

Travaux pratiques d'infographie

POV-Ray

1. Introduction

Cette année, les étudiants de Licence 3 STAMN ont dû réaliser dans le cadre du TP d'infographie, une scène sur Povray avec comme thème « atelier ». Ce projet devait être réalisé en binôme sur une période d'un mois, et la consigne était accompagnée de contraintes techniques, les voici :

- Réalisation de quatre outils différents, dont un constitué d'un **assemblage de matériaux** et un autre **d'un seul matériau** ;
- D'éléments géométriques de type **lathe** et **prism** avec un profil de type **bezier_spline**, ainsi que **sphere_sweep** avec un profil de type **cubic_spline**, et **text** ;
- D'un **fragment transparent** ;
- De **deux sources lumineuses**, un standard et une de type spotlight ;
- D'un léger **flou focal** sur la caméra.

2. Note d'intention

Réalisé par Téo L'Huillier et Sarah Mauriaucourt, « **Brique a Brac** » met en scène un atelier de bricolage.

Lorsqu'on entend le mot bricolage, intuitivement, on l'associe à l'utilisation d'outillage physique, à la quincaillerie, mais aussi aux actions qu'elle permet de réaliser. En effet, bricoler, c'est arranger, réparer, fabriquer quelque chose dans plusieurs secteurs d'activité.

C'est ainsi que nous avons décidé de réaliser un aménagement intérieur d'un atelier de bricolage. Pour répondre à la demande, nous avons réfléchi aux différents éléments que nous pourrions modéliser. Notre choix, c'est porté sur la réalisation de :

- **Burin** : C'est un outil qui peut se créer avec **un seul fragment de type « lathe » avec un profil « bezier_spline »** ;
- **Marteau** : C'est un outil qui se crée avec un **assemblage de fragments** et sa tête peut se faire avec un **fragment de type « prism » avec un profil « bezier_spline »** ;
- **Clef plate** : C'est un outil qui se crée avec un **assemblage de fragments** ;
- **Tournevis** : C'est un outil qui se crée avec un **assemblage de fragments**, son manche peut se faire avec un **fragment de type « lathe » avec un profil bezier_spline** et peut intégrer de la **couleur transparente** ;
- **Fil** : C'est un objet qui se crée avec un **fragment de type « sphere_sweep » avec un profil « cubic_spline »** ;
- **Horloge** : C'est un objet qui se crée avec un **assemblage de fragments** et qui peut intégrer des **éléments textuels**.

Ensuite, nous avons réfléchi à des éléments pour agrémenter notre scène : multiprise, cendrier, cigarette, rame de papier, livres, clou, rondelle, écrou, vis.

2020-2021

Après s'être mis d'accord sur les éléments que devaient intégrer prioritairement la scène, nous avons réfléchi à leur agencement dans la mise en scène. Nous voulions insérer des éléments électroniques à l'aspect rétro et avoir de la profondeur dans la scène. Voici le croquis que nous avons fait et qui a été notre image de référence :



Et comment, on l'a fait évoluer, en fonction du premier rendu :



3. Détails techniques

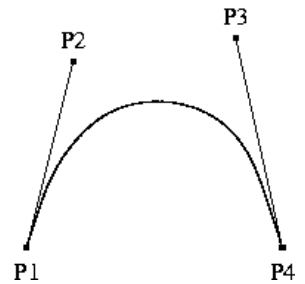
Après avoir réfléchi au concept et au design de la scène, nous avons débuté la conception de celle-ci.

a. Modélisation des formes et des matériaux des objets

Pour la modéliser des différents objets, nous avons pris des mesures réelles pour les avoir à l'échelle humaine. Nous avons débuté la conception de nos éléments par les objets et fragments imposés dans le sujet.

