**Unity vs Unreal**

1. Installation

| Unity | Unreal |
| --- | --- |
| * Installation relativement simple et rapide. * Différentes plateformes | * Processus d'installation relativement simple. * Différentes versions à télécharger |
|  | * Très longue * Peut nécessiter une configuration matérielle plus puissante. * Différentes versions à télécharger |

1. Multi-Projections

| Unity | Unreal |
| --- | --- |
| * Supporte la projection multi-écrans et VR. * Possibilité de gérer les affichages multiples et de stéréo. | * Prise en charge native de la projection multi-écrans et de la VR. * Peut gérer des configurations multi-projections complexes. |
| * Peut nécessiter des optimisations pour de meilleures performances. | * Peut nécessiter une optimisation pour les performances maximales. |

1. Utilisateur

| Unity | Unreal |
| --- | --- |
| * Semble rapide et fluide sur des petits projets en tout cas | * Arrive plus facilement à gérer des scènes complexes |
| * Peut être moins performant dans des scènes complexes. | * Nécessite pas mal de performances * Les premiers lancements m’ont l’air longs mais s’améliore avec le temps |

1. Programmeur

| Unity | Unreal |
| --- | --- |
| * Programmation accessible aux débutants. * Documentation complète et support actif. * Forte compatibilité matérielle et logicielle. * Pas mal de templates pour créer différents styles de projets * Peut télécharger des assets sur l’asset store * Peut ajouter des plugins | * Programmation basée sur le langage de script visuel Blueprint. * Pas besoin de savoir coder en C++ forcément * Accès complet au code source du moteur. * Grande communauté et support actif. * Pas mal de templates pour créer différents styles de projets * Peut télécharger facilement des assets avec Quixel Bridge * Ajout facile de plugin ou d'éléments avec Epic Games |
| * Plusieurs approches de programmation sont disponibles, ce qui peut rendre le code source difficile à suivre. | * Programmation basée sur le Blueprint peut être limitée pour des fonctionnalités très spécifiques. |

Interface quasi similaire dans l’idée. Dans les deux cas, conventions à connaître pour ranger nos objets, matériaux, assets… Principe facile de création de controller grâce à l’héritage des classes des acteurs par exemple. Pour créer des niveaux et les gérer, cela semble assez simple avec le Modelling Mode. Peut-être plus d’options ?

1. Les trucs en plus

| Unity | Unreal |
| --- | --- |
| * Grande communauté d'utilisateurs et ressources disponibles. | * Puissante suite d'outils d'édition de niveau et de création de contenu. * Prise en charge des effets spéciaux et des simulations avancées. |
| * On ne peut pas changer le Flyout Navigation Viewport, il faut script | * Installer le bon driver pour contourner le warning (GeForce Experience pour reconnaître celui qu’il faut et le télécharger directement) |
| * Toutes les tuiles se chargent bien et les rendus de haut sont bons mais dès qu’on s’approche… pas fou * arbres : aléatoire | * Pour Cesium for Unreal : installer 5.1 * Certaines tuiles ne s’actualisent pas mais les rendus ont l’air plus propres * Les lumières incroyable |