

SYLLABUS OFFICIEL

Examen UCA Programmer

Programmation avancée
en C# avec Unity

Certification : Unity Certified Associate — Programmer

Niveau : Associate / Post-secondary | Public : Futurs Unity Programmers / Junior developers

1. Présentation de la certification

L'examen **Unity Certified Associate : Programmer (UCA Programmer)** valide vos compétences fondamentales en **programmation C# avec Unity** pour obtenir votre premier rôle professionnel de programmer Unity. Cette certification atteste de votre maîtrise du code, de l'UI, du debugging et de l'asset management nécessaire à la contribution dans une équipe de développement Unity.

La réussite de cet examen unique conduit à l'obtention de la certification **Unity Certified Associate — Programmer**, conçue pour les candidats avec 2 à 3 semestres post-secondaires de cours Unity et un portfolio diversifié. Elle démontre votre capacité à évaluer et intégrer du code dans un système existant créé par un lead programmer.

Informations clés

| | |
|------------------------------|--|
| Code de l'examen | UCA Programmer (Unity Certified Associate — Programmer) |
| Intitulé officiel | Unity Certified Associate : Programmer |
| Certification obtenue | Unity Certified Associate — Programmer |
| Éditeur officiel | Unity Technologies |
| Centre de test | Certiport (Pearson VUE) / Unity Education Store (Pearson VUE) |
| Niveau | Associate — post-secondaire / futurs professionnels |
| Version Unity ciblée | Unity 6 (version 2026) |
| Durée de l'examen | Environ 90 à 100 minutes |
| Nombre de questions | Environ 100 questions (QCM, scénarios, cas pratiques) |
| Score requis | 500 sur 700 (échelle officielle Unity : 200 à 700 points) |
| Langues disponibles | Anglais, chinois (simplifié et traditionnel), japonais, coréen, espagnol (Amérique latine) |
| Prérequis recommandé | 2 à 3 semestres post-secondaires de cours Unity ou étude indépendante équivalente. Portfolio de projets Unity diversifiés. |

| | |
|-------------------------------------|---|
| Validité de la certification | 3 ans à partir de la date de réussite (stackable, attribuée une seule fois) |
| Politique de reprise | Délai d'attente de 24 heures avant la 1re reprise (voucher retake à utiliser sous 60 jours) |
| Modalité | En centre agréé Certiport (CATC) — OpenCertif est un centre Certiport autorisé |
| Badge numérique | Badge officiel Credly délivré automatiquement après réussite |

2. Profil du candidat

En tant que candidat à l'examen UCA Programmer, vous démontrez des compétences professionnelles débutantes en programmation Unity. Vous êtes capable de :

- Évaluer et intégrer du code dans un système existant.
- Appliquer des coding standards guidés par des senior programmers.
- Implémenter des transitions entre scènes (SceneManager).
- Sauvegarder des données entre scènes et sessions : static variables, PlayerPrefs.
- Utiliser l'API Unity avec la documentation officielle.
- Choisir les data structures appropriées : lists, arrays, dictionaries.
- Distinguer héritage de base et interfaces.
- Construire une application pour WebGL ou PC.
- Créer des UI fonctionnelles avec anchors, pivots et UnityEvent.
- Débuguer avec debug messages, identifier null variables, refactorer.
- Gérer les Prefabs : nested prefabs, prefab variants.
- Utiliser un système de version control.

L'examen évalue spécifiquement quatre familles de compétences essentielles à tout Unity programmer junior :

- Unity Programming — code, API, data structures, build.
- UI — anchors, layouts, UnityEvent system.
- Debugging — logs, compilation errors, refactoring, profiling.
- Asset Management — Prefabs, version control.

3. Prérequis et public cible OpenCertif

Unity Technologies recommande aux candidats de disposer de :

- **2 à 3 semestres post-secondaires de cours Unity** ou étude indépendante équivalente.
- Expérience avec une variété de projets Unity.
- Maîtrise de l'import d'assets / code (Asset Store, Package Manager).
- Capacité à déboguer des problèmes non complexes.
- Capacité à interpréter du code pre-existant bien documenté.
- Capacité à créer, éditer et utiliser des Prefabs.
- Capacité à déployer un build basique.
- Connaissance fondamentale de C# (héritage, interfaces, collections).