2020-2021

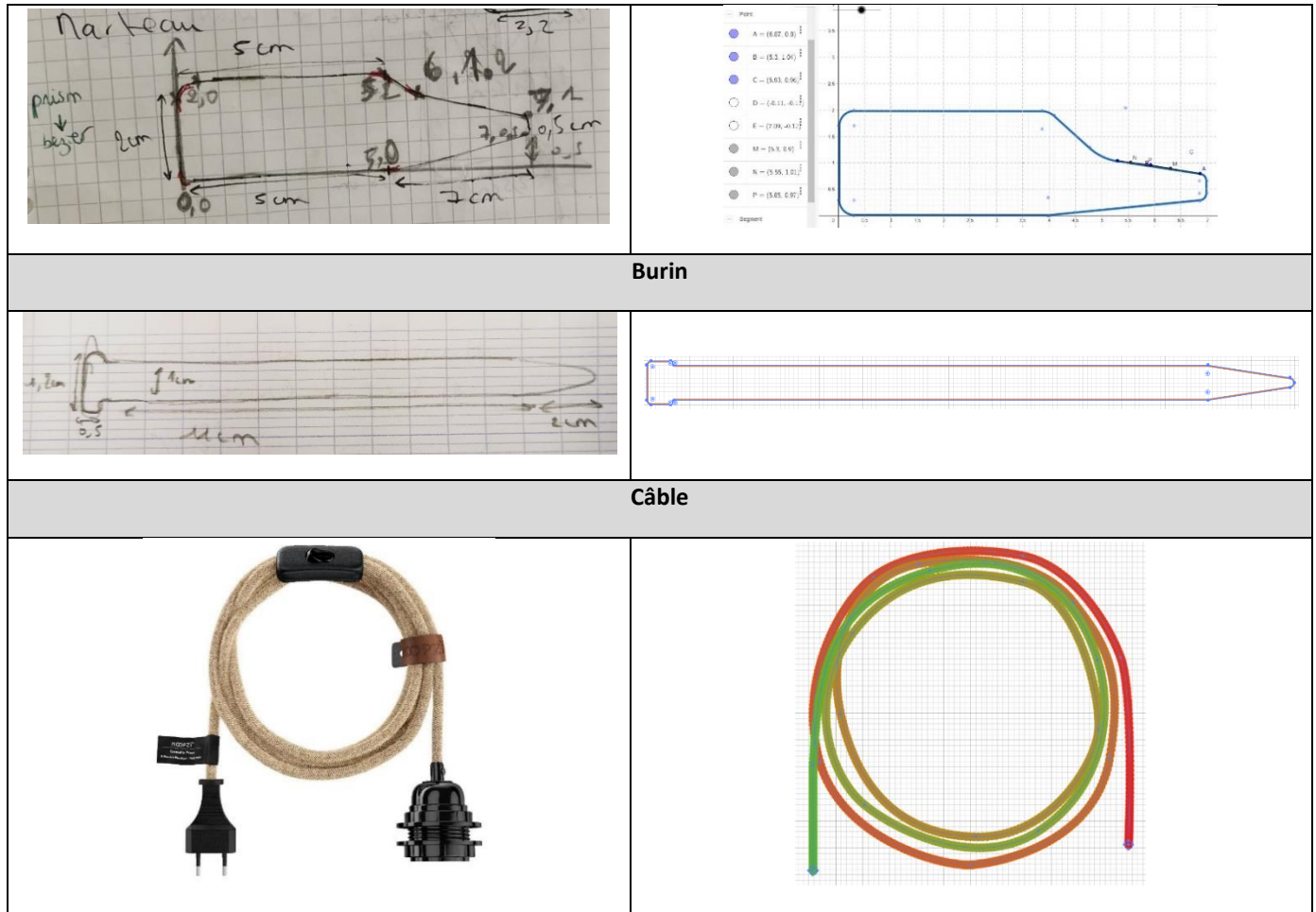
Ainsi, pour la modélisation géométrique, nous devons réaliser 2 fragments composés d'un profil de type « bezier_spline ». Cependant, lors de nos premiers essais, nous avons eu du mal à comprendre comment régissait ce profil. Puis, nous avons compris qu'une courbe de Bézier est paramétrée par 4 points de contrôle. Avec, le premier et le dernier qui permettent de donner une information sur la longueur de déplacement, tandis que les deux autres points donnent une information sur la direction de la courbe. Or, même avec ces informations, nous avons eu de la difficulté à positionner les points donnant l'information de direction entre les points de début et de fin.



Pour nous aider, nous avons eu l'idée d'utiliser GeoGebra. En effet, après avoir réalisé le croquis à l'échelle de nos éléments, nous les avons vectorisés sur Illustrator en gardant la même échelle, pour ensuite les importer sur GeoGebra. Ainsi, sur GeoGebra, nous avons utilisé un script de courbe de Bézier à 4 points de contrôle que nous avons positionné sur notre vectorisation, ce qui a permis de récupérer les coordonnées des points donnant l'information de direction. Nous avons aussi utilisé cette méthode pour le fragment « sphere_sweep » au profil « cubic_spline ».

