

RAPPORT

ANIMATION D'UN

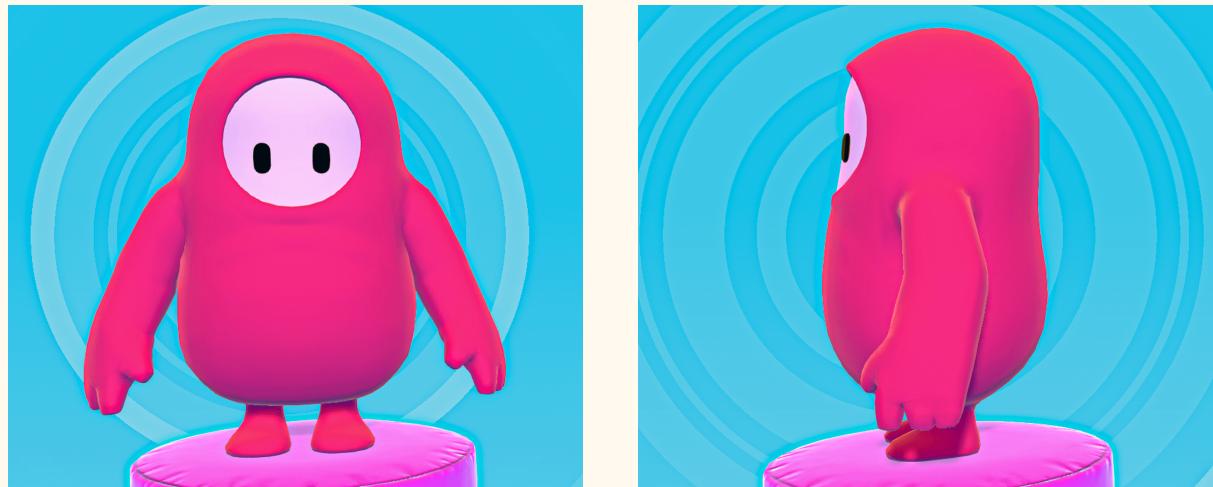
PERSONNAGE 3D

THEO LIEFFROY - ENZO LICATA



Choix du sujet

Le but du projet étant de modéliser et animer un personnage 3D sur Blender, nous avons décidé de reproduire un personnage du célèbre jeu vidéo *Fall guys* et de l'intégrer dans une scène ressemblant au monde virtuel qui lui est associé :



Fall guys est un jeu multijoueur, sorti en juin 2019, dans lequel chaque joueur incarne un “haricot” dans une course à obstacles.

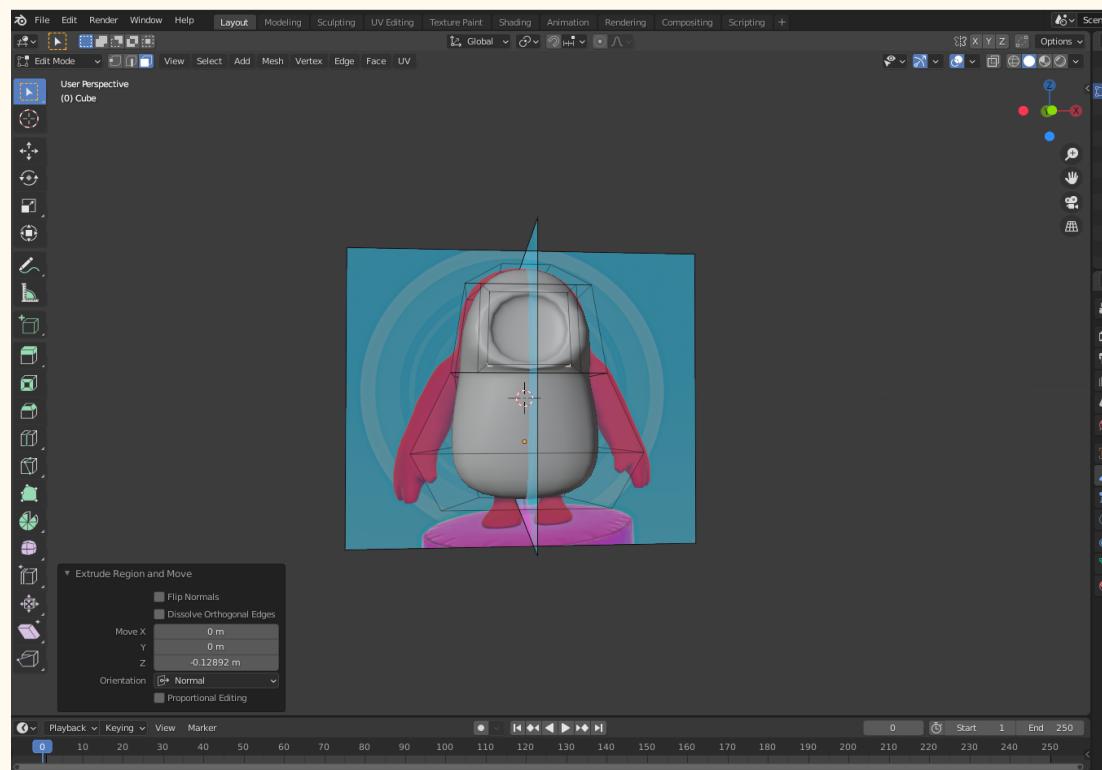
Le but du jeu est donc d’arriver premier parmi une cinquantaine de joueurs. Le haricot est capable de courir, sauter, plonger vers l'avant, et attraper d'autres joueurs.



