.2	Actions . . . . .	8
7.3	Logiciel . . . . .	8
<b>8</b>	<b>Node physique</b>	<b>8</b>
8.1	Informations . . . . .	8
8.2	Actions . . . . .	9
8.3	Logiciel . . . . .	9

# 1 Position

Il y a trois coordonnées de positionnement. Les deux coordonnées (x et y) sont sur le plan du sol et sont à valeur discrète et représente la taille d'un pixel (si aucune projection n'est faite). La coordonnée de profondeur est à valeur discrète également mais ne sert qu'à l'affichage pour savoir dans quelle ordre afficher les éléments.

## 1.1 Informations

**Entier** x

**Entier** y

**Entier positif** depth<sup>1</sup>

## 1.2 Actions

**Modifier** x

**Modifier** y

**Modifier** depth

## 1.3 Logiciel

Vide

# 2 Node

Un node est un élément de base de la scène.

Il contient ses propres informations géométrique telles que sa position, sa rotation et son échelle. On peut alors le transformer grâce aux transformations de base : translation<sup>2</sup>, rotation<sup>3</sup> et mise à l'échelle<sup>4</sup>.

Un node peut posséder des enfants nodes. Les enfants de node auront alors en plus de leur propre transformation locale, la transformation du parent. La relation parent-enfant est bidirectionnel donc un node possède également un parent. Le parent d'un node peut ne pas être défini. Cela signifie que ce node est la racine de la scène.

Les nodes seront utilisés par le moteur graphique et le moteur physique en même temps. C'est donc une ressource critique. Un node possède donc également un mutex de protection.

---

1. 0 correspond au niveau du sol

2. La translation sur x et y est libre mais la translation sur z n'influe pas sur la réelle altitude du node, seulement sur son ordre d'affichage.

3. seul la rotation sur l'axe Z de la profondeur est gérée pour rester dans un espace 2D

4. la mise à l'échelle sur l'axe Z n'est pas possible pour la même raison que la rotation

## 2.1 Informations

**Transformation** transformation locale

**Ensemble de Node** liste des enfants

**Node** parent

**Mutex** protection

## 2.2 Actions

**Appliquer** une transformation  $t$

**Appliquer** une rotation d'angle  $a$  (en radian)

**Appliquer** une translation de vecteur  $v$

**Appliquer** une mise à l'échelle sur  $x$  de  $s_x$  et sur  $y$  de  $s_y$

**Récupérer** la liste des enfants

**Modifier** le parent

**Supprimer** le parent

**Protéger** les accès multithread du node

**Libérer** les accès multithread du node

## 2.3 Logiciel

Vide

## 3 Node de caméra

Correspond à un node représentant une caméra. Une caméra est une fenetre de visualisation de la scène. Elle contient les meme informations que le node<sup>5</sup>.

La caméra a un point d'attache ainsi qu'une focale. Elle n'a pas de point de vision car elle est toujours orienté sur l'axe vertical  $Z$ . Entre les  $Z+$  et les  $Z-$ , la caméra est orienté vers les  $Z-$ <sup>6</sup>. La mise a l'échelle de la caméra est ignoré par celle-ci (aucun changement sur l'aspect graphique et encore moins sur l'aspect physique).

L'affichage se fait à travers les caméras. Les entités visibles<sup>7</sup> depuis la caméra sont projeté sur elle puis envoyé au moteur de rendu.

---

5. Par héritage dans la programmation

6. mieux vaut voir le sol que d'être tete en l'air

7. Détection par culling grâce à un quadtree et le graphe de scène

### 3.1 Informations

**Position** point d'attache<sup>8</sup>

**Scalaire** focale

### 3.2 Actions

**Modifier** le point d'attache

**Modifier** la focale

### 3.3 Logiciel

Vide

## 4 Node de fenetre de vision

C'est un node qui va de pair avec la caméra puisqu'il correspond à un node d'écran. C'est à dire que si il y a des caméras pour filmer, il faut également des écrans pour afficher le résultat d'une caméra.

Ce node contient donc une référence vers la caméra auquel il est branché ainsi qu'une taille. Le jeu peut donc se voir avec une fenetre ayant comme caméra, une qui a une focale de 1.0. Une minimap correspondra alors à une autre fenetre de visualisation ayant comme caméra, une centré sur toute la carte (dans le cas d'un STR) et ayant une distance focale suffisamment petite pour pouvoir voir toute la carte.

Le node de fenetre contient également toutes les informations et les actions du node de base.

### 4.1 Informations

**Node de camera** caméra

**Taille** dimension de la fenetre

### 4.2 Actions

**Modifier** la caméra

**Modifier** la dimension

### 4.3 Logiciel

Vide

---

8. C'est la meme chose que la position du node reçu par héritage

## 5 Node de lumière

Ce type de node a toute les caractéristiques qu'à le node de base.

Ce node correspond à la lumière dans la scène. Il y a en tout 4 types de lumières différentes<sup>9</sup>.

Le premier type est la lumière ambiante. Elle agit sur toute la scène, qu'il y ait des mur ou non.

Le second type est la lumière unidirectionnelle. Les rayons lumineux sont tous parallèles et vont tous dans la même direction. Cela permet de représenter les rayons solaires.