Public cible OpenCertif

- Étudiants en BTS / DUT / Licence pro / Bachelor informatique ou game development.
- Auto-didactes ayant un portfolio Unity avec plusieurs projets terminés.
- Junior developers cherchant à valider leurs compétences pour leur premier emploi.
- Profils en reconversion vers le développement Unity.

- Candidats aux métiers de Junior Unity Developer, Junior Gameplay Programmer.
- Profils ayant passé l'UCU Programmer et souhaitant monter en niveau.

4. Domaines de compétences mesurées

L'examen est structuré autour de 4 grands domaines de compétences. Le tableau ci-dessous indique le poids relatif de chaque domaine dans l'évaluation finale (version 2026 — aligné sur Unity 6). Les pondérations sont des estimations issues du guide officiel Unity / Certiport.

| Domaine | Intitulé | Pondération |
|---------|---------------------|-------------|
| 1 | Unity Programming | 40 — 45 % |
| 2 | UI (User Interface) | 15 — 20 % |
| 3 | Debugging | 20 — 25 % |
| 4 | Asset Management | 15 — 20 % |

Remarque : l'examen UCU Programmer dure environ 50 minutes pour 40 questions, soit environ 1 minute 15 par question. La gestion du temps est essentielle. Le score requis pour valider est de **500 sur 700** (sur une échelle officielle Unity de 200 à 700 points).

5. Détail des compétences mesurées

Cette section détaille de manière exhaustive l'ensemble des compétences couvertes par l'examen UCA Programmer, en s'appuyant sur les Objective Domains publiés par Certiport et Unity Technologies (version 2026 — aligné sur Unity 6).

1 Unity Programming

40 — 45 %

1.1 Intégration et coding standards

- ▶ Évaluer du code pour intégration dans un système existant créé par un lead.
- ▶ Appliquer des coding standards et best practices guidés par des senior programmers.
- ▶ Déterminer le code qui accomplit une interaction spécifiée.

1.2 Scene management et persistance

- ▶ Déterminer le processus pour implémenter des transitions entre scènes.
- ▶ Utiliser SceneManager.LoadScene, LoadSceneAsync.
- ▶ Sauvegarder des données entre scènes et sessions : **static variables**, **PlayerPrefs**.
- ▶ Comprendre le DontDestroyOnLoad pattern.

1.3 API Unity et documentation

- ▶ Obtenir un résultat défini en utilisant les méthodes Unity API (avec la doc).
- ▶ Sélectionner les propriétés, scripts et composants appropriés pour une tâche.
- ▶ Naviguer la Unity Scripting Reference.

1.4 Héritage et interfaces

- ▶ Expliquer les différences entre **basic inheritance** et **interfaces**.
- ▶ Implémenter une classe abstraite vs une interface.
- ▶ Comprendre le polymorphisme et l'encapsulation.

1.5 Data structures et types

- ▶ Choisir les data structures appropriées : **lists, arrays, dictionaries**.
- ▶ Choisir les data types appropriés : floats, bools, strings.
- ▶ Comprendre les collections génériques C# (List<T>, Dictionary<TKey,TValue>).

1.6 Build et publication

- ▶ Construire une application pour WebGL.
- ▶ Construire une application pour Windows / macOS / Linux PC.
- ▶ Configurer les Player Settings.

2 UI (User Interface)

15 — 20
%

2.1 Layout et positionnement UI

- ▶ Arranger des composants UI sur le canvas selon un layout défini.
- ▶ Utiliser correctement **anchors, pivots** et **groups**.
- ▶ Comprendre les Layout Groups : Horizontal, Vertical, Grid.

2.2 Affichage de données

- ▶ Identifier le processus pour afficher des données dans divers éléments UI.
- ▶ Utiliser Text, TextMeshPro, Slider, Image, Toggle.
- ▶ Mettre à jour dynamiquement les UI depuis le code.

2.3 UnityEvent system

- ▶ Expliquer comment utiliser le **UnityEvent system** pour répondre au User Input.
- ▶ Configurer les onClick et autres callbacks UI.
- ▶ Créer des UnityEvents custom dans le code.

3 Debugging

20 — 25
%

3.1 Debug messages

- ▶ Programmer des debug messages pour identifier les causes de code qui ne s'exécute pas comme attendu.
- ▶ Utiliser Debug.Log, Debug.LogWarning, Debug.LogError.
- ▶ Utiliser Debug.DrawLine et Debug.DrawRay pour la visualisation.