Modélisation du personnage

Dans un premier temps, il nous fallait modéliser notre haricot en 3D. Pour ce faire, nous avons utilisé deux images planes qui nous ont servi de calques.

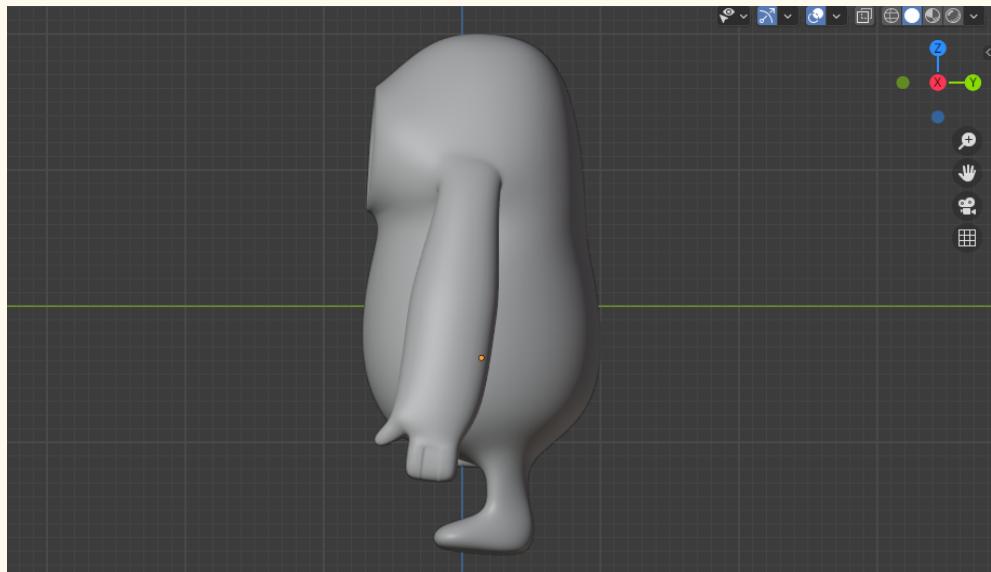
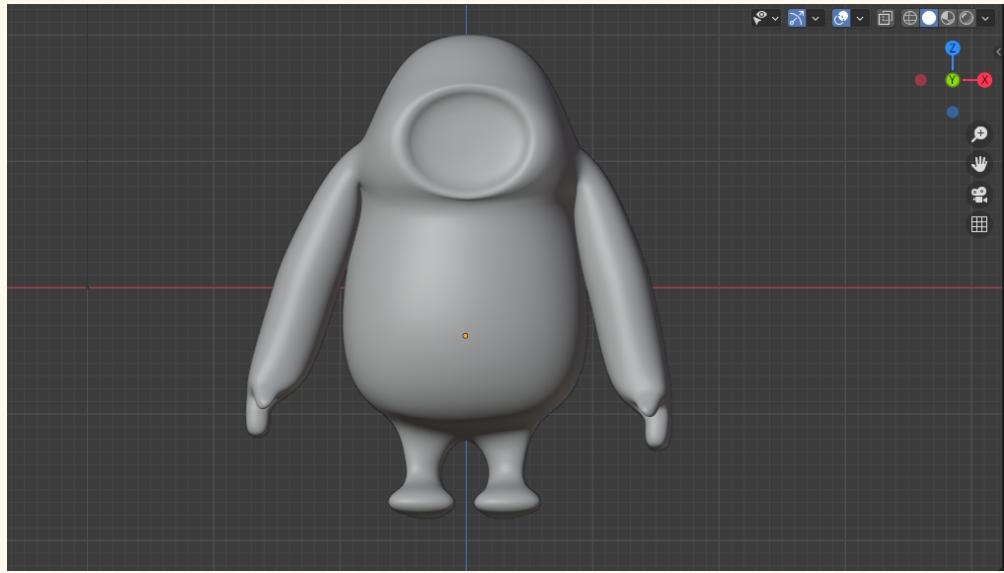
Ainsi, à partir d'un cube que l'on extrude plusieurs fois puis auquel on ajoute un *subdivision surface* et des *loop cuts* on vient épouser la forme du corps de notre personnage :



