**Unity**

**Gestion des textures**

Texture en puissance de 2 : optimisation pour le calcule

Si pas en puissance de 2 prend plus de place en mémoire vidéo

Pas besoin de puissance de 2 pour le type GUI

Unity supporte les textures en non puissance de 2. Il optimise lui-même.

Ecrire sur une texture c’est changé la couleur des pixels

Pour ce faire, le dev a besoin d’une texture activé en écriture et lecture Read/Write enabled

*Texture Type* : Advanced (permet de mieux gérer sa texture, et on peut y utiliser tous les autres pressets)

Alpha from Gray Scale, permet de gérer une texture qui n’a pas de couche alpha mais en niveaux de gris

Alpha transparence : pour créer une transparence pour la couche alpha de la texture

On voit que la texture a une couche alpha si le bouton d’activation est présent dans le preview

Création d’un alpha avec : Alpha is Gray Scale

**Mip Map** : Plus petite version d’une texture : Permet de gérer l’affichage de la texture avec la distance. Evite les bouillis de pixels vus de loin. Evite le problème de flickering

On l’active sur un jeu 3D.

Wrap Mode : Clamp : pose des limites > pas d’enroulement (début et fin de textures pas plus). On clampe quand la texture ne doit pas etre répétée.

Sinon Repeat : Trait blanc (début de la texture d’après) si la texture n’est pas répétée

Filter mode : On met point quand on ne veut pas que la texture s’étire, permet d’avoir un ratio de 1 et le texel. (pixel de l’écran et texel (pixel) de la texture) : Il faut que la texture s’affiche à l’écran dans la meme résolution qu’est la sienne (texture de 256\*256 prend à l’écran 256\*256 pixel)

Sprite avec mode repeat + filtre on va bien voir le trait

Max size : défini la taille max de la texture et change dynamiquement les surplus (1024 \* 1100 passe en 1024 \* 1024 avec déformation)

Transformation sans détérioration : Miroir, Croping (rogner)

Choisir Automatic Truecolor et Automatic 16bits (aucune compression)

Material : permet de dire comment on doit rendre la texture

Couleur de la texture

Couleur de la diffuse

Choix du shader (quand on appuie un shader on applique une passe de rendu supplémentaire)

Shader le moins couteux : Unlit ; ensuite c’est le VertexLit et enfin Diffuse (mélange de la couleur de la lumière et couleur texture)

Bumped : deux passes de rendu (drawcall) veut 2 textures

Paralax diffuse : texture de base, normal map et height map

Tiling = nombre de fois qu’une texture se répète.

Offset décale la tête de lecture de la texture (souvent utilisé pour les gifs)

**Contrainte d’une interface :**

* Premier plan
* Résolution adaptative (coordonnées relatives (au viewport = vue de la caméra))
* Les éléments 2D ne tournent pas = intérêt de l’interface 3D

ex : GUITexture ne peut pas tourner

**Support de l’interface**

Minimum deux faces pour une interface 2D

Orienté vers la caméra : Z vers la caméra

Quad (dans Create > Quad) = plan de 2 faces avec un Z qui nous regarde.

Sur le quad, on y applique le material qui possède la texture.

* Contrainte de résolution ? (va dans la mémoire vidéo) Mieux vaut avoir la texture en puissance de 2.

Comme l’interface ne bouge on peut faire de la non-puissance de 2. Mais faut pas qu’il bouge. Parce que l’info est envoyée qu’une fois à la mémoire vidéo.

La size correspond à la moitié de la hauteur de la résolution voulue (celle affichée au final) (**pour la caméra orthographique**).

Si on a un jeu en 800\*600, on met la hauteur/2 dans le champ : Size, soit 300 (on peut diviser par 100 ou 1000 donc 3 ou 0.3).

Pour une size de 300 on aura un plane (qui porte la texture de 256/256) de 256. (en X et Y)

Size = 3 > Plane = X 2.56 ; Y 2.56 (on peut aussi en Z)

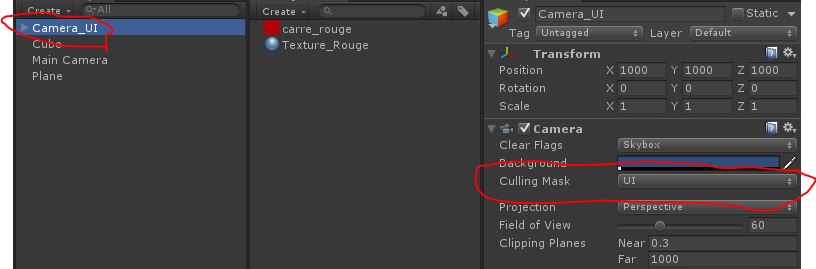
**On choisi un material de type unlit pour la lumière n’interagisse pas avec l’objet de l’interface.**

**Point = Aucune interpolation de couleur de la texture. Moins couteux !**

On hiérarchise les objets de l’interface dans la caméra dédiée.

On joue avec le culling mask pour jouer avec le rendu.

Camera de l’HUD doit avec le culling mask de l’HUD

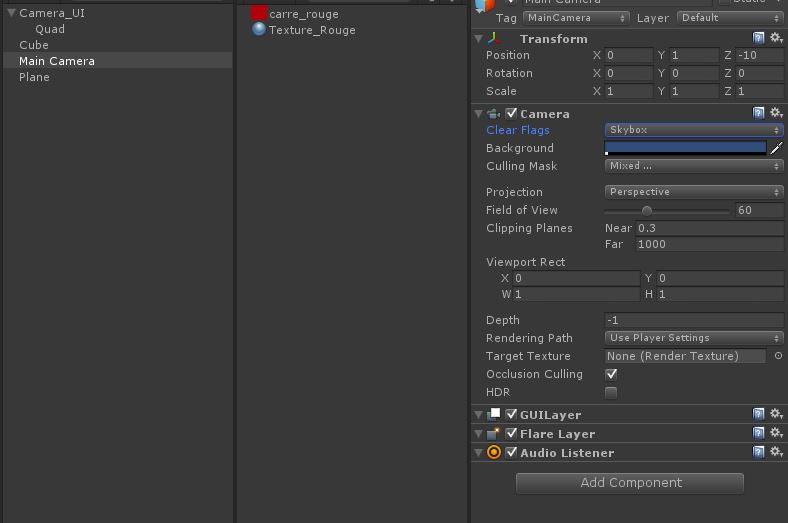


Les objets doivent avoir le layer du culling mask



Main Camera on enlève le culling de l’HUD

MainCamera doit avoir SolidColor et Skybox (**Clear Flage**) : Depth = -1



Camera HUD : depth only : Depth = 0 ou 1 (>-1)

