

# Pratique Examen final

## Notion à savoir (liste non complète)

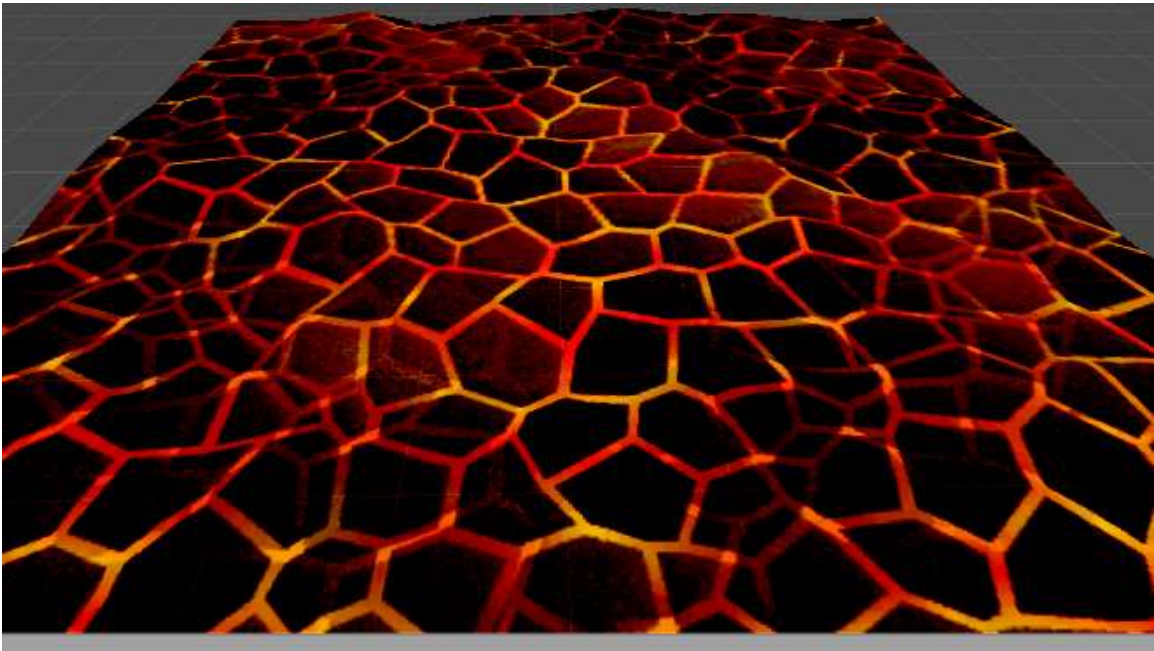
- **Être capable d'utiliser les shaders**
  - Être capable de modifier l'alpha dans un shader pour faire un shader transparent
  - Être capable d'utiliser plusieurs textures dans un shader
  - Être capable de faire un post-processing shader
  - Être capable de modifier les sommets (« vertex ») à l'aide d'un shader
  - Être capable de modifier le facteur de « Tessellation » dans un shader
  - Comprendre la input Structure d'un shader et être d'en utiliser les propriétés
  - Être capable d'écrire dans une texture à l'aide d'un shader
- **Être capable de faire un jeu multijoueur**
  - Connaître les différents « composants » du multijoueur, leur fonction et savoir comment les configurer
    - NetworkManager
    - NetworkHud
    - NetworkIdentity
    - NetworkTransform
  - Être capable d'utiliser les fonctions de base de la classe NetworkBehavior
    - IsLocalPlayer
    - IsServer
  - Être capable d'instancier des objets dans un contexte multijoueur
  - Être en mesure de faire bouger les objets de manières uniformes entre les divers clients
  - Être capable d'exécuter une commande sur le serveur seulement
  - Être capable d'exécuter une commande sur les clients
- **Connaître le Perlin noise et savoir comment il est possible de s'en servir**
- **Être capable de sauvegarder et de charger des sauvegardes**
  - Connaître et comprendre les différentes fonctions nécessaires à la sauvegarde et au chargement
  - Être capable de créer des classes sérializable et de créer une structure de sauvegarde cohérente
- **Être capable d'effectuer de simple modélisation dans Blender 2.8**
  - Ajout de mesh
  - Modification de sommet
  - Être capable d'extruder
  - Exportation en FBX
  - Importation dans Unity
- **Autres notions**
  - Être capable d'utiliser et configurer un particle système
  - Être capable de faire jouer des sons