Afin d'avoir une surface plus lisse, on *smooth* notre modélisation.

Les bras et les jambes sont de simples extrusions de faces du corps principal sur lesquelles ont appliqués du scaling et des rotations.

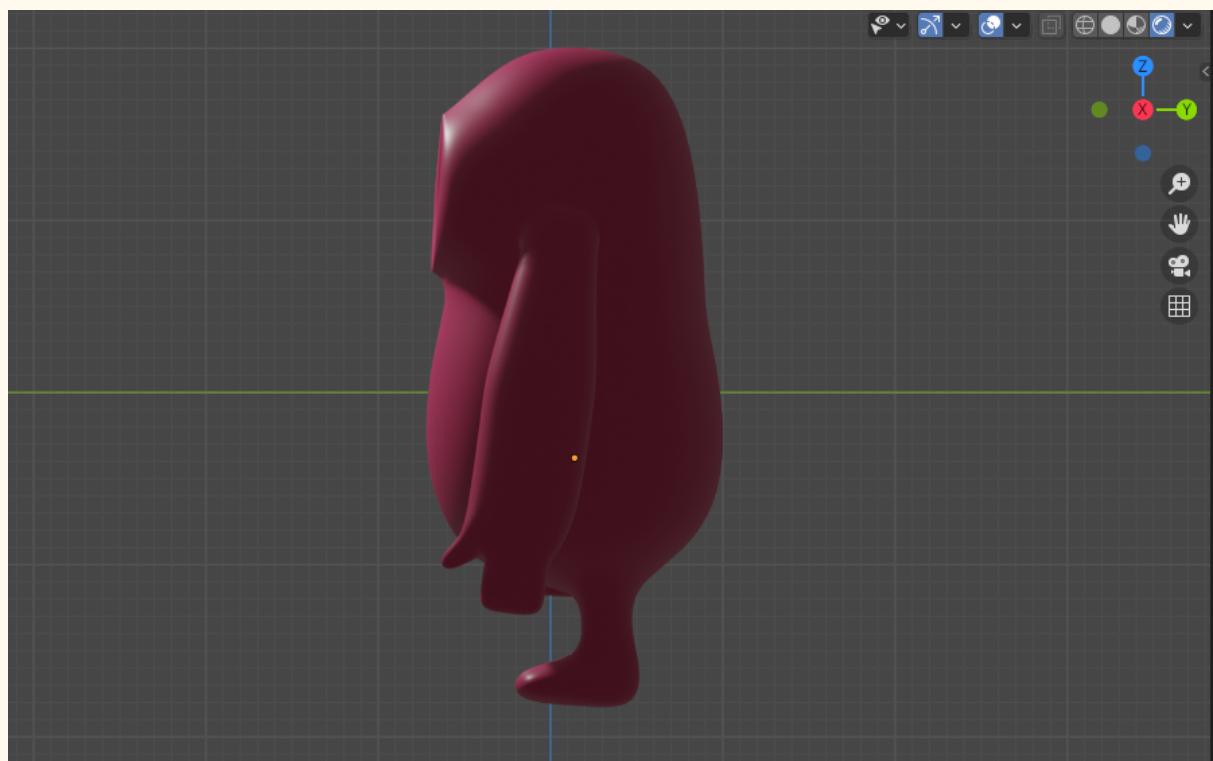
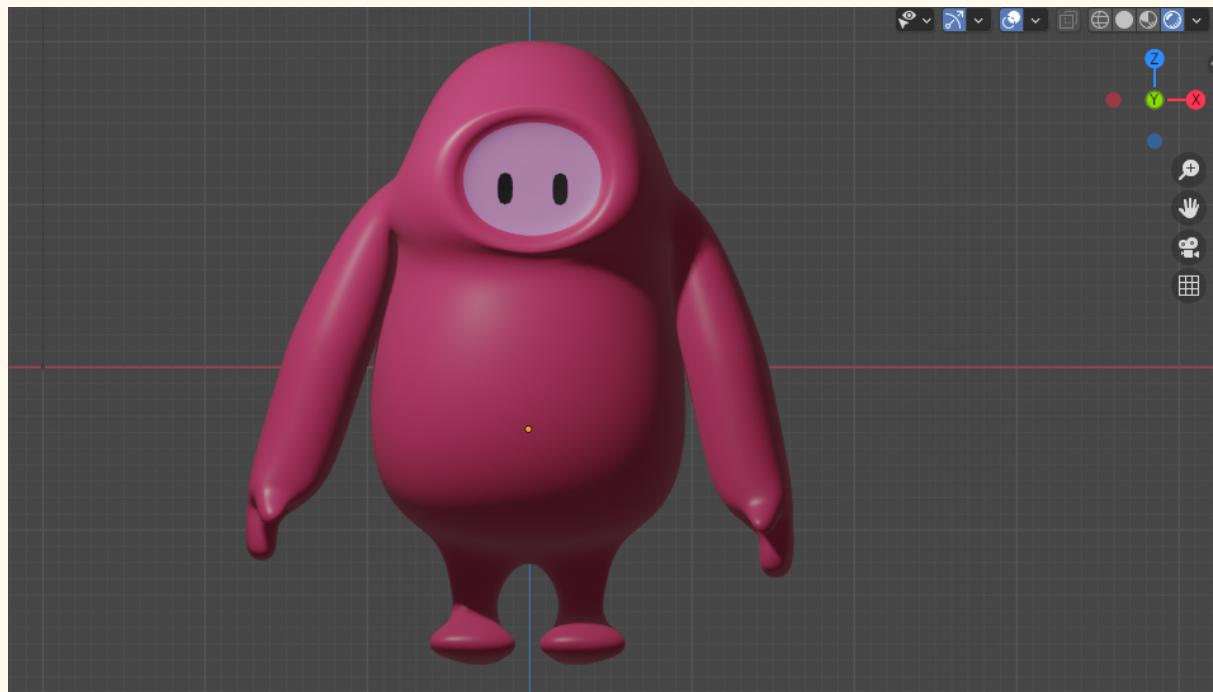
Ainsi, nous obtenons le résultat suivant :



