* Liste des pièces manipulées :
* [Coulissants | 2 Vantaux | Montant intermédiaire](https://backoffice.modeliz.v0.ql6625.fr/#pieceConfig0)
* [Volet | Eugene 02](https://backoffice.modeliz.v0.ql6625.fr/#pieceConfig1)
* [Volet | Eugene 01](https://backoffice.modeliz.v0.ql6625.fr/#pieceConfig2)
* [Eugene | Pilotage C2V2R](https://backoffice.modeliz.v0.ql6625.fr/#pieceConfig3)
* [Volet roulant | 2 elements Eugene](https://backoffice.modeliz.v0.ql6625.fr/#pieceConfig12)
* [Volet | Eugene 03](https://backoffice.modeliz.v0.ql6625.fr/#pieceConfig13)
* **Note importante** 
  + **Il est fortement recommander de jeter un coup d’œil sur la doc de Babylon.js** et plus précisément le getting started **https://doc.babylonjs.com/start**
  + Pour toute manipulation d’éléments de la scène, il faut toujours garder en mémoire que chaque pièce a **sa propre origine** qui diffère de l’origine de la scène. C’est ainsi que pour une véranda réduite ici à sa toiture et son corps, la position initiale de la toiture sera son origine de même que celle du corps.
* **La position initiale** ici veut dire ici la position que l’on a donné à la pièce lors de sa création sur Blender.
* **Récupération d’une pièce**: le mot clé **pieceInstance** permet d’accéder à la pièce à manipuler.
  + **pieceInstance.computed.mesh.getChildren**() permet d’avoir la liste d’éléments constituant la pièce.
  + **pieceInstance.computed.mesh** permet d’avoir l’instance mesh de la pièce. Ce qui permettra des manipulations de la pièce entière
  + pieceInstance.computed.mesh.translate : exemple de manipulation de la pièce
* Récupération et **manipulation** d’un élément de la pièce :

let coulisses = pieceInstance.computed.mesh.getChildren()[0];

coulisses.morphTargetManager.getTarget(0).influence = (Hauteur - 2150) / 1000;

* + **getTarget(**n**)** réfère à la « **shape key** n »va chercher la déformation « n » que l’élément de la pièce peut subir.
  + **Influence** : set ou get la valeur de la déformation
* **Autre** façon d’accéder et de déformer la pièce :
* pieceInstance.computed.mesh.getChildren().**forEach(mesh =>** {
* if (mesh && mesh.morphTargetManager && mesh.morphTargetManager.**getTarget(0))** {
* mesh.morphTargetManager.getTarget(1).**influence** = (Longueur - 2400) / 1000;
* mesh.morphTargetManager.getTarget(0).influence = (Hauteur - 2150) / 1000;
* }
* });
* **Note :** une rotation est faites par rapport à un axe. Ainsi, pour incliner vers l’avant une pièce par le bas suivant le repère (x,y,z) on effectuera la rotation sur l’axe x de la pièce.
* De même pour une rotation aux alentours de la pièce, on effectuera la rotation sur l’axe y de la pièce.
* Enfin, pour une rotation de la pièce suivant le sens des aiguille d’une montre, on effectuera la rotation sur l’axe z de la pièce.
* **Rotation de la pièce** 
  + pieceInstance.computed.mesh.addRotation(0, 0, Math.PI / 2);
  + Math.PI = 180 degrés
* pieceInstance.computed.mesh.translate(new Vector3(1, 0, 0), 150 / 1000);