

Florent LE PRINCE ([f1leprin@enib.fr](mailto:f1leprin@enib.fr))

Nicolas GINDRE ([n1gindre@enib.fr](mailto:n1gindre@enib.fr))



## Réalité Virtuelle

### Musée d'Ys

#### I - Prise en main

1. **ZQSD** : déplacement de la caméra
2. **Clic gauche** : orientation de la caméra (si le masquage du pointeur est désactivé)
3. **Espace** : Saut (doubles-sauts activés et illimités)

#### II - Fonctionnalités

1. **Événements de salle** : lorsque le joueur entre dans une pièce, un événement déclenche des animations (musique pour les pièces des tableaux, film pour la salle de projection, lumières pour les 4 salles). Le déclenchement peut néanmoins échouer si le joueur reste dans l'entrée sans entrer franchement dans la salle.
2. **Parcours via objets remarquables** : lorsque le joueur vise une table avec des bibelots, la “hitbox” de l’ensemble apparaît progressivement et le curseur s’anime. Si le joueur ne touche pas aux boutons ZQSD (s'il reste immobile), fait sortir le curseur de la “hitbox” et vise une seconde fois la “hitbox”, la caméra va se déplacer toute seule vers la table et s’arrêter devant elle.
3. **Apparence** : Le joueur possède une lampe qui se déplace avec lui. Une autre boule lumineuse se trouve dans le hall et se déplace quand on entre en collision avec elle. Les murs sont infranchissables et on peut monter l’escalier. La mer est animée (uniquement tout près de la plate-forme pour économiser des ressources). Un pingouin se déplace dans le musée. La skybox est une plage de la baie de Douarnenez (emplacement supposé proche de la ville d’Ys). Faute de temps, les tableaux et la musique n’ont pas de rapport avec la ville d’Ys (mais le film oui).

#### III - Précisions techniques

1. La physique de l'environnement virtuel (murs et sols infranchissables, gravité appliquée au joueur et à la sphère au sol), la mer animée et les contrôles améliorés (ZQSD, saut et pointeur masqué) sont possibles à l'aide de la bibliothèque “**aframe-extras**” disponible à cette adresse : <https://github.com/donmccurdy/aframe-extras>. Il faut ajouter l'attribut “kinematic-body” aux objets contrôlés par le joueur (caméra en l'occurrence) afin qu'ils soient infranchissables et soumis à la gravité, “static-body” aux objets infranchissables mais fixes, et “dynamic-body” aux objets qu'on peut pousser et soumis à la gravité. Les modèles complexes ne peuvent pas bénéficier de ces effets, il faut leur ajouter manuellement un mesh-collider (exemple : un plan sur l'escalier).
2. Les **textures** sont en général trouvées via google images en y incluant le terme “seamless” afin de pouvoir répéter les textures sur une surface sans que cela ne se voie, en conservant une certaine homogénéité. Nous les avons compressé à 80% (format JPEG) et réduit aux alentours de 1024x1024 pour fluidifier le jeu. La skybox de la plage a été trouvée dans un album photo, prise en mode panoramique avec un smartphone.
3. Les **modèles** sont libres et ont pour la plupart été récupérés sur le site suivant : <http://www.sweethome3d.com/freeModels.jsp>. Seuls ceux qui pesaient moins de 30 Ko ont en général été sélectionnés.
4. Une **lumière** directionnelle a été placée de façon à coïncider avec l'image du soleil sur la skybox. Une lumière ambiante a été ajoutée pour simuler la lumière reflétée par tout l'environnement. Les lumières restantes sont de type point et sont attachées à diverses sphères.
5. Des **animations** diverses (hitboxes, curseur et positions) sont déclenchées sur écoute de l'événement “stateadded” pour lorsque l'état en question est un survol de souris. On peut ensuite émettre des événements personnalisés pour déclencher les animations voulues. Les animations sur présence dans une salle sont déclenchées en écoutant l'événement émis lorsque le joueur entre en collision avec des plans invisibles disposés dans ces salles.
6. Faute de temps, l'animation du **pingouin** a simplement été réalisée à l'aide d'un switch-case qui réceptionne les événements du pingouin et émet celui qui doit succéder pour que les animations se déclenche dans le bon ordre. Cette écriture est cependant trop lourde et est peu pratique s'il faut ajouter d'autre pingouins.

## IV - Problèmes

1. Bien que ce projet fonctionne très bien sur les ordinateurs utilisés à l'ENIB, les résultats peuvent être assez divers selon la configuration de l'utilisateur, même lorsque son ordinateur est plus puissant (trop de modules Firefox ?) : **problèmes de fluidité**.
2. Les **collisions** sont encore très expérimentales : si on traverse trop vite un mesh censé être infranchissable (lors d'une chute), on peut passer à travers. De même, il peut arriver que le joueur s'enfonce dans le sol (généralement quand il recule avec S) là où nous avons joint les plans (dans

ce cas, utiliser la touche de saut pour se dégager). En résumé : La touche S (le recul) provoque presque systématiquement des problèmes de collision dans “aframe-extras”, éviter de l'utiliser dans la mesure du possible.