Maintenant, il ne nous reste plus qu'à ajouter les yeux et les différentes couleurs.

Pour les yeux nous utiliserons deux plans modifiés et auxquels nous avons appliqué un *shrinkwrap* (mettre en contact deux objets).

Ainsi, le résultat final obtenu est le suivant :



Modélisation de la scène

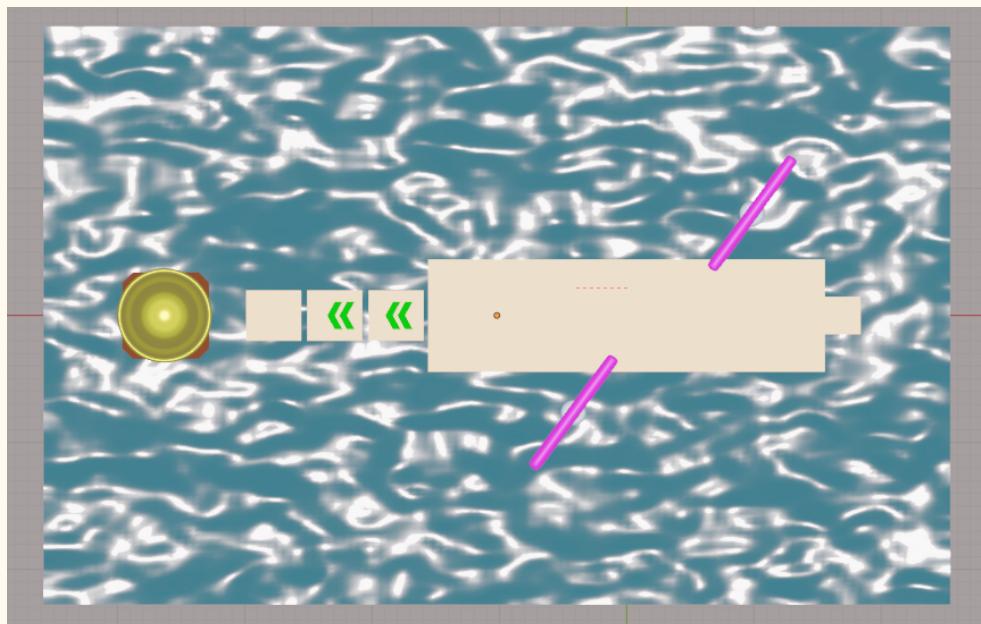
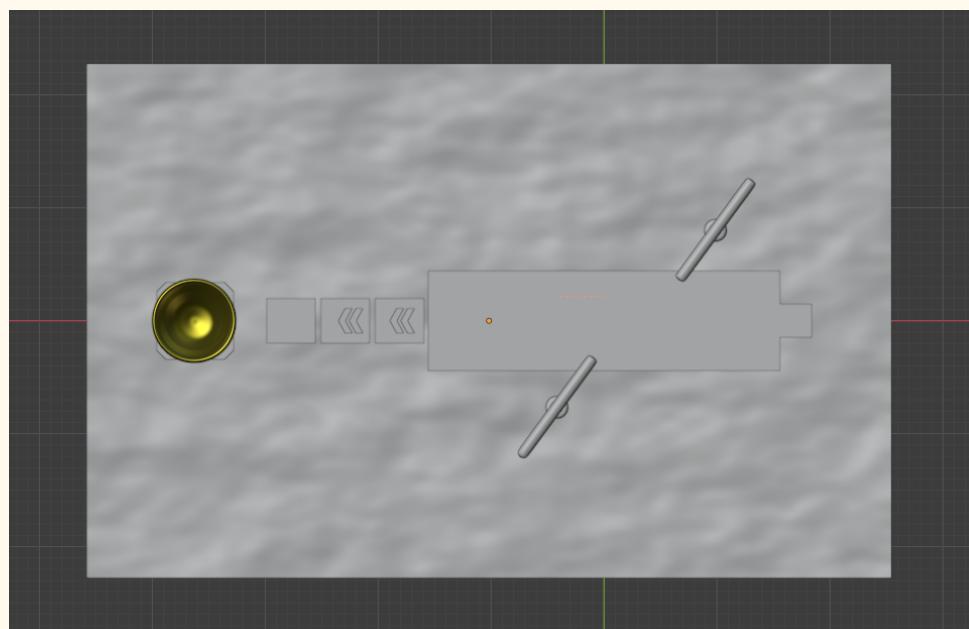
Afin de mettre en scène notre personnage, il nous faut un environnement dans lequel il pourra évoluer. Pour ce faire, nous nous inspirons de la carte du jeu en modélisant un de ses obstacles incontournables : les boudins rotatifs.