- Être capable de créer un singleton avec une classe static
- Être capable de reconnaître une situation qui requiert un « DontDestroyOnLoad » et de l'utiliser
- Être capable d'effectuer des changements de scène
- Être capable de faire de simple animation
- Être capable d'ajouter des transitions dans un animator controller
- Être capable d'utiliser les événements des collisions et de comprendre la matrice de collision
- Être capable d'utiliser les diverses fonctions en lien avec la physique et un rigidBody
- Être capable de faire un raycast
- Être capable de comprendre la lumière(baked, emission ainsi que les divers émetteur)
- Être capable de comprendre la bonne utilisation du Time.Deltatime
- Être capable d'utiliser l'outil de terrain
- Être capable d'ajouter des boutons et d'effectuer des fonctions suite des actions sur ceux-ci
- Être capable de mettre à jour le texte dans l'interface

## Question 1 Shader

1. Créez un plan dans la scène
2. Créez un shader
  - a. Nommer ce shader « LavaShader »
3. Ouvrez le code de votre shader
4. Retirez tout code concernant le « Fog »
5. Ajoutez/modifiez le nécessaire pour que votre shader ait une propriété de texture
  - a. Nommer cette propriété « \_LavaTexture »
  - b. Lorsque utilisez avec un matériel, cette propriété doit avoir la description « texture de lave »
  - c. Cette texture doit être utilisée comme couleur principale
  - d. Cette texture doit défiler « Scroll » en X et Y
6. Ajoutez/modifiez le nécessaire pour que votre shader ait une seconde propriété de texture
  - a. Nommer cette propriété « \_RockTexture »
  - b. Un matériel utilisant votre shader doit avoir cette propriété disponible sous le nom « Texture de roche »
  - c. Il sera décrit plus loin comment utiliser cette texture
7. Ajouter/modifiez le nécessaire pour que votre shader ait une seconde propriété de type float
  - a. Nommer cette propriété « \_DisplacementFactor »
  - b. La valeur doit être comprise entre 0 et 1
8. Ajoutez/modifiez le nécessaire pour que votre shader ait une quatrième propriété de texture
  - a. Nommer cette propriété « \_DispTexture »

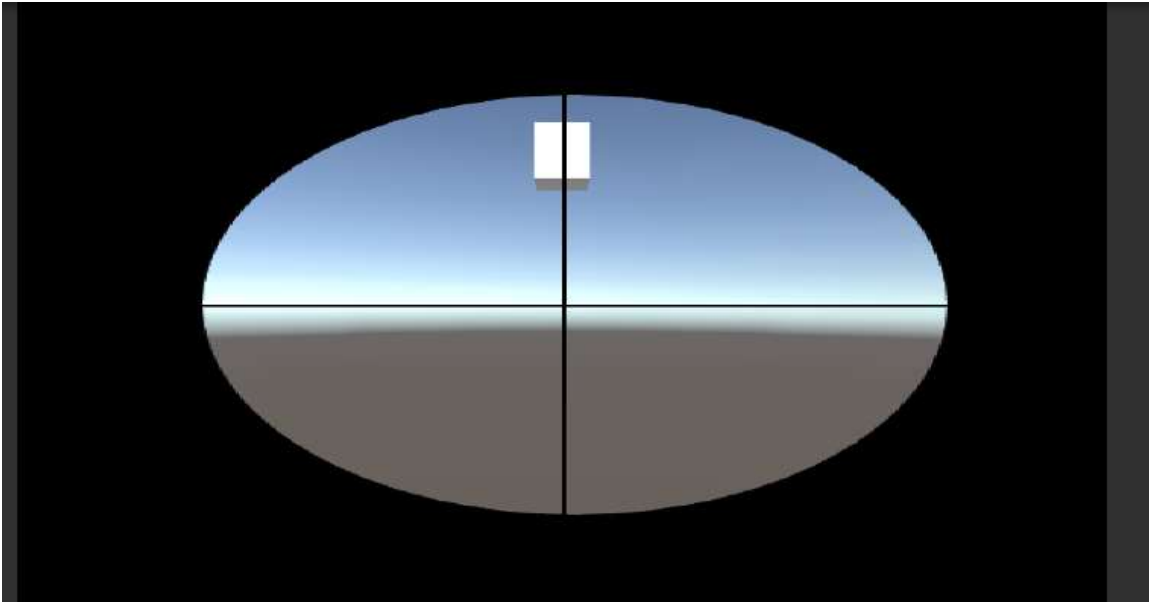
- b. Un matériel utilisant votre shader doit avoir cette propriété disponible sous le nom « Texture de déplacement »
  - c. Modifier le nécessaire pour modifier le haut des sommets en utilisant cette texture
    - i. Une valeur de couleur noire indique le sommet doit être soulevé par rapport à la normale
    - ii. Une valeur de couleur blanche indique qu'aucune modification n'est nécessaire au sommet
    - iii. Toute valeur entre le blanc et le noir doit être proportionnellement soulevée selon les instructions i et ii.
    - iv. Utilisez la valeur « `_Displacement` » comme facteur de déplacement, 0 indique aucun déplacement alors que 1 indique beaucoup de déplacement.
9. Utiliser la texture de roche en combinaison avec la couleur pour obtenir un résultat similaire
- a. Créer un Material à partir de votre shader
  - b. Appliquer le Material à votre plan