Croquis	Vectorisation
Tournevis	
Tête de marteau	

2020-2021



Grâce à GeoGebra, nous avons récupéré les coordonnées de nos différents points et nous avons pu donner les formes de courbes à utiliser pour nos fragments de type « prism » et « lathe ». Voici ce que ça donne pour la **tête de marteau**, le **burin**, le **manche de tournevis** et le **câble** :

```
// Tête : Marteau
#declare TeteMarteau = prism
{
  bezier_spline
  linear_sweep
  0, 2, 4*13,
  <0,.4>, <0,.24>, <.13,0>, <.32,0>,
  <.32,0>, <2,0>, <3.65,0>, <4,0>,
  <4,0>, <4,0>, <6.87,.28>, <6.87,.28>,
  <6.87,.28>, <6.94,.28>, <7,.37>, <7,.43>,
  <7,.43>, <7,.43>, <7,.66>, <7,.66>,
  <7,.66>, <7,.72>, <6.95,.8>, <6.87,.8>,
  <6.87,.8>, <6.3,.9>, <5.55,1.01>, <5.3,1.04>,
  <5.3,1.04>, <5.17,1.06>, <4.9,1.17>, <4.78,1.3>,
  <4.78,1.3>, <4.78,1.3>, <4.1,1.9>, <4.1,1.9>,
  <4.1,1.9>, <4.06,1.94>, <3.94,2>, <3.85,2>,
  <3.85,2>, <3.85,2>, <3,2>, <3,2>,
  <3,2>, <.19,1.99>, <.02,1.87>, <0,1.7>,
  <0,1.7>, <0,1.7>, <0,.4>, <0,.4>
}
```

```
// Burin
#declare Burin = prism
{
  bezier_spline
  linear_sweep
  0, 2, 4*15,
  <0,.1>, <0,.04>, <.06,0>, <.1,0>,
  <.1,0>, <.1,0>, <.55,0>, <.55,0>,
  <.55,0>, <.58,0>, <.6,.03>, <.6,.05>,
  <.6,.05>, <.6,.1>, <.62,.1>, <.65,.1>,
  <.65,.1>, <.65,.1>, <.13,.1>, <.13,.1>,
  <.13,.1>, <.13,.1>, <.14,9,.4>, <.14,9,.4>,
  <.14,9,.4>, <.14,96,.41>, <.15,.47>, <.15,.5>,
  <.15,.5>, <.15,.54>, <.14,96,.6>, <.14,9,.6>,
  <.14,9,.6>, <.14,9,.6>, <.13,.9>, <.13,.9>,
  <.13,.9>, <.13,.9>, <.65,.9>, <.65,.9>,
  <.65,.9>, <.62,.9>, <.6,.94>, <.6,.96>,
  <.6,.96>, <.6,.98>, <.58,1>, <.55,1>,
  <.55,1>, <.55,1>, <.1,1>, <.1,1>,
  <.1,1>, <.06,1>, <.0,.96>, <.0,.9>,
  <.0,.9>, <.0,.9>, <.0,.1>, <.0,.1>
}
```

```
// Manche : Tournevis
#declare TournevisManche = lathe
{
    bezier_spline
    4*6,
    <0,0>, <.36,0>, <.75,.17>, <.85,.29>,
    <.85,.29>, <1.17,1.18>, <1.33, 3.46>, <.85,4.89>,
    <.85,4.89>, <.7,5.83>, <.91,6.78>, <1.11,7>,
    <1.11,7>, <1.11,7>, <1.11,7.29>, <1.11,7.29>,
    <1.11,7.29>, <.85,7.25>, <.63,7.39>, <.6,7.5>,
    <.6,7.5>, <.6,7.5>, <.2,7.5>, <.2,7.5>
}

// Câble/Fil
#declare Fil = sphere_sweep
{
    cubic_spline
    22,
    <0, .2, 2>, .2
    <0, 1, 2>, .2
    <0, 5, 1.9>, .2
    <5, 13, 1.8>, .2
    <12.2, 9.1, 1.7>, .2
    <9.7, 1.8, 1.6>, .2
    <2.2, 2.9, 1.5>, .2
    <1.7, 10.2, 1.4>, .2
    <9, 12.4, 1.3>, .2
    <12.2, 6.1, 1.2>, .2
    <7, 1.5, 1.1>, .2
    <1.1, 6.8, 1.1>, .2
    <4.5, 13.3, 1.2>, .2
    <11.2, 11.6, 1.3>, .2
    <12.8, 4.8, 1.4>, .2
    <6.8, 0.3, 1.5>, .2
    <0.2, 4.7, 1.6>, .2
    <2.5, 12.7, 1.7>, .2
    <9, 13.6, 1.8>, .2
    <13.2, 9.3, 1.9>, .2
    <13.5, 1.2, 2>, .2
    <13.5, .2, 2>, .2
}
```

Cela, nous a permis de comprendre qu'un fragment « lathe » s'obtenait en faisant tourner son profil autour de l'axe y, ce qui permet de faire une symétrie axiale. Alors que pour le « prism », il était nécessaire de fermer la forme créée, en ayant le premier et dernier point semblable.

Toutes les formes créaient, on était fait au niveau de l'origine, afin d'optimiser la recherche des éléments lors de l'assemblage.

b. Aspects des objets

Pour avoir un rendu photoréaliste, nous avons appliqué des textures aux différentes formes et matériaux. Afin que les textures soient conformes à la réalité, nous avons utilisé la librairie qu'offre POV-Ray.

Pour un effet de liège, on a utilisé un motif de bois « wood » composé d'un « color_map » de deux couleurs.

```
#declare Liege = material
{
    texture
    {
        pigment {
            wood turbulence 0.2 rotate <-80,-20,0> scale <1,8,1>*0.07 // : Echelle des pigments donnent l'effet de grain
            color_map { [ 0 rgb <149,115,45>/255*1.1 ] [ 1 rgb <149,115,45>/255*0.6 ] } // Couleur du pigment "wood" : 2 couleurs marrons claires pour l'effet de liège
        }
        finish { ambient 1 diffuse 0.6 specular 0.4 roughness 0.01 }
    }
}
```