3.2 Erreurs de compilation

- ▶ Identifier la cause d'une erreur de compilation depuis un bloc de code.
- ▶ Reconnaître les erreurs de syntaxe C# courantes.
- ▶ Corriger les missing semicolons, brackets, types.

3.3 Null references

- ▶ Identifier les erreurs causées par une variable null.
- ▶ Prévenir les NullReferenceException avec null checks.
- ▶ Utiliser l'opérateur null-conditional (?.) et null-coalescing (??).

3.4 Refactoring

- ▶ Identifier les techniques pour refactor du code selon les coding standards.
- ▶ Renommer variables et méthodes pour la clarté.
- ▶ Extraire des méthodes pour éviter la duplication.

3.5 Profiling

- ▶ Sélectionner les outils de profiling appropriés.
- ▶ Utiliser le Profiler Window de Unity.
- ▶ Utiliser le Frame Debugger pour le rendering.
- ▶ Identifier les sources de problèmes de performance.

4 Asset Management

15 — 20
%

4.1 Prefabs

- ▶ Expliquer comment utiliser les Prefabs dans une scène.
- ▶ Créer des Prefab instances et répliquer des objets.
- ▶ Override des propriétés sur les instances vs Prefab Asset.

4.2 Nested Prefabs et Prefab Variants

- ▶ Décrire le processus et les outcomes pour changer un **nested prefab** ou **prefab variant**.
- ▶ Comprendre la chaîne d'héritage des variants.
- ▶ Gérer les overrides et reverts.

4.3 Version control

- ▶ Expliquer les principaux usages du **version control** en Unity.
- ▶ Connaître Git et Unity Version Control (Plastic SCM).
- ▶ Configurer un .gitignore Unity propre.
- ▶ Comprendre Asset Serialization pour le diff Unity.

6. Modalités pédagogiques OpenCertif

OpenCertif accompagne les candidats au UCA Programmer à travers un parcours blended-learning complet, combinant ressources e-learning interactives, projets pratiques en C#, Unity API, PlayerPrefs, scene management, UI Toolkit, debugging, profiling et Prefab variants et accompagnement tutoré.

Format de la formation

| | |
|-----------------------------|---|
| Durée recommandée | 180 à 250 heures (OpenCertif structure ce parcours sur 90 à 120 heures de formation tutorée complétées par 90 à 130 heures de projet personnel) |
| Modalité | 100 % distanciel asynchrone, ou blended (distanciel + classes virtuelles) |
| Support pédagogique | Unity Certified User Courseware officiel (GMetrix) + ressources OpenCertif (modules Rise 360, scénarios immersifs) |
| Plateforme LMS | Imsopencertif.fr (Moodle) — accès 24/7 pendant 12 mois |
| Encadrement | Tutorat asynchrone par expert Unity certifié + classes virtuelles bimensuelles |
| Pratique requise | Au moins 150 heures de pratique Unity (recommandation officielle Unity Technologies) |
| Évaluations | Quiz formatifs par module, 3 projets pratiques Unity, examens blancs CertPREP |
| Certification finale | Passage de l'examen UCA Programmer en centre OpenCertif (CATC Certiport) |

Parcours d'apprentissage proposé

- **Module 1** : Junior Programmer Pathway — fondamentaux C# Unity.
- **Module 2** : Coding standards et best practices.
- **Module 3** : Scene management et transitions.
- **Module 4** : Persistance — PlayerPrefs, static, ScriptableObjects.
- **Module 5** : Héritage, interfaces et polymorphisme.
- **Module 6** : Data structures C# — List, Dictionary, Queue.
- **Module 7** : API Unity — navigation de la documentation.
- **Module 8** : UI — Canvas, Anchors, RectTransform.
- **Module 9** : UnityEvent system.

- **Module 10** : Debug.Log et debugging stratégique.
- **Module 11** : Erreurs de compilation et null references.
- **Module 12** : Refactoring et clean code.
- **Module 13** : Profiling — Profiler, Frame Debugger.
- **Module 14** : Prefabs avancés — nested, variants.
- **Module 15** : Version control — Git et Unity VCS.
- **Module 16** : Build — WebGL, PC, multi-plateforme.
- **Module 17** : Mini-projet final — jeu complet avec persistance et UI.
- **Module 18** : Examen blanc CertPREP et préparation finale.

