

Fichiers scénarios : description d'une scène 3D

Ce document décrit le format de fichiers scénarios utilisés pour ce projet. Nous utiliserons l'extension `.ray` pour ces fichiers. Le répertoire `scenarios` du projet contient quelques exemples de tels fichiers. Le fichier `src/scenario.mli` contient une représentation OCaml (ou *AST*) d'un scénario, et nous vous fournissons également les lexers/parseurs correspondants.

1 Une scène 3D

Une scène 3D est composée des éléments suivants:

- un ensemble d'objets géométriques (plans, sphères et cubes) placés dans l'espace tridimensionnel. Chaque objet possède une *texture*. Cette texture est composée de:
 1. une couleur de surface, qui est le triplet des valeurs pour les trois couleurs primaires rouge, vert et bleu, chaque valeur étant comprise entre 0 et 1 ;
 2. un coefficient de réflexion diffuse (entre 0 et 1) ;
 3. un coefficient de réflexion spéculaire (entre 0 et 1) ;
 4. un coefficient de Phong (un réel strictement positif).

Le rôle de ces trois coefficients est expliqué dans le document présentant la mathématique du lancer de rayon.

Noter que la surface d'un objet a la même texture partout.

- l'intensité de l'illumination ambiante (valeur entre 0 et 1).
- un ensemble de sources lumineuses, chacune avec une direction dans l'espace et une intensité (valeur entre 0 et 1).
- la distance de la caméra à l'origine.
- l'angle horizontal du champ de vision.

2 Le langage de description

Un *scénario* est un fichier de description d'une scène. Plutôt que de lister directement les éléments qui composent une scène, on va chercher à *programmer* cette scène, à l'aide en particulier d'une notion de procédure, afin de pouvoir fabriquer des scènes complexes de manière concise et évolutive.

Un fichier scénario est composé de trois parties :

1. les propriétés globales du scénario ;
2. une suite de définitions de procédures ;
3. une suite d'instructions pour placer des objets et sources lumineuses.

Tout texte entre le caractère `#` et la fin de ligne est un commentaire.

Propriétés globales. Cette partie contient successivement:

- la spécification de la caméra : entre les mots-clés **camera** et **end**, on doit trouver **distance** et une expression pour la distance de la caméra à l'origine, puis **angle** et une expression pour l'angle horizontal du champ de vue.
- le mot clef **ambient**, suivie d'une expression pour l'intensité d'illumination ambiante
- et enfin une liste (éventuellement vide) de *sources lumineuses*, décrites chacune par **light vector** r_x, r_y, r_z **intensity** i **end** où r_x, r_y et r_z sont des expressions décrivant une rotation devant être appliquée à la direction de la source de lumière (par défaut située à la verticale en haut), et i une expression qui dénote l'intensité par une valeur entre 0 et 1.

Définitions de procédure. Une définition de procédure commence par le mot-clé **proc**, puis le nom de la procédure, puis les paramètres formels entre parenthèses et séparés par des virgules, puis une liste d'instructions, et se termine sur le mot-clé **end**.

Instructions. Les différentes instructions sont:

- l'affectation numérique **let** *identificateur* = *expression* ;
- l'affectation géométrique **object** *identificateur* = *object-expression* ;
- l'appel d'une procédure, donné par le nom de la procédure suivi des arguments entre parenthèses et séparés par des virgules ;
- le placement d'un objet sous la forme **put** *object-expression* ;
- une instruction conditionnelle **if** ... **then** ... **else** ... **end**, ou une conditionnelle incomplète **if** ... **then** ... **end**.

Les arguments d'une procédure sont forcément des expressions numériques, pas des objets. Par contre, une procédure peut utiliser toute variable globale existant lors de son appel, qu'il s'agisse d'une variable numérique (créée par **let**) ou géométrique (créée par **object**). Au cours d'un appel à une procédure, l'effet d'une affectation reste local: une procédure ne peut pas modifier la valeur d'une variable globale (mais une variable locale peut cacher une variable globale).

Expressions. Les expressions dénotent des valeurs flottantes. Elles sont bâties sur des identificateurs, des constantes et des opérateurs. On utilise les opérateurs arithmétiques habituels et les fonctions trigonométriques *sin* et *cos*. L'identificateur particulier **pi** est directement traduit en la constante $\pi = 3.1415...$ lors du parsing des scenarios. Le point flottant des constantes numériques n'est pas obligatoire, car toutes les constantes dénotent des valeurs flottantes.

Objets. Ils sont bâties à partir des variables d'objets, des créations d'objets de bases (plans, sphères, boîtes), et des transformations appliquées à d'autres objets. On peut également grouper plusieurs objets via **group** ... **end** pour les manipuler ensuite comme une seule entité.

- Un *plan* est décrit par **plane rotation** r_x, r_y, r_z **shift** d t **end** où les angles r_x, r_y, r_z forment une rotation, d est une distance à l'origine, et t une texture. Le plan considéré initialement est le plan horizontal (xOz), que l'on oriente vers le haut en choisissant (Oy)

comme vecteur normal. Puis on applique à ce plan (xOz) la rotation décrite par les angles r_x, r_y, r_z , ce qui nous donne un nouveau plan P de vecteur normal \vec{n} . Enfin on translate ce plan de $d \cdot \vec{n}$. Ce décalage d peut être négatif.

- La *sphère* de centre c , de rayon r et de texture t est décrite par **sphere center c_x, c_y, c_z radius r t end.**
- La *boîte* de centre c , de dimensions l_x, l_y, l_z selon chacun des axes, et de texture t est décrite par **box center c_x, c_y, c_z length l_x, l_y, l_z t end.** Par rapport à la caméra placée sur l'axe (Oz), l_x est donc la largeur de la boîte, l_y sa hauteur, et l_z sa profondeur.

Et pour les transformations d'objets:

- Une *translation* de l'objet o par le vecteur v est décrite par **translate o by v_x, v_y, v_z end.**
- Une rotation de l'objet o par les angles r_x, r_y, r_z est décrite par **rotate o by r_x, r_y, r_z end.**
- Une *dilatation* ou homothétie de rapport k de l'objet o est décrite par: **scale o by k end.** Ces dilatations sont toujours centrées sur l'origine o .

Textures. Une texture est donnée par **kd e_d ks e_s phong e_p color e_r, e_v, e_b** où $e_d, e_s, e_p, e_r, e_v, e_b$ sont des expressions donnant les coefficients de réflexion diffuse, de réflexion spéculaire, l'exposant de Phong, et les intensités de rouge, vert et bleu pour la couleur. Les intensités des couleurs prennent des valeurs entre 0 et 255 (attention, l'échelle change entre la représentation interne et le langage de description).

Variable temporelle. Notez qu'une variable particulière nommée **time** peut être utilisée dans un scénario sans être définie au préalable, voir par exemple **scenarios/planets.ray**.

- Pour une image simple, **time** vaudra 0;
- Pour un film d'animation à n images, **time** contiendra le numéro de l'image en cours de rendu (entre 1 et n). Cela permettra de faire évoluer la scène au fur et à mesure des images.