Pour le sol de l'atelier, nous avons mis un effet de bois de chêne sur le bureau et le sol, grâce à l'importation d'une image 2D. Et grâce à la propriété « reflection », on a adapté ce que réfléchit l'objet et grâce à la propriété « ior », on a réglé l'indice de réfraction. On a fait en sorte que la texture soit légèrement vernie pour rajouter du réalisme lorsque la lumière est reflétée.

```
#declare Chene = material
{
    texture {
        pigment { image_map { jpeg "/Ressources/img/chene.jpg" } }
        normal { bump_map { jpeg "/Ressources/img/chene.jpg" } bump_size 0.1 }
        finish { reflection { 0, 0.7 fresnel on } ambient .3 diffuse 1 }
        rotate x*60
        scale <1/2,1,1>*40
    }
    interior { ior 1.03 }
}
```


2020-2021

Pour un effet de fumée, nous avons utilisé le paramètre « density », afin de faire varier la densité dans l'espace du corps qui aura cette propriété. Ainsi, cette densité définie a une forme sphérique et sa couleur intérieure est définie par un « density_map » où on a attribué 4 types de nuances de gris. Puis, on a ajouté un paramètre de « turbulence » pour avoir un aspect nuageux.

```
#declare MonoxydeDeCarbone = material
{
  texture { pigment { rgbt 1 } }
  interior
  {
    media
    {
      absorption 5 // Opacité de la fumée
      density
      {
        spherical density_map
        {
          [0 rgb 0] [0.5 rgb 0] [0.7 rgb .5] [1 rgb 1]
        }
        scale 1/2 warp { turbulence 0.5 } scale 2
      }
    }
  }
  scale <1.5,6,1.5> translate y
}
```

Pour l'effet de papiers empilés, nous avons utilisé un « color_map », ou nous avons assignés une nuance de gris à différents points de l'espace allant de 0 à 1. À cette texture, est associée une « macro », car nous utilisons deux fois cette texture, mais orientée par un axe différent.

```
#macro EmpilementFeuilles(Axe)
material
{
  texture
  {
    pigment {
      gradient Axe
      color_map {
        [0.0 colour White ]
        [0.1 colour Gray90 ]
        [0.2 colour White ]
        [0.3 colour Gray95 ]
        [0.4 colour White ]
        [0.5 colour Gray80 ]
        [0.6 colour White ]
        [0.6 colour Gray95 ]
        [0.8 colour White ]
        [0.9 colour Gray90 ]
        [1.0 colour Gray85 ]
      }
    }
  }
  finish { ambient 0.6 diffuse 0.7 }
  scale <0, -3, 0>
}
#end
#declare EmpilementFeuillesY = EmpilementFeuilles(y)
#declare EmpilementFeuillesX = EmpilementFeuilles(z)
```

Nous avons réutilisé cette texture pour les pages des livres. En effet, au départ, pour créer la texture des livres, nous voulions essayer d'insérer une image sur des fragments de type « box » et « polygon », or, lorsque nous faisons ça, l'image ne s'adaptait au ratio des fragments. Alors, nous avons essayé avec un paramètre « uv_mapping » qui consiste à projeter une image 2D sur la surface d'un modèle 3D « box » correspondant à son patron. Or, le rendu n'était pas très réalisé, c'est pourquoi nous avons décidé de ne pas projeter d'image de livres sur une « box ». Voici le résultat du code ci-dessous :

```
// Test : Texture livre
box {
  <0,0,0>,<30, 18, 15>
  texture
  {
    uv_mapping
    pigment
    {
      image_map
      {
        png "..\index\Ressources\img\tranches_livres.png"
        map_type 0
        once
      }
      translate<0,1.11,0> scale <.5, .3, 0>
    }
    finish { phong 1 } finish { ambient 1 diffuse 0 }
  }
  rotate <0, 60, 0>
}
```



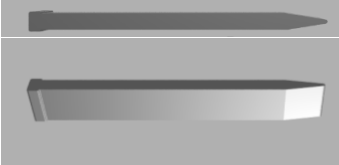
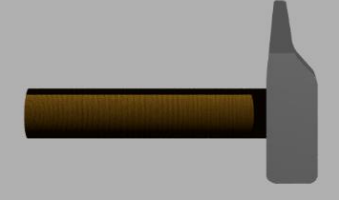
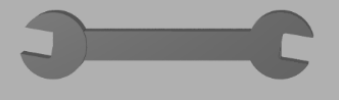
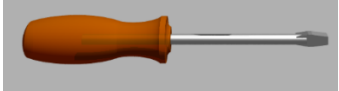
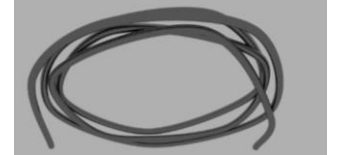


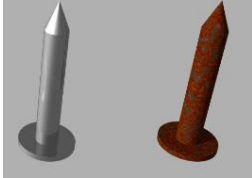

Pour les autres textures, nous avons utilisé des éléments plus simples, comme l'utilisation des textures ou de couleurs de la librairie POV-Ray.

