

SYLLABUS OFFICIEL

Examen UCP Programmer

Développement avancé
en C# et Unity (Professionnel)

Certification : Unity Certified Professional — Programmer

Niveau : Professional / Senior | Public : Unity Developers expérimentés / Gameplay
Programmers

1. Présentation de la certification

L'examen **Unity Certified Professional : Programmer (UCP Programmer)** valide les compétences avancées nécessaires pour contribuer à l'**exécution technique d'un projet Unity professionnel** de la conception au lancement et au-delà. Cette certification atteste de votre capacité à coder des solutions à des problèmes complexes, scripter les interactions utilisateur, implémenter de la logique d'état applicative, simuler la physique, déboguer et optimiser les performances.

La réussite de cet examen unique conduit à l'obtention de la certification **Unity Certified Professional — Programmer**, conçue pour les professionnels en poste depuis **2 ans ou plus** sur des rôles tels que Unity Developer, Software Engineer, Software Developer, Mobile Application Developer ou Gameplay Programmer. Elle démontre que vous êtes prêt pour votre prochain challenge professionnel.

Informations clés

Code de l'examen	UCP Programmer (Unity Certified Professional — Programmer)
Intitulé officiel	Unity Certified Professional : Programmer
Certification obtenue	Unity Certified Professional — Programmer
Éditeur officiel	Unity Technologies
Centre de test	Unity Education Store (Pearson VUE) / Certiport
Niveau	Professional — 2 à 4 ans d'expérience professionnelle requise
Version Unity ciblée	Unity 2022 LTS (PDF officiel à jour : Unity 2022 LTS)
Durée de l'examen	Environ 90 à 100 minutes
Nombre de questions	Environ 100 questions (QCM, scénarios complexes, cas pratiques)
Score requis	Indicatif 500 / 700 (échelle officielle Unity : 200 à 700 points)
Langues disponibles	Anglais et japonais

Prérequis recommandé	2 ans ou plus d'expérience pratique professionnelle en développement / création artistique 3D temps réel avec Unity
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) ou à distance OnVUE — 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 UCP Programmer, vous êtes un professionnel expérimenté capable de :

- Architecturer des solutions techniques pour des projets Unity complexes.
- Coder en C# au niveau senior : design patterns, async/await, generics avancés.
- Scripter les interactions utilisateur (gameplay, UI, input avancé).
- Implémenter la logique d'état applicative (state machines, FSM).
- Simuler la physique : Rigidbody avancé, joints, soft bodies, raycasts complexes.
- Optimiser les performances : profiling, batching, draw calls, memory management.
- Travailler dans un pipeline professionnel : version control, unit tests, code reviews.
