

# Exercice intro Shader

## Initialisation du projet

1. Créer un nouveau projet Unity
  - a. Appeler ce projet Cours16\_Shader
  - b. Créer un repository sur Github pour ce projet
2. Appeler votre scène « ShaderSand »
3. Créer un « Standard surface shader » dans unity
4. Créer un un material avec ce shader.
5. Créer un « Plane » dans votre scène.
6. Appliquer le material à votre plan.

## Création du surface Shader

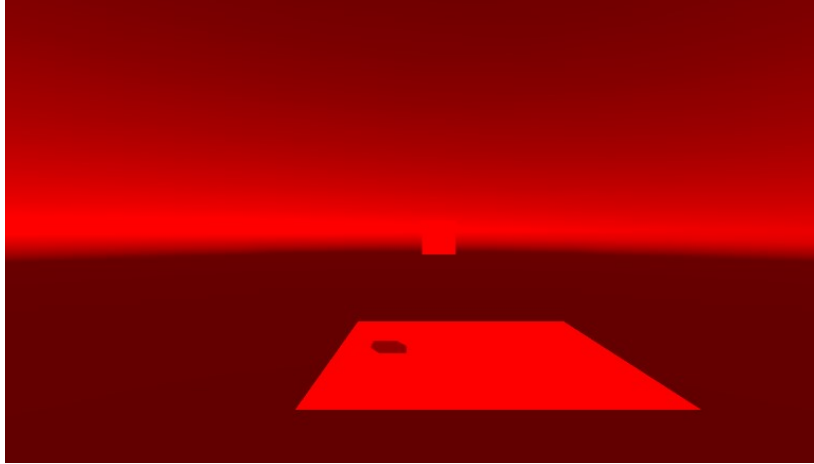
1. Regarder le vidéo fourni avec l'exercice. Vous devez reproduire ce shader.
  - a. Utiliser un standard surface shader
2. Dans les propriétés de votre Shader vous devez avoir
  - a. Une texture principale.
  - b. Un nombre entre 1 et 100 pour la vitesse de déroulement
3. Ajouter la texture « sand.png » dans votre projet
4. Ajouter la texture « sand.png » comme texture dans votre material
5. Ajouter la variable nécessaire pour avoir accès à la vitesse de déroulement dans le SubShader.
6. Dans la fonction surf
  - a. Vous devez faire dérouler la texture sur elle-même. Une est fixe et l'autre déroule.
    - i. Une variable globale « \_Time » est disponible.
    - ii. La fonction « tex2D » prend 2 paramètres :
      1. sampler2d : La texture
      2. fixed2 : La position où aller prendre la valeur du pixel (x,y)
  - b. Effectuer l'opération nécessaire pour obtenir le résultat présent dans la vidéo.
  - c. Pour l'instant il est normal que la transparence ne soit pas fonctionnelle.
7. o.Alpha doit être transparent ou il y absence de couleur. o.Alpha = ?
8. Rechercher sur internet pour insérer la bonne instruction dans l'instruction « Tags » pour la gestion de la transparence
9. Rechercher sur internet pour ajouter le paramètre nécessaire pour la gestion de la transparence à la ligne : `#pragma surface surf Standard fullforwardshadows`
10. Faire un commit avec une version fonctionnelle

## Création d'un post Processing image shader

Ce type de shader sera pertinent pour effectuer des modifications à l'ensemble du rendu.

1. Créer un shader de type « Image effect shader ». Dans un shader de ce type :

- a. `#pragma fragment frag` : dit que la fonction « Frag » sera responsable de fragment. Cette fonction doit définir la couleur d'un pixel dans un shader de ce type.
- b. Modifier la fonction « Frag » pour que l'image soit seulement en nuance de rouge. Aucun bleu et aucun vert. Attention, la couleur vert et bleu ne doivent pas être de couleur noire. Voici le résultat à la fin de toutes les instructions.



- c.
2. Créer un material à partir du shader.
3. Créer un script sur la caméra : « CameraEffect »
4. Ajouter avant la définition de la classe
  - a. `[ExecuteInEditMode]`
- b. 

```
[ExecuteInEditMode]
public class CameraEffect : MonoBehaviour {
```
5. Ajouter une variable private de type Material
6. Dans une fonction awake :
  - a. `material = new Material(Shader.Find(...));`
7. Ajouter la fonction `OnRenderImage`
  - a. `private void OnRenderImage(RenderTexture source, RenderTexture destination)`
  - b. Rechercher sur internet comment la fonction `Graphics.Blit` fonctionne pour transformer l'image avec votre Shader.