c. Assemblages

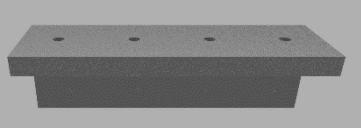
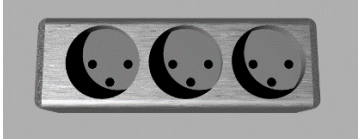

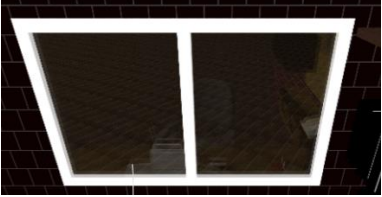
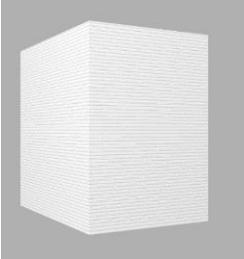

Après avoir créé les différents fragments de chaque objet et créé les textures, il a fallu les assembler.

Rendu de l'objet	Réalisation des objets
------------------	------------------------

2020-2021

	<p>Le burin est composé d'un seul fragment de type « prism » au profil « bezier_spline ».</p>
	<p>Le marteau est composé d'un assemblage de fragments. Pour sa tête, nous avons créé un fragment de type « prism » au profil « bezier_spline », puis un « cylindre » avec une échelle aplatie pour le manche. Ensuite, nous avons uni les deux matériaux.</p>
	<p>La clé plate est composée d'un rectangle pour le manche uni à deux cylindres auxquels on a fait une différence avec des « Round_Box » pour les fourches. Puis l'angle entre les fourches et le manche est généralement de 15°.</p>
	<p>Le tournevis est composé d'un fragment de type « lathe » au profil « bezier_spline », d'un « cylindre » pour son corps et d'un fragment de type « prism » au profil « bezier_spline » pour sa tête. Tous ces fragments ont été uni.</p>
	<p>Le câble est composé d'un seul fragment de type « sphere_sweep » au profil « cubic_spline ».</p>
	<p>L'écrou est composé d'un « prism » au profil « linear_spline » auquel on a fait une différence avec une « sphere ».</p>
	<p>La rondelle est composée d'un seul fragment de type « disc ».</p>
	<p>Les clous sont un assemblage de deux « cylindres », l'un pour la tête et l'autre pour le corps, ainsi que d'un « cône » pour la pointe. On a fait deux textures, l'une à l'aspect rouillé et l'autre à l'aspect métallisé.</p>
	<p>L'élaboration de l'horloge a nécessité trois étapes principales. Pour le contour de l'horloge, nous avons fait une différence entre un « torus » et un « cylindre » que nous avons uni à un autre « cylindre » pour avoir le socle de celle-ci. Ensuite, nous avons créé le verre de l'horloge grâce à une « sphere ». Puis, pour les aiguilles, nous avons uni trois « cylindres » et pour les heures, nous avons créé quatre textes grâce à un fragment « text » qui utilise la « macro ».</p> <p>Pour le fragment « text », nous avons importé la typographie « crystal.ttf »</p>