On crée d'abord une simple plateforme qui nous servira de parcours.

Deux boudins rotatifs sont placés de part et d'autre de cette plateforme afin de gêner la progression du personnage, puis, pour rejoindre la coupe symbolisant la fin de la course, il devra passer par trois petites plateformes.

Afin d'étoffer la scène, on ajoute une couche d'eau sous la plateforme.

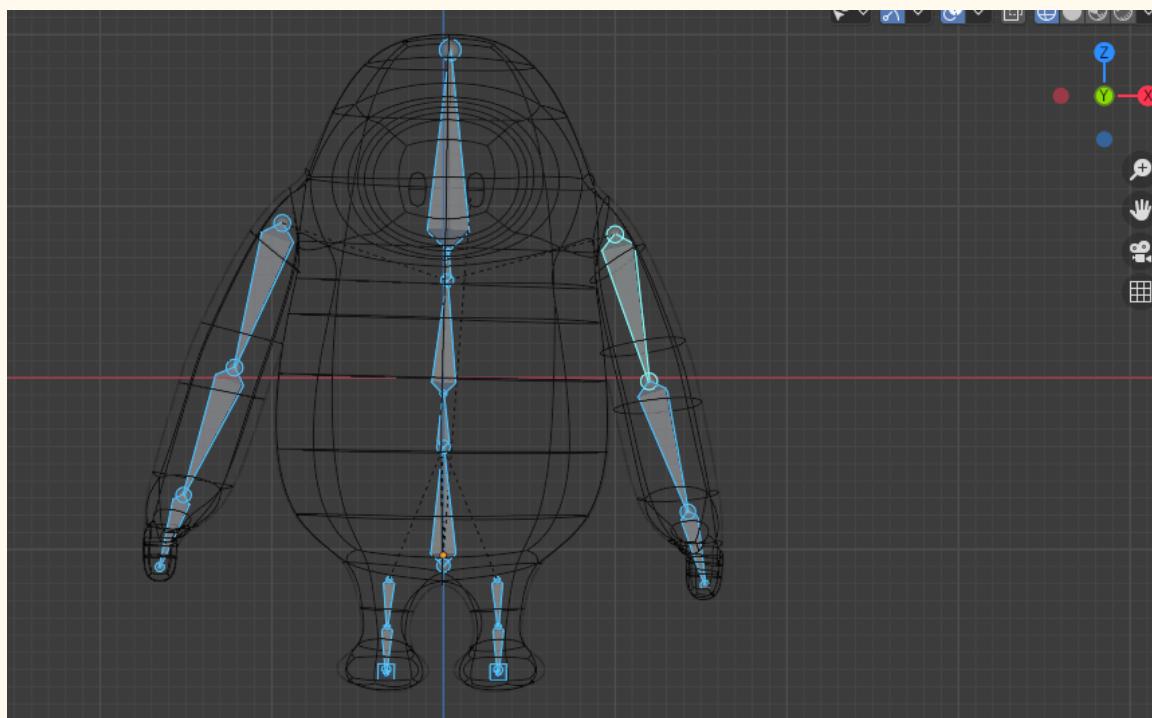
Pour l'éclairage, nous utilisons l'add-on *Dynamic Sky* déjà intégré à Blender afin d'habiller le ciel et d'avoir une ambiance plus fidèle au jeu.