7. Ressources d'étude officielles

En complément du parcours OpenCertif, les ressources officielles Unity Technologies et Certiport suivantes sont fortement recommandées :

- Unity Learn (learn.unity.com) — Pathway Junior Programmer.
- Unity Scripting API (docs.unity3d.com/ScriptReference).
- Documentation officielle C# Microsoft (learn.microsoft.com/dotnet/csharp).
- UCA Programmer Courseware (Unity Education Store / Pearson VUE).
- Official Practice Test UCA Programmer (Unity Education Store).
- PDF officiel : [Unity_Certified_Associate_Programmer_Exam_Objectives_document](#).
- Unity Discussions et communauté.
- Chaîne YouTube officielle Unity — tutoriels programming.
- Page Certiport : certiport.pearsonvue.com/Certifications/Unity.
- Page officielle Unity : unity.com/products/unity-certifications/associate-programmer.
- Badge officiel délivré via Credly (credly.com).
- Pages OpenCertif dédiées : opencertif.fr/unity.

8. Modalités de passage de l'examen

| | |
|-------------------------------------|---|
| Inscription | Via OpenCertif ou directement auprès d'un centre Certiport |
| Centre d'examen | OpenCertif — Centre Certiport Authorized Testing Center (CATC) / Pearson VUE |
| Mode de passage | En centre uniquement (Unity n'autorise pas l'examen OnVUE à distance pour les certifications UCU — présence sur site requise) |
| Pièce d'identité | 1 pièce d'identité avec photo obligatoire le jour de l'examen (pour les mineurs : autorisation parentale et CNI / passeport) |
| Aménagements | Demande possible auprès de Certiport (temps additionnel, assistance technique) |
| Résultat | Score communiqué immédiatement à la fin de l'examen (échelle 200-700, seuil de réussite 500) |
| Validité de la certification | 3 ans à partir de la date de réussite — attribuée une seule fois (stackable, pas de renouvellement payant requis) |

| | |
|-----------------------------|---|
| Politique de reprise | Délai d'attente de 24 heures avant la 1re reprise. Voucher retake à utiliser sous 60 jours après l'échec. |
| Badge numérique | Badge officiel délivré via Credly et intégrable à LinkedIn, CV, portfolio, sites de recrutement |

9. Contact et inscription

Pour toute information complémentaire, demande de devis ou inscription à la formation préparatoire au UCA Programmer, l'équipe OpenCertif reste à votre disposition. OpenCertif est un Centre Certiport Authorized Testing Center (CATC) habilité à délivrer les certifications Unity Certified User.



10. Mentions légales et version

Ce syllabus est établi par OpenCertif sur la base des Objective Domains officiels publiés par Certiport pour la certification UCA Programmer, dans sa version applicable (version 2026 — aligné sur Unity 6). Les compétences mesurées, les pondérations et les objectifs présentés reflètent fidèlement la structure de l'examen telle que publiée par Unity Technologies via Certiport.

Unity, le logo Unity, Unity Editor, Unity Hub, Unity Learn et Unity Certified Associate sont des marques déposées de Unity Technologies. C# et .NET sont des marques déposées de Microsoft Corporation. Visual Studio est une marque déposée de Microsoft Corporation. Git est une marque déposée de Software Freedom Conservancy. Plastic SCM est une marque déposée de Codice Software / Unity Technologies. Certiport et CertPREP sont des marques déposées de Pearson Education Inc. Pearson VUE est une marque déposée de Pearson Education Inc. Credly est une marque déposée de Pearson Education Inc.

OpenCertif n'est pas affilié à Unity Technologies. Ce document est fourni à titre informatif. Pour la version officielle et à jour des Objective Domains, consulter certiport.pearsonvue.com/Certifications/Unity et unity.com/products/unity-certifications.

Version du syllabus : 2026.05 — Édition mai 2026

Source officielle Certiport : certiport.pearsonvue.com/Certifications/Unity/Certified-User/Certify

Source officielle Unity : unity.com/products/unity-certifications/user-programmer

Page OpenCertif : opencertif.fr/unity-user-programmer