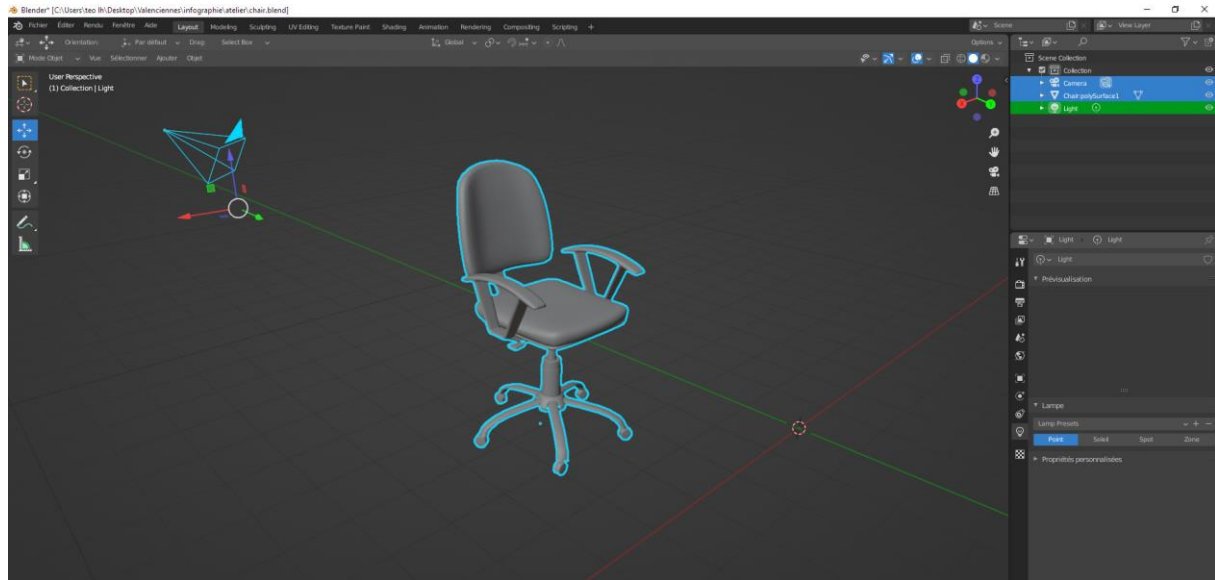
2020-2021

	<p>L'élaboration du rangement de tournevis a nécessité trois étapes principales. Pour le support de rangement, nous avons fait une différence de deux « box ». Puis, nous avons créé un « cylindre » que nous avons répété dans une boucle, pour qu'il s'affiche à un pas donné. Puis nous avons fait une différence du support de rangement avec les cylindres.</p>
	<p>L'élaboration de la multiprise murale a nécessité trois étapes principales. On a créé une boucle « for » pour créer une prise type E. Ainsi, la prise est constituée d'un « cylindre », auquel on a fait une différence avec trois autres « cylindres ». Puis, on a fait une différence avec le support des prises qui est une « Round_Box » avec la prise type E.</p>
	<p>Le cendrier est composé d'une différence d'un « cône », d'une « sphère » et de deux « cylindres ».</p> <p>La fumée de tabac est composée d'une « sphère » aplatie et creuse grâce à « hollow on ».</p> <p>La cigarette est composée d'une union de trois « cylindres ». Les cylindres blanc et orange ont la même épaisseur, pour avoir les pigments des deux couleurs qui se superposent.</p>
	<p>La fenêtre est composée d'une « box » principale pour l'encadrement de celle-ci, auquel on fait une différence avec quatre « box » pour les vantaux. Ensuite, on a affiché les vantaux pour avoir la texture de verre minérale. Et dans le mur, on a fait une différence avec l'encadrement de la fenêtre, pour voir ce qui se passe à l'extérieur.</p>
	<p>La rame de papier est composée d'un seul fragment de type « box » avec une texture qui donne l'impression que des papiers sont superposés.</p>
	<p>Les livres ont nécessité trois étapes principales.</p> <p>Nous avons créé la couverture du livre avec deux « box » pour le premier et la quatrième de couverture, ainsi qu'un « cylindre » avec une échelle aplatie pour la tranche. Puis, précédemment, nous avons créé des « macros » pour la rame de papier et pour l'effet de superposition de feuilles, nous avons récupérés ces codes pour faire les pages adaptés au livre. Ainsi, nous avons uni le tout pour créer un seul livre.</p> <p>Ensuite, on a créé une boucle « for » pour créer plusieurs livres avec un intervalle de pas entre chaque livre. Nous avons aussi inséré une variable aléatoire pour avoir un rangement moins strict.</p> <p>Puis, à l'intérieur de cette boucle, nous en avons créé une deuxième, pour mettre une couleur aléatoire sur chaque livre.</p>

2020-2021

Nous avons aussi réalisé l'assemblage d'un miroir, il est basé sur des procédés simples que nous avons utilisés précédemment (différence de deux « box »).


d. Export d'objet Blender en POV-Ray

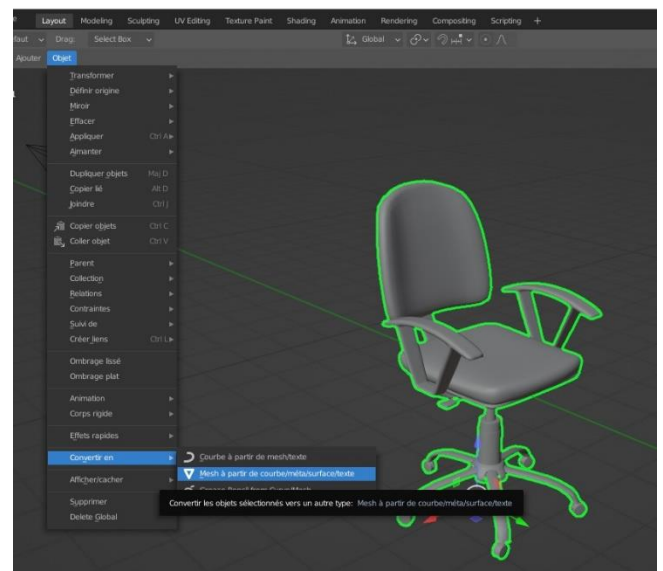


Pour enrichir visuellement notre scène et la peupler de davantage de contenus, nous avons également implémenté des objets importés. Ceci avait pour but notamment de faciliter et d'accélérer l'ajout d'appareils électroniques très souvent complexes à modéliser. Voici une explication du processus de conversion entre les formats compatibles blender et le code POV.

