

Rapport Web3D – Maison 3D

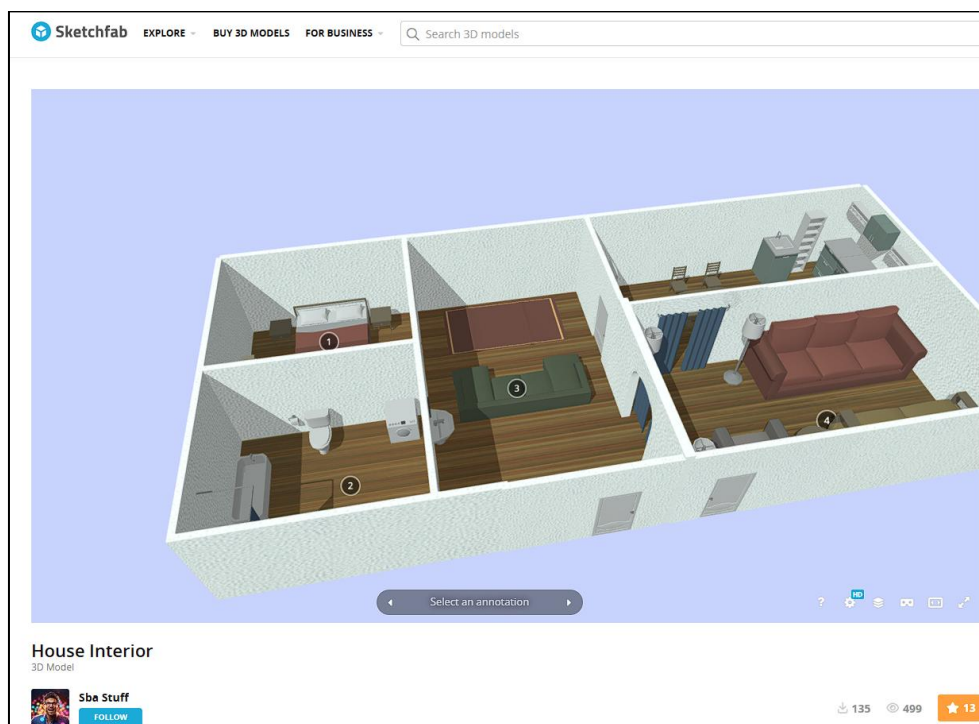
Pierre HE - Aurélien AH-SANE

Nous devons modéliser une maison où nous pouvons naviguer à partir d'un plan, intégrant un mode marche, et ainsi que des interactions avec des objets.

Le modèle 3D.....	1
La conversion X3D	2
Le Plan.....	2
La Navigation	3
Le mode Marche	4
Interaction avec les objets	5