Animation du personnage

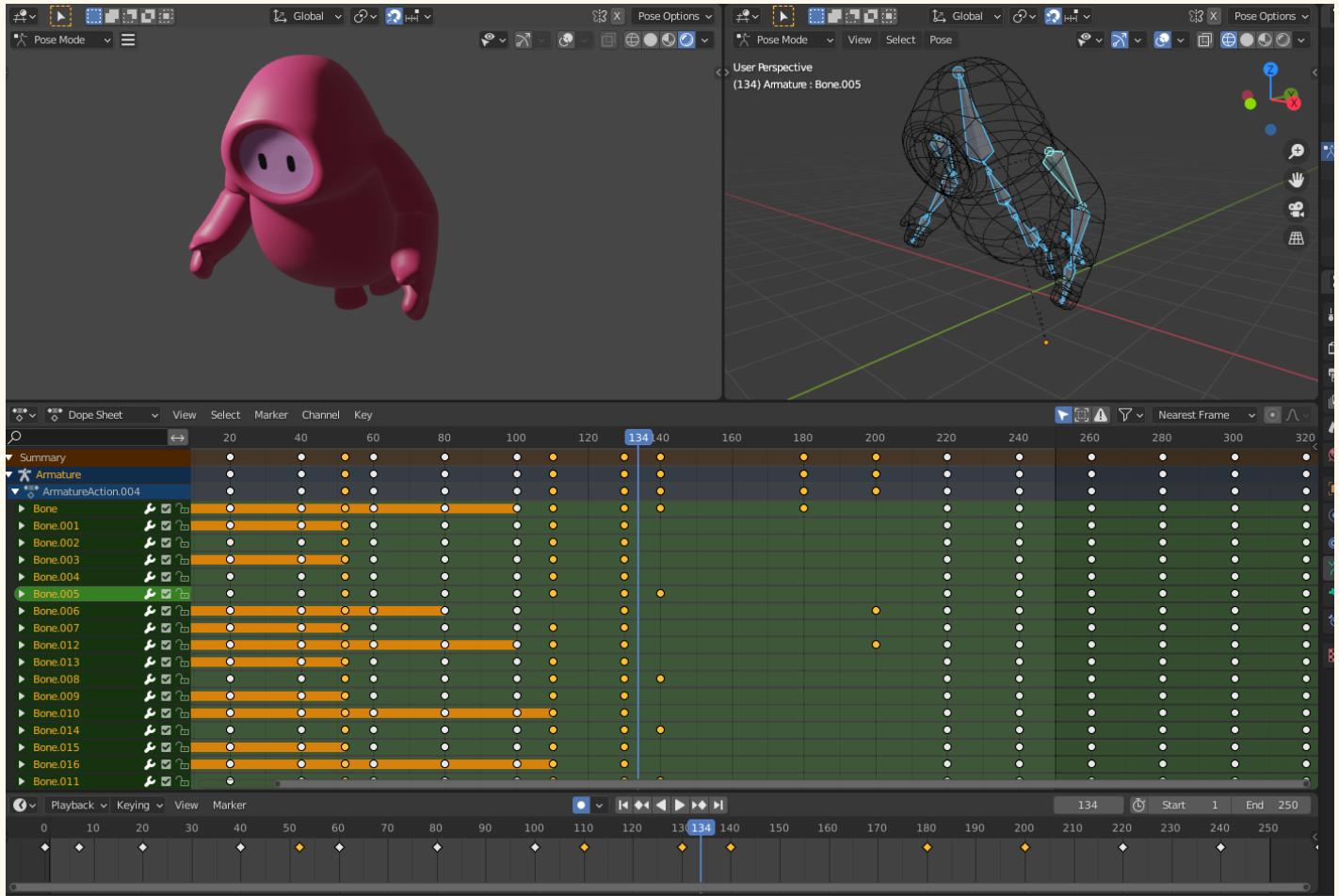
Avant d'incruster notre personnage dans notre scène, il nous faut l'animer. Pour ce faire, nous avons choisi de faire marcher notre personnage puis de lui faire réaliser un saut en avant, puis de lui faire reprendre une course normale. En effet, ce type d'animation nous permettra de reproduire au mieux les mouvements que nous pouvons retrouver dans le jeu officiel *Fall guys*.