## Question 2 Shader post-processing

1. Créez un shader du type « Image effect shader » et appelez le « ShaderQ2 »
2. Ajouter/modifiez le nécessaire pour que votre shader ait une seconde propriété de type texture 2D
  - a. Nommer cette propriété « `_AimTexture` »
  - b. Utiliser la texture pour modifier l'image et obtenir le résultat à la fin de la question.
3. Créez un script sur la caméra et appelez le « AimPostProcessing »
  - a. Faites en sorte que ce script s'exécute sans être en mode « Play »

- b. Faites le nécessaire pour que le script utilise le shader pour modifier l'image rendue.



### Question 3 Multijoueur

1. Créez un empty GameObject et nommez le « NetworkManager »
  - a. Ajouter le/les component nécessaire pour la gestion du multijoueur en mode local « LAN ». Il doit y avoir le menu qui permet de créer une partie « Host » et la possibilité de rejoindre une partie comme client.
2. Ajoutez le nécessaire pour obtenir 2 positions de départ pour les joueurs.
  - a. Le prefab « Player » doit être instancié en alternance entre les positions de départs. Des modifications et/ou ajouts sur le prefab « Player » peuvent être nécessaires.
3. Ajouter le nécessaire au prefab « Trap » pour qu'il seulement visible par le joueur propriétaire « Owner ». La modification du Script « Trap » sera nécessaire.
  - a. Le piège « Trap » doit être instancié pour tous les joueurs, mais est seulement visible pour le propriétaire. La fonction « SetOwner » dans le script « Trap » est l'endroit suggéré pour ajouter le code nécessaire.
  - b. La collision doit être seulement exécutée par le serveur, et ce pour tous les pièges.
4. Ajoutez/modifiez le nécessaire pour que joueur « Player » fonctionne correctement dans le cadre d'une partie multijoueur. Les modifications des scripts « PlayerMovement » et « LayTrap » seront nécessaires.
  - a. Le joueur local doit uniquement contrôler son personnage
  - b. Ajoutez/modifiez le nécessaire pour permettre la synchronisation des déplacements du personnage.
  - c. Le client doit être responsable de définir la position du joueur local.

## Question 4 Perlin noise

Le but de cette question est de générer la création et le mouvement de banc de poissons en utilisant le perlin noise.

1. Dans le script « FishGenerator »
  - a. Complétez la méthode GenerateFishes pour générer des poissons en utilisant le perlin noise
    - i. Utilisez la Classe noise.
    - ii. Modifiez les paramètres dans l'éditeur pour obtenir plusieurs bancs de poissons en créant des objets avec le prefab « Fish »
2. Dans le script « MovementGenerator »
  - a. Complétez la méthode « Update »
    - i. Utilisez la Classe noise.
    - ii. Utilisez le Rigidbody2D pour appliquer une force
    - iii. Méthode pertinente possible : Math.Abs, Math.Round, Modulo (%)
    - iv. Toutes les 5 secondes, changer la direction des bancs de poissons.

## Question 5 Sauvegarde et chargement

Bien que vous en soyez libre, vous ne devriez pas à avoir à modifier les boutons ou toute autre méthode que celle spécifier dans les instructions.

1. Compléter le code nécessaire à la sauvegarde dans le script « SaveAndLoad »
  - a. Créez dans le script la classe pour conserver les informations du joueur
    - i. Quantité de pétrole
    - ii. Quantité d'or
    - iii. Quantité d'argent
  - b. Complétez la fonction « SaveGame »
  - c. Complétez la fonction « LoadGame »