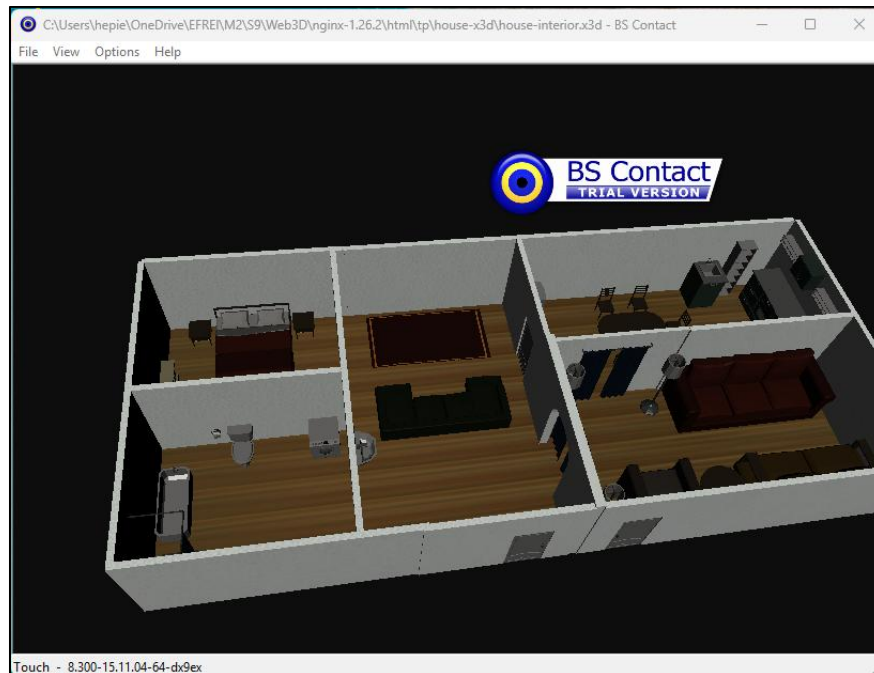
Le modèle 3D

Bien qu'il soit possible de le modéliser par soi-même, le temps alloué nous ont contraint à trouver un modèle en ligne sur Sketchfab: *House Interior* par Sba Stuff. Ce modèle convenait à nos besoins par sa simplicité et sa gratuité. Les meubles étaient aussi issus d'assets génériques, évitant d'éventuels problèmes de licence.



La conversion X3D

Sketchfab permet de télécharger le modèle sous différents formats possibles, cependant *House Interior* ne pouvait être obtenu sous forme .X3D. Nous avons dû passer par Blender avec son Add-on “Web3D X3D/VRML2 Format” qui permet l’export en X3D. On s’assure ensuite que le modèle puisse être exploité en utilisant BSContact.



Le Plan

Pour obtenir un plan rapidement, nous avons utilisé le *Model Inspector* de Sketchfab : activation d’un Wireframe noire avec les couleurs de base pour éviter les shaders intégrés :

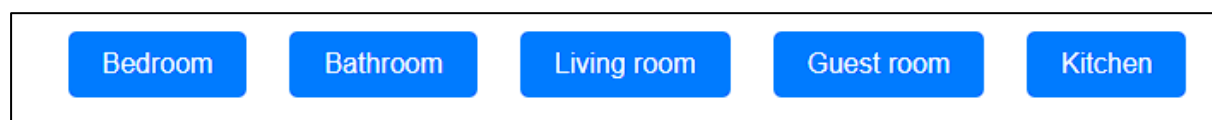


La disposition suit :

1. Chambre
2. Salle de bains
3. Salon
4. Salle de Réception
5. Cuisine

La Navigation

Pour des raisons de simplicité, nous avons simplement créé des boutons assignés à chaque salle à côté du plan. Bien évidemment, il aura été possible d'intégrer les boutons sous forme de zone cliquable directement sur le plan.



Ces boutons utilisent des **Viewports**, déplaçant la caméra dans un endroit précis de chaque pièce, permettant à l'utilisateur de naviguer dans la maison.