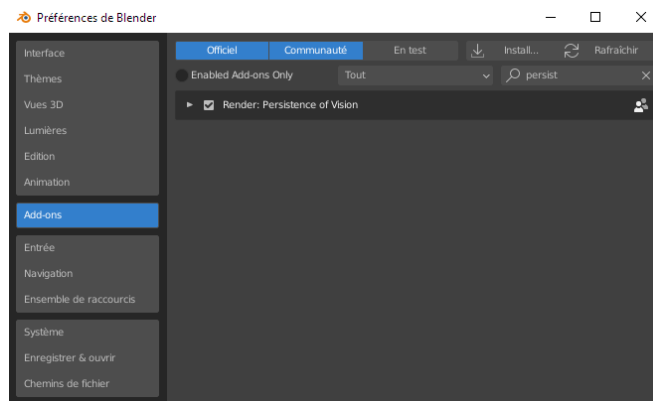
Tout d'abord, il est possible d'intégrer des objets venant d'Internet dans Blender, avec différents formats d'objets 3D (fbx, obj, stl, 3ds) qui possèdent chacun leur particularité (textures ou brut, polygonaux, etc.) ou bien des scènes blender (.blend). Prenons l'exemple de la chaise de bureau.

Lorsque l'on importe cette chaise, dans une scène basique comprenant une caméra et l'objet en question. Afin de maximiser la compatibilité avec POV-Ray, il est très fortement recommandé de convertir en « mesh » donc en filet de points de coordonnées formant un solide (et qui se traduiront par des matrices dans le code final POV-Ray). Ceci est symbolisé

par cette icône triangulaire :  Chair:polySurface1. Pour convertir en « mesh », il faut sélectionner l'ensemble de l'objet, se rendre dans l'onglet objet, puis convertir en « mesh ».



2020-2021

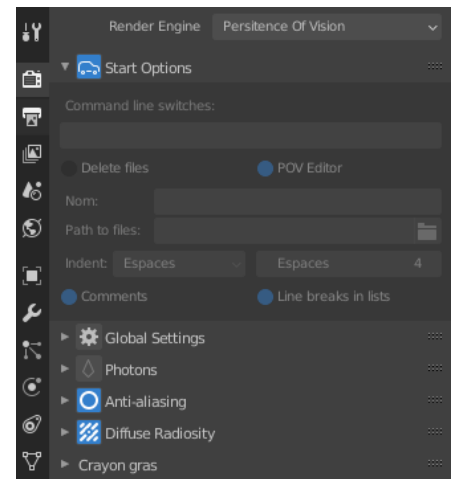


À présent, la scène étant optimisée pour POV-Ray, elle est prête à être exportée. Nous aurons besoin du greffon externe « Persistence of Vision » inclus dans toutes les versions récentes de Blender (les nôtres étant 2.7 et 2.8). Pour l'activer, il faut se rendre dans l'onglet préférences, présent dans « éditer », puis dans « add-ons ». On peut rapidement trouver le greffon grâce à une recherche textuelle et cocher la case : « Render : Persistence of Vision ».

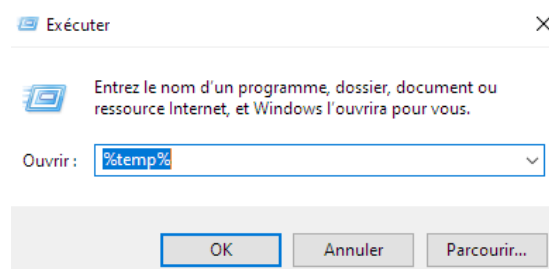
Quittons maintenant les préférences et sélectionnons le moteur de rendu POV-Ray. Pour cela, il y a une petite icône en forme de télévision dans la partie droite de l'interface de Blender, et un petit menu déroulant.

La configuration requise est :

- Le moteur « Persistence of Vision » ;
- La case « Delete files » décochée ;
- La case POV « editor » cochée pour conserver les fichiers temporaires de rendu en mémoire de la machine.



Pour lancer le rendu, on appuie sur la touche « F12 », POV-Ray se lance automatiquement. Il faut conserver chaque fenêtre ouverte et se rendre dans les fichiers temporaires du système.



En appuyant sur « OK », une fenêtre de l'explorateur de fichiers s'ouvre. Pour trouver le fichier.pov rapidement, il est conseillé de trier le dossier par date la plus récente.

AppData > Local > Temp			
Rechercher dans : Temp			
Nom	Modifié le	Type	Taille
alltext.out	25/01/2021 15:14	Fichier OUT	5 Ko
tmpw2fkk04.png	25/01/2021 15:14	Fichier PNG	175 Ko
tmp0o39n5ol.ini	25/01/2021 15:14	Paramètres de co...	1 Ko
tmpvyee8gn4.pov	25/01/2021 15:14	POV-Ray scene so...	2 456 Ko
~DFAA7D4CE81DDB2526.TMP	25/01/2021 15:14	Fichier TMP	1 Ko
chair_20964_autosave.blend	25/01/2021 15:12	Blender File	3 567 Ko

En double-cliquant, on peut ouvrir ce fichier sous le logiciel POV-Ray. Les premières lignes concernent la caméra et l'éclairage, si l'on souhaite l'intégrer dans une autre scène, on peut supprimer cette partie pour alléger le code. Sous cette partie de code, il y a beaucoup de lignes générées automatiquement par le logiciel Blender sous forme de matrices, de coordonnées d'axes et de points. Pour simplifier à la lecture du fichier final, nous nous sommes servi des unions. Chaque objet est pointé par son union dans le fichier annexe blender.pov sous la forme :