Dans premier temps ils nous faut créer une armature à notre personnage :



Cette armature constituée d'os nous permettra de générer des mouvements à notre modèle 3D une fois les contraintes de poids appliquées.

Afin de réaliser le film de notre animation, on génère des *keyframes* affiliées à des positions spécifiques que nous voulons que prenne notre personnage :



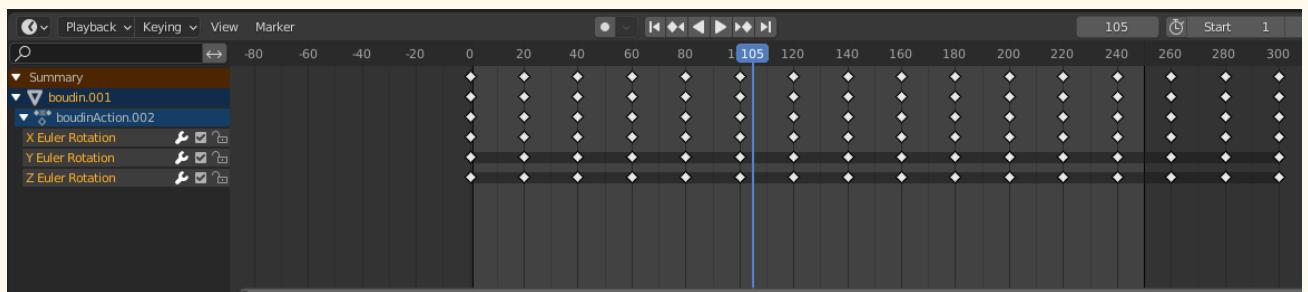
Nous avons, ainsi, réalisé notre animation. Il ne reste plus qu'à insérer notre personnage dans notre scène finale.

Animation de la scène

Pour l'animation de la scène, il suffit de faire rotationner les boudins sur l'axe Z grâce à un script python écrit dans Blender.