Le troisième type est la lumière en spot. Elle envoie des rayons lumineux dans une forme de cône.

Le dernier type est la lumière omnidirectionnelle comme une lampe normale. Elle envoie les rayons lumineux tout autour d'elle.

Le moteur est en deux dimensions, pourtant pour les lumières, il aurait été bien d'intégrer une troisième dimension afin d'avoir des effets de profondeur avec la lumière. Il aurait alors été possible de pouvoir dire que le Z dans la position de la lumière représente vraiment l'altitude et que le Z des autres objets est 0<sup>10</sup>. Cependant, cela rendrait mal dans certains cas d'utilisation. Par exemple, si il y a un fossé dans la scène avec une source lumineuse qui n'est pas totalement au dessus du fossé, alors normalement il y aurait de l'ombre en fond. Mais il aurait du y avoir une prise en compte de l'altitude du sol et du fossé. C'est possible de simuler ça en plaçant des murs pour la lumière ayant une certaine hauteur mais cela reviendrait au même que créer un décor en 3D. La lumière sera donc mise au niveau du sol et son impact ne variera pas selon Z. Seul le plan d'affichage aura une incidence de l'intensité décroissante de la lumière.

Une lumière a une intensité ainsi qu'un rayon d'action. A l'emplacement de la source, l'intensité de la lumière est au maximum et décroît progressivement jusqu'à atteindre le rayon d'action. La loi de décroissance de l'intensité est une simple loi proportionnelle<sup>11</sup>.

Dans le cas d'un soleil, le rayon d'action peut être défini avec un nombre très grand devant la taille de la scène. De cette façon, la décroissance de l'intensité n'est pas visible.

La lumière possède également une teinte pour pouvoir simuler différents types de lumière (feu, lampe, couché de soleil, UV, ...).

### 5.1 Informations

**Scalaire** intensité

**Longueur** rayon d'action

---

9. Cela pourra se traduire par de l'héritage au niveau du code

10. Puisque tous les nodes sont plaqués sur le sol pour l'affichage

11. Il faudra effectuer des tests pour vérifier si il n'y a pas l'intégration d'autres lois

**Couleur** teinte

## 5.2 Actions

**Modifier** l'intensité

**Modifier** le rayon d'action

**Modifier** la teinte

## 5.3 Logiciel

Vide

# 6 Node de mur de lumière

Ce type de node a toute les caractéristiques qu'à le node de base.

Avec l'intégration de la lumière, il faut également intégrer des mur pour éviter que la lumière puisse passer partout. Les autres nodes n'ont aucune sémantique d'obstacle à la lumière. Et meme si il y a des nodes pour la collision physique, cela n'a pas de rapport avec les obstacles pour la lumière.

Les murs de lumières sont représenté par une ligne et ont une hauteur infini toujours en accord avec le fait que le moteur n'ai que 2 dimensions. La position du mur représente donc le centre du mur et la longueur du mur est représenté par sa mise à l'échelle sur X. La taille du mur sur Y est toujours 0. La rotation locale du mur permet de faire correspondre de X local du mur avec le Y global.

Le mur possède également un taux de filtre. Si ce taux est à 1.0, alors toute la lumière est bloqué. Si il est à 0.0, alors toute la lumière passe. Si il est à 0.5, seulement la moitié passe.

## 6.1 Informations

**Taux** filtre

## 6.2 Actions

**Modifier** le taux de filtre

## 6.3 Logiciel

Vide

## 7 Node graphique

Ce type de node a toute les caractéristiques qu'à le node de base.

Ce node correspond a un élément graphique dessinable.

Il possède une texture matricielle. Cette texture est chargé grâce à un gestionnaire de texture permettant d'éviter les doublons. L'affichage se fait par couche de profondeur. Cela permet de placer les éléments le plus éloigné d'abord permettant ensuite de placer les éléments au premier plan qui vont recouvrir les autres. Une autre approche serait d'utiliser un masque d'affichage avec un rendu dans l'autre sens : du plus proche au plus éloigné. Seul les pixels n'ayant pas été modifié se plaque sur l'écran et si le masque est totalement plein, alors le rendu est terminé. Meme si il reste des éléments derrière, c'est sur qu'ils ne seront pas affichés<sup>12</sup>.

### 7.1 Informations

**Texture** texture matricielle

### 7.2 Actions

**Modifier** la texture

### 7.3 Logiciel

Vide

## 8 Node physique

Ce type de node a toute les caractéristiques qu'à le node de base.

Ce node correspond à un élément physique avec une masse, une inertie, une vitesse, une accélération. Celui-ci est représenté par un carré à la base. Cependant plusieurs nodes peuvent fusionner pour former un élément physique plus complexe. Ce node correspond a un élément graphique dessinable.

### 8.1 Informations

**Masse** masse

**Distance / Temps** vitesse

**Distance / Temps<sup>2</sup>** acceleration

---

12. Test à faire pour vérifier qu'elle technique est plus rentable



## 8.2 Actions

**Modifier** la masse<sup>13</sup>

**Modifier** la vitesse

**Modifier** l'accélération

**Appliquer** une force

**Appliquer** une fusion entre deux nodes physique

## 8.3 Logiciel

Vide

---

13. a voir si on peut modifier dynamiquement la masse où si elle est constante