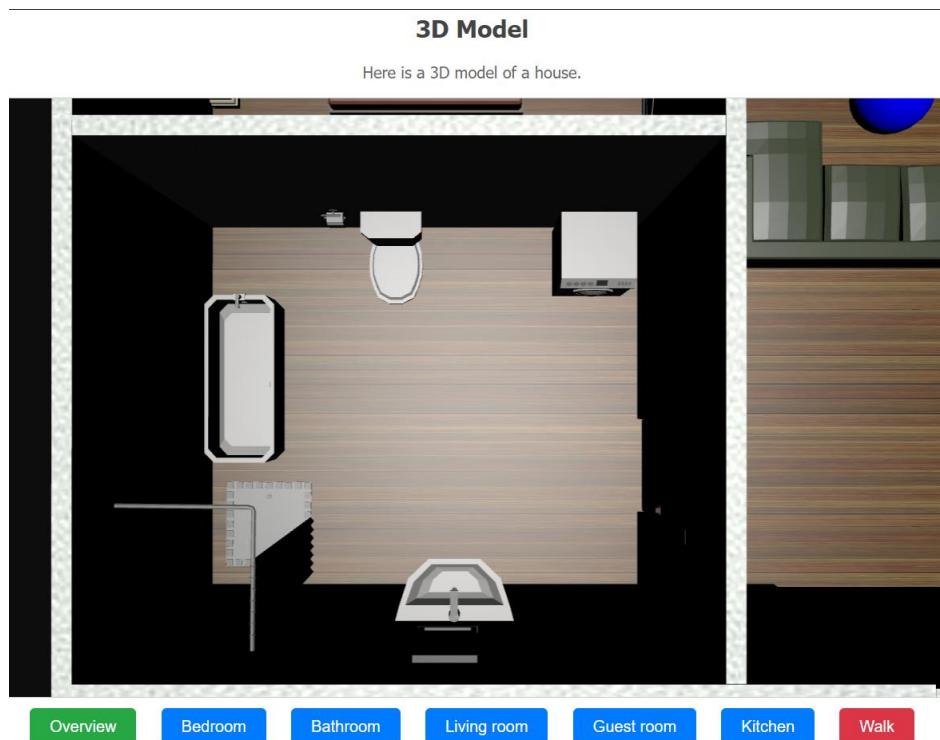
```
<div class="container">
  <h2>3D Model</h2>
  <p>Here is a 3D model of a house.</p>

  <x3d width='100%' height='600px'>
    <scene>
      <!-- Inline inclusion of the XML file -->
      <inline url='./house/house-interior.x3d'></inline>

      <!-- Viewpoints -->
      <viewport DEF="house-bedroom" position="-500 1000 -200" orientation="1 0 0 -1.57" description="Bedroom"></viewport>
      <viewport DEF="house-bathroom" position="-500 1000 300" orientation="1 0 0 -1.57" description="Bathroom"></viewport>
      <viewport DEF="house-livingroom" position="0 1000 0" orientation="1 0 0 -1.57" description="Living room"></viewport>
      <viewport DEF="house-guestroom" position="1000 1000 300" orientation="1 0 0 -1.57" description="Guest room"></viewport>
      <viewport DEF="house-kitchen" position="1000 1000 -200" orientation="1 0 0 -1.57" description="Kitchen"></viewport>
    </scene>
  </x3d>

  <div class="nav-button">
    <!-- Button for navigation -->
    <!-- Bedroom -->
    <button onclick="navigateTo('house-bedroom')">Bedroom</button>
    <!-- Bathroom -->
    <button onclick="navigateTo('house-bathroom')">Bathroom</button>
    <!-- Living room -->
    <button onclick="navigateTo('house-livingroom')">Living room</button>
    <!-- Guest room -->
    <button onclick="navigateTo('house-guestroom')">Guest room</button>
    <!-- Kitchen -->
    <button onclick="navigateTo('house-kitchen')">Kitchen</button>
  </div>
```

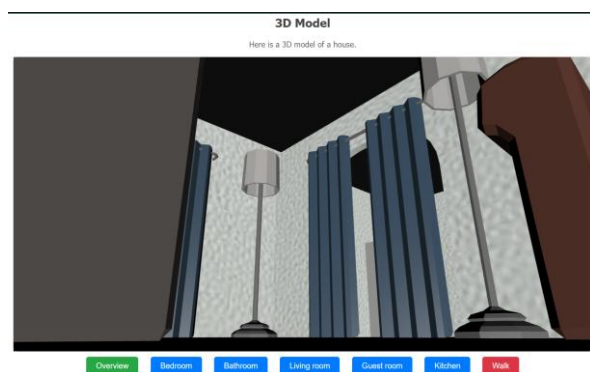
A cause de la conversion en X3D, le fichier associé était difficilement exploitable quant à la dénomination des meubles et de leurs positions. Par conséquent, trouver les coordonnées spécifiques à chaque pièce a été une tâche très manuelle car il fallait les déterminer à tâtons. Heureusement, le modèle avait pour centre le salon, ce qui nous a donné un point de départ pour la recherche.