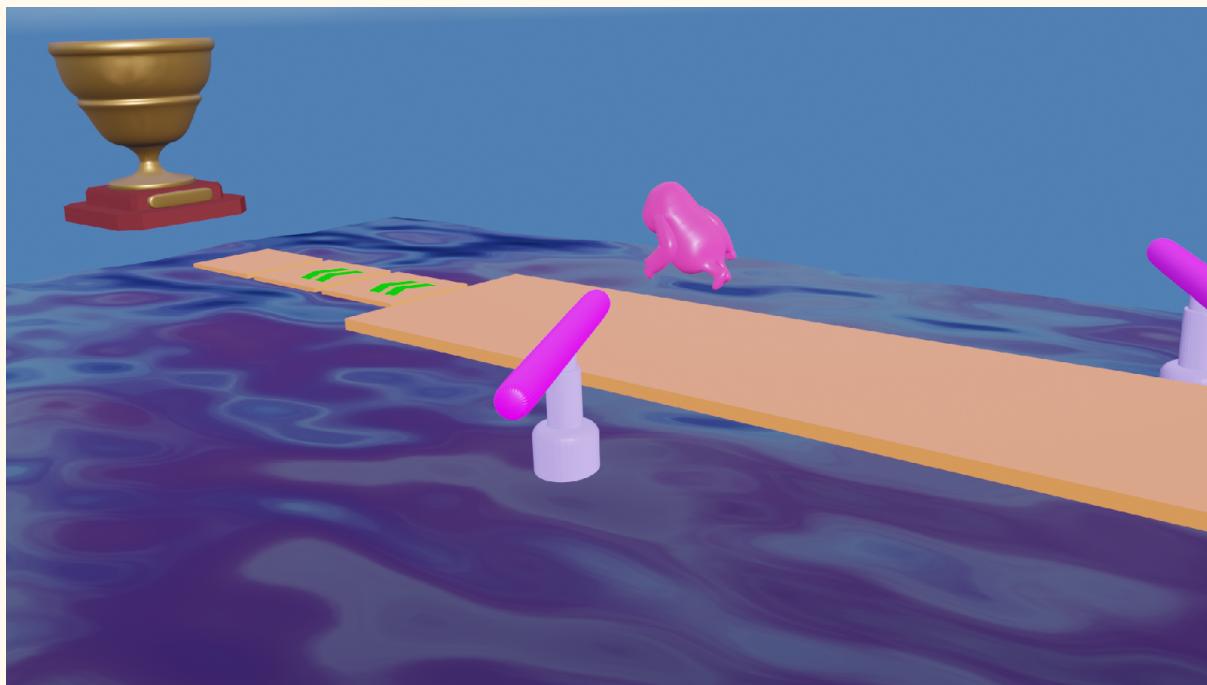
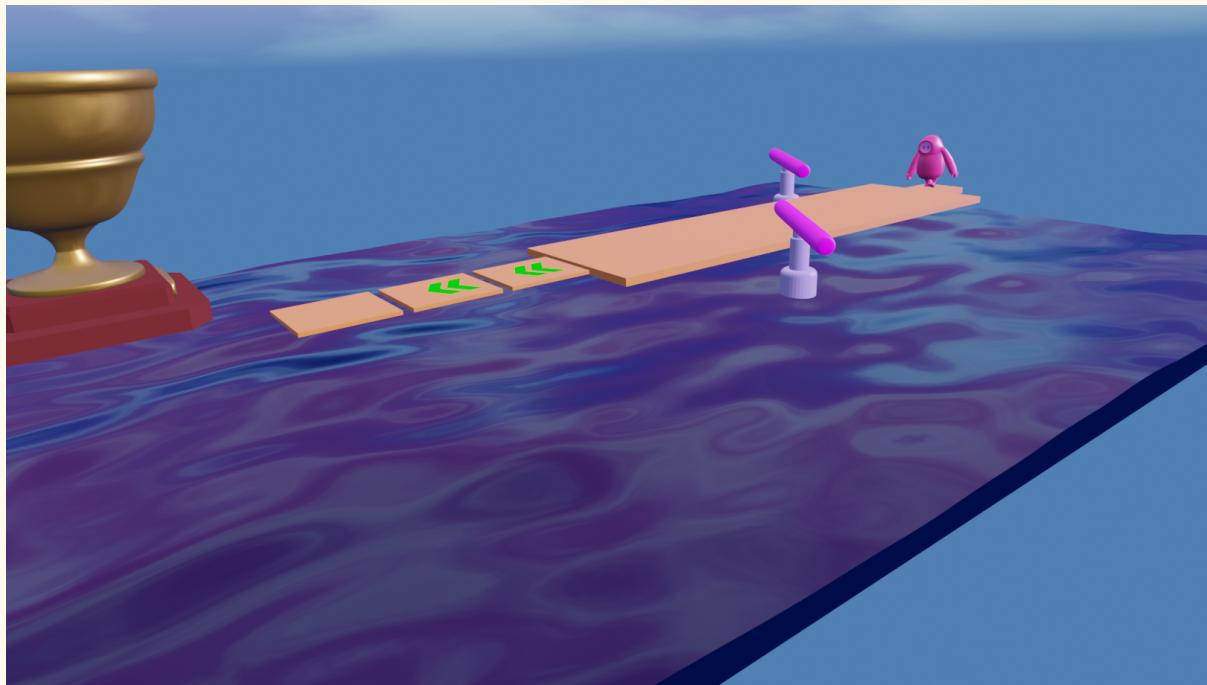
```
1 import bpy
2
3 obj= bpy.data.collections["collect"].objects
4 for i in range(16):
5     bpy.ops.transform.rotate(value=i/16 * 3.14159, orient_axis='Z', orient_type='GLOBAL', orient_matrix=((1, 0, 0),
6     (0, 1, 0), (0, 0, 1)), orient_matrix_type='GLOBAL', constraint_axis=(False, False, True), mirror=True,
7     use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1,
8     use_proportional_connected=False, use_proportional_projected=False)
9     obj[0].keyframe_insert(data_path = "rotation_euler" , frame = i * 20) |
```

Ce script permet de générer des *keyframes* à intervalles réguliers de 20 frames sur lesquels les boudins effectuent leur rotation.



Scène finale

Il est maintenant temps de rassembler notre scène avec notre personnage et de corriger les petits écarts d'animation :





FIN