

Formation & Certification • Unity Certified User – VR Developer

# Unity Certified User – VR Developer

Cette formation prépare à la certification Unity Certified User – VR Developer en validant les bases de la création d'expériences VR : concepts clés, construction de scènes, interactions immersives, scripting C#, tests et optimisation.

Distributeur officiel Certiport

Centre d'examen Certiport

Learn • Practice • Certify

<b>Durée</b> 30 h (recommandé)	<b>Examen</b> UCU- VR Developer
<b>Modalité</b> Distanciel	<b>Niveau</b> Développeur VR junior

**INSCRIPTION / RÉSERVATION**



Je m'inscris maintenant

- **Learn** : Notions essentielles de VR, outils Unity XR, conception de scènes, UX et locomotion, scripting et tests.
- **Practice** : Ateliers Unity pour créer et optimiser des environnements VR avec interactions et débogage.
- **Certify** : préparation structurée à l'examen Unity Certified User – VR Developer et passage de la certification dans notre centre Certiport.

## OBJECTIFS PÉDAGOGIQUES

- Comprendre les concepts clés de la VR et leur gestion dans Unity.
- Créer et organiser des scènes VR complètes avec environnement et navigation.
- Mettre en œuvre une UX VR confortable avec interactions et feedback.
- Programmer, tester, déboguer et optimiser des expériences VR.

## PUBLIC CIBLE

- Pour les débutants et étudiants souhaitant se spécialiser en VR avec Unity.
- Pour les développeurs voulant certifier leurs compétences VR.
- Pour les enseignants intégrant Unity et la VR dans leurs cours.

## PRÉREQUIS

- Bases d'Unity et notions élémentaires de C# recommandées.
- Compréhension générale de la 3D et de la navigation virtuelle.
- Capacité à suivre un cours technique en anglais avec support en français.

## PROGRAMME DE LA FORMATION – DÉTAILLÉ

### **Concepts Unity de base pour le développement VR**

#### **Concepts essentiels de la VR**

- Définir la vision stéréoscopique et le rendu pour deux yeux ; comprendre comment la VR se différencie des autres formes de XR (AR, MR).
- Identifier les méthodes de tracking (suivi de la tête, des mains / contrôleurs, room-scale, inside-out, outside-in) et les méthodes d'entrée VR (contrôleurs, mains, regard, etc.).

#### **Utilisation du Package Manager pour la XR**

- Utiliser le Package Manager pour gérer les packages XR, notamment le XR Interaction Toolkit, les dépendances et les mises à jour.

#### **Importer et modifier des assets**

- Importer des assets 3D, textures, sons dans Unity ; créer et modifier des Prefabs pour structurer la scène VR et réutiliser des objets.

#### **Manipuler le composant Transform**

- À partir d'un scénario, utiliser le composant Transform pour positionner, faire pivoter (rotate) et mettre à l'échelle (scale) un objet dans la scène VR.

#### **Identifier les fenêtres de l'interface Unity adaptées à une tâche**

- Identifier la fenêtre principale à utiliser (Scene, Game, Hierarchy, Project, Inspector) pour une action donnée dans l'espace de travail par défaut.

#### **Gérer les composants dans la fenêtre Inspector**

- Ajouter, configurer, activer/désactiver et supprimer des composants via l'Inspector ; comprendre l'impact des paramètres sur le comportement en VR.

### **Construction d'une scène pour la VR**

#### **Techniques de pré-planification**

- Utiliser des techniques de préproduction comme le greyboxing (blocs simples pour poser les volumes) et le dimensionnement proportionnel du niveau (proportional level scaling) pour anticiper l'expérience VR.

#### **Mise en place de l'environnement avec des objets 3D**

- Utiliser des assets finalisés pour construire l'environnement VR ; organiser les GameObjects, la hiérarchie et les Prefabs pour une scène claire et performante.

## PROGRAMME DE LA FORMATION – DÉTAILLÉ

### **Éclairage en VR**

- Identifier les différents types de lumières (directionnelle, ponctuelle, spot, zone) et choisir entre éclairage Baked et Realtime en fonction des besoins de performance VR.

### **Mise en œuvre de l'UX pour la VR**

#### **Créer une interface utilisateur (UI) en World Space**

- Créer une UI en World Space pour une scène VR : utilisation de Canvas, Button, Image, Text et de l'Event System pour des interactions naturelles dans l'espace.

#### **Manipulation physique d'objets en VR**

- À partir d'un scénario, déterminer les composants nécessaires pour permettre à l'utilisateur de saisir et manipuler des objets (Colliders, XR Grab Interactable, Rigidbodies, etc.).

#### **Types de locomotion du joueur**

- Identifier les types de locomotion : degrés de liberté, déplacement d'avatar, mouvement sur 3 axes, rotation sur 6 axes, rotation sur un axe, téléportation, etc.

#### **Interactions VR et santé & sécurité**

- À partir d'un scénario, identifier les interactions VR optimales pour réduire le mal des transports, respecter l'ergonomie, les limites de confort et la sécurité de l'utilisateur.

#### **Audio 2D et audio spatial**

- Différencier les attributs des sources audio : 2D vs audio spatial (3D), paramètres de distance, diffusion, panoramique, etc., pour renforcer l'immersion VR.

### **Scripting avec Unity**

#### **Sélectionner du code C# de base adapté à un objectif donné**

- À partir d'un scénario, choisir le code C# approprié en fonction des propriétés, variables, méthodes, types de données de base et opérateurs binaires nécessaires.

#### **Utiliser les structures Unity de base**

- Choisir les bonnes structures de données Unity : Vector3, GameObject, Collider, Rigidbody, AudioSource pour atteindre un objectif de gameplay ou d'interaction en VR.

## PROGRAMME DE LA FORMATION – DÉTAILLÉ

### **4.3 Gestion des collisions et triggers**

- Identifier comment gérer les événements de collision et de trigger : Enter, Stay, Exit, et déclencher des comportements adaptés en VR.

### **Dépannage et playtesting**

#### **Éléments à vérifier suite à un problème détecté en test**

- À partir d'un problème découvert lors du playtest, identifier les zones à vérifier : objets marqués Static, Colliders manquants, Rigidbodies manquants, paramètres incorrects, options Is Trigger, Is Kinematic ou Use Gravity inappropriées.

#### **Types de logs dans la Console Unity**

- Identifier les différents types de messages dans la Console (logs, warnings, errors) et leur usage pour le débogage.

#### **Identification et correction des erreurs dans la Console**

- Reconnaître les erreurs typiques (références nulles, erreurs de syntaxe, marqueur de fin de ligne manquant, etc.) et décrire les étapes nécessaires pour les corriger.

#### **Optimisation d'une scène VR**

- Identifier les méthodes d'optimisation adaptées à la VR : occlusion culling côté caméra, suppression d'objets inutilisés, gestion du niveau de détail (LOD), etc.