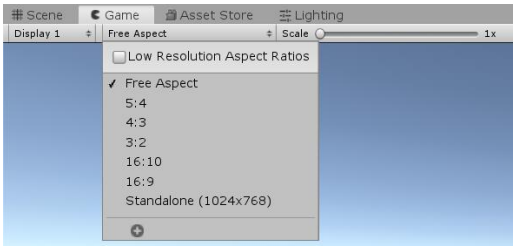
TP2 Unity

Pour réaliser ce TP, repartez du projet créé lors du TP1.

1. Interface graphique

Image

- Créez une nouvelle scène et appelez-la « Menu ».
- Choisissez une image de fond pour votre menu et importez-la dans votre projet. Modifiez ses paramètres d'import en « Sprite (2D and UI) ».
- Créez un objet de type « UI / Image », placez-la à la position (0,0) et appliquez-lui l'image précédemment importée. Cliquez sur « Set Native Size ».
- Dans l'onglet « Game », testez différents ratios d'affichage. L'image est-elle affichée correctement ?



- Activez l'option « Maximize on play » et cliquez sur play. L'image est-elle affichée correctement ?

- Paramètres à appliquer au Canvas (composant Canvas Scaler) :
  - UI Scale Mode : Scale with screen size
  - Reference Resolution : résolution de l'image (1920x1080)
- Testez de nouveau différents ratios d'affichage.
- Ajoutez un composant « Aspect Ratio Fitter » à l'image, sélectionnez le mode « Envelope Parent » et testez à nouveau.

Pour manipuler les objets de votre interface graphique, vous pouvez utiliser l'outil « Rect Tool » :

Texte

- Créez un objet de type « UI / Text » et importez les assets nécessaires
- Créez un titre pour le menu.
- Utilisez l'option « Anchor » ainsi que les coordonnées X et Y pour positionner le titre au milieu et vers le haut de l'écran.

Vous pouvez également utiliser le type « Text – TextMeshPro » qui offre plus d'options et un meilleur rendu. Il faut pour cela importer des assets supplémentaires dans le projet (proposés automatiquement par Unity).

- Sélectionnez votre texte dans la scène, et activer le mode « Pivot » :

- Le point pivot de votre objet apparaît. Déplacez-le à l'aide du « Rect Tool » et observez les paramètres du composant « Rect Transform » : comment les coordonnées évoluent-elles ? Appliquez à votre texte une rotation autour de l'axe Z, et testez avec différentes positions du point pivot.

Bouton

- Importez les images « Bouton_normal », « Bouton_highlighted » et « Bouton_pressed », et modifiez leurs paramètres d'import en « Sprite (2D and UI) ».
- Créez un objet de type « UI / Button ».
- Utilisez les paramètres du composant « Rect Transform » pour positionner le bouton au centre de l'écran.
- Appliquez l'image « Bouton_normal » au bouton.
- Composant « Button » : sélectionnez le mode de transition « Sprite Swap ».
- Appliquez les deux autres images aux états « highlighted » et « pressed ».
- Testez en cliquant sur play.

Comme pour le texte, vous pouvez ici utiliser un objet de type « Button – TextMeshPro ».

Pour personnaliser l'aspect de votre texte et de vos boutons, vous pouvez ajouter une police de caractère à votre projet (format .ttf ou .otf) et l'appliquer aux objets que vous souhaitez.

Evènement

- Créez un objet vide dans la scène et appelez-le « UIManager ». Ajoutez-y un nouveau script appelé « UIManager ».
- Créez dans ce script une méthode publique. Cette méthode doit charger la scène principale de l'application :

```
SceneManager.LoadScene
```

- Dans les paramètres du bouton, dans la boîte « On Click() », faites glisser l'objet UIManager et sélectionnez la méthode publique que vous venez de créer.
- Testez en cliquant sur play.

Pour faire un build, vous devez ajouter la scène « Menu » dans les options de build et la placer en premier pour qu'elle soit chargée au lancement de l'application.

Interface graphique dans le jeu

- Retournez dans la scène principale.
- Ajoutez un élément d'interface graphique en bas à gauche de l'écran affichant en temps réel le nombre de balles tirées par le joueur.

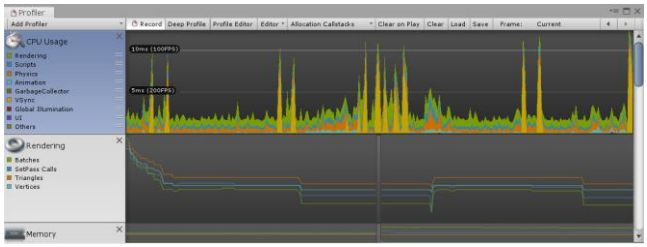
2. Post-processing

- Installez le package « Post Processing » via le package manager (Window -> Package Manager).
- Ajoutez un composant « Post Process Layer » à la caméra de la scène. Sélectionnez le layer de votre environnement pour y appliquer le post-processing (cela devrait être « Environment », vérifiez que les objets du décor soient bien dans ce layer).
- Activez l'anti-aliasing.
- Testez en cliquant sur play. Voyez-vous une différence ?
- Ajoutez un objet « Post-process Volume » dans la scène et placez-le dans le même layer que les objets du décor. Sélectionnez l'option « Is Global ».
- Créez un nouveau profil et testez différents effets.
- Vous pouvez voir un aperçu des effets grâce à la fenêtre « Game ».
- Quels effets rendent le mieux sur votre scène ?
- Voyez-vous des impacts sur les performances ?

3. Optimisation

Profiler

- Ouvrez la fenêtre **Profiler** (Window -> Analysis -> Profiler).
  Cet outil est très pratique pour analyser une application et voir ce qui prend le plus de temps à l'exécution.



  Exécutez votre application et observez le profiler. Qu'est-ce qui prend le plus de temps ?

Occlusion Culling

- Avant d'appliquer l'occlusion culling, vous devez marquer les élements du décor comme statiques pour les occultations. Marquez tous les éléments du décor comme « occludee static ». Marquez les objets pouvant en masquer d'autres comme « occluder static ». Par exemple :
  - Les différentes portions de couloir (tous les objets s'appelant Corridor_L et Corridor_T)
  - Les objets du toit (floor_6). Pour ces objets, appliquez l'opération au prefab pour vous assurer qu'elle est bien appliquée à toutes ses instances.
  - Les fenêtres occultées (big_screen). Là encore, appliquez l'opération au prefab.
- Ouvrez la fenêtre **Occlusion Culling** (Window -> Rendering -> Occlusion Culling).
- Lancez le calcul (bouton « bake »).
- Sélectionnez l'onglet « Visualization » et observez votre scène. Que se passe-t-il si vous déplacez votre caméra ?