2020-2021

```
#declare chaise = union{ Le code de chaque objet suivant cette technique }
```

Le fichier blender.pov est inclus dans le fichier principal, on pourra donc réutiliser ces unions pour les afficher et leur faire subir des translations, rotations et mises à l'échelle.

```
#include "../Ressources/blender/blender.pov"
```

e. Mise en scène

i. Lumières

Au niveau des lumières, nous en avons mis une de type « standard » qui génère la lumière de la pièce, puis une autre de type « Spotlight » qui simule une lumière extérieure. En effet, cette lumière extérieure orangée, passe par les vitres de la fenêtre, afin de représenter un lampadaire.

ii. Caméra

Nous avons positionné notre caméra proche du plan de travail pour avoir une vue en légère contre-plongée. Notre objectif était d'avoir des objets en amorce (ici à gauche), afin de faire un flou dessus. Ainsi, nous avons réglé l'ouverture de la caméra et la quantité de flou à mettre sur la localisation du flou focal. Ensuite, nous avons simulé la profondeur de champ focale, pour finir sur la précision de calcul de chaque pixel. Nous avons mis une petite précision de calcul, afin d'avoir un meilleur rendu.

4. Rendu de la scène finale



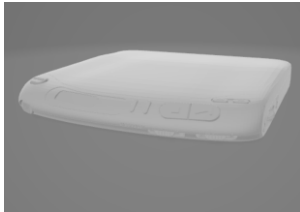
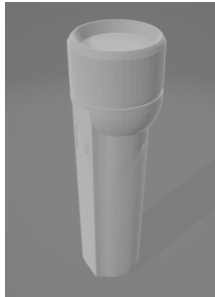
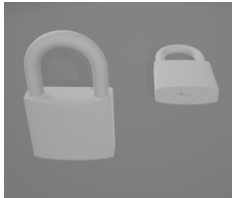

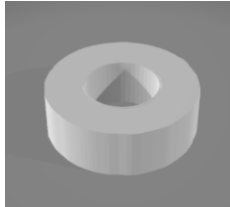
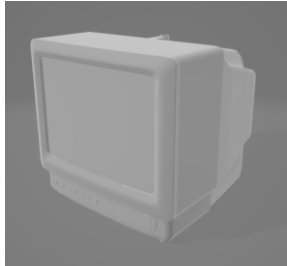
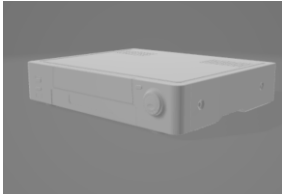

Au niveau des statistiques de travail, le temps du rendu est de 2 minutes 19. Cela est dû à l'importation des objets blender et au flou focal. Ensuite, nous avons passé une quarantaine d'heures chacun sur ce projet. Voici le résultat final de notre projet « **Brique a Brac** » :



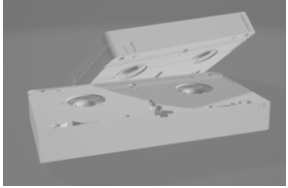




2020-2021

5. Webographie

Certains accessoires dans le décor sont extérieurs à nous. C'est-à-dire que nous avons récupéré des éléments Blender que nous avons importé en POV-Ray. Afin de ne pas mélanger les éléments que nous avons créés et ceux que nous avons récupérés, nous avons créé un fichier « blender.pov » contenant toutes les sources autres que les nôtres. Puis, nous avons appelé les différents objets dans notre scène. Voici les objets que nous avons récupérés et leurs sources de téléchargement :

Marteau 	Téléphone fixe 	Lecteur de disque 	Lampe torche 
Cadenas 	Pince 	Rouleau de scotch 	Poste de télévision 
	Lecteur de cassette 	Clef 	
Format initial : 3DS, mis à l'échelle, ajout des textures et exporté sur Blender. Source : https://www.wiley.com/legacy/compbooks/fleming/models.htm			

 <p>Chaise de bureau</p> <p>Source : https://free3d.com/3d-model/chair-16205.html</p> <p>Format initial : OBJ, mis à l'échelle et exporté sur Blender</p>	 <p>Radio</p> <p>Source : https://free3d.com/3d-model/radio-51766.html</p> <p>Format initial : 3DS, mis à l'échelle et exporté sur Blender</p>	 <p>K7</p>
<p>Bureau et tiroirs</p> <p>Source : https://free3d.com/3d-model/wood-desk-93009.html</p> <p>Format initial : OBJ et changement de texture sur POV-Ray</p> <p>Nous l'avons modifié sur Blender, ceci comprend :</p> <ul style="list-style-type: none"> • Ecartement des blocs de soutien • Ajout des tiroirs latéraux <div style="display: flex; justify-content: space-around;">   </div>		

Pour la texture du bureau et du sol, nous avons récupéré une texture de parquet

- **Source :** <https://www.pierreetparquet.fr/produits/parquet/parquet-contrecolle/che%CC%82ne-havane-fume-incolore-contrecolle/>