Le mode Marche

Par défaut, l'utilisateur survole la maison, mais il peut aussi s'y promener comme s'il marchait grâce à ce tag qu'on place dans la scène :

```
<NavigationInfo type='"WALK"' speed="100.0" visibilityLimit="100.0"
avatarSize="1.6 0.25 0.5" />
```



Pour avancer ou reculer, il suffit d'utiliser les touches de la souris, et faire en sorte d'augmenter la vitesse car sinon les mouvements de la caméra ne seront pas très visibles. Il est aussi possible de dimensionner notre avatar, que ça soit la hauteur de la vue, notre collision, ou la hauteur maximale pour enjamber les objets.

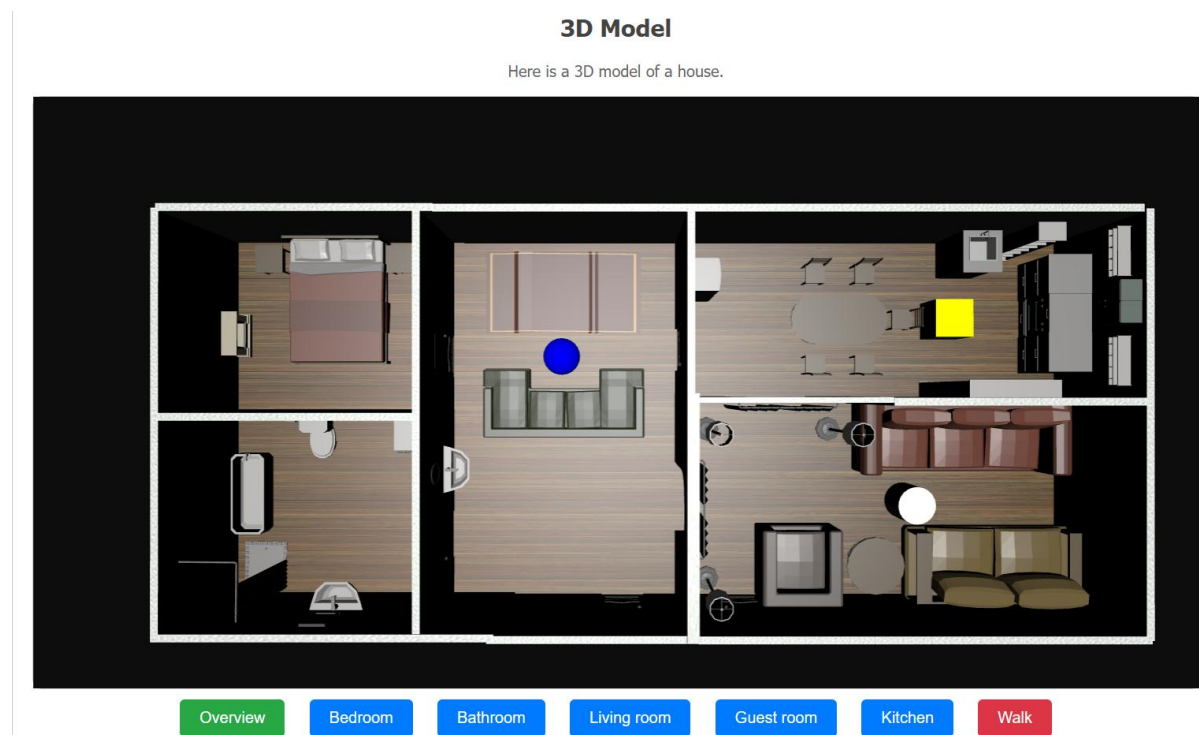
Pour des raisons inconnues, la vision ressemble à celui d'un animal de petite taille

(malgré les changements réalisés à la hauteur de vue, ou l'échelle des objets de la scène).

Interaction avec les objets

Toujours avec la conversion en X3D, les ID des composants ne sont pas très explicites. En effet, le fichier comporte plusieurs "Object" suivant une suite numérique et chacun ayant des paramètres de position comprenant plusieurs centaines de lignes.

Pour contourner ce problème, nous avons décidé de placer des objets primitifs (sphère bleue, cube jaune, et cylindre blanc) et les associer un événement si l'utilisateur clique dessus. La logique suit les boutons onClick() qu'on retrouve en HTML/Javascript.



Cliquer dessus permet d'afficher une alerte dans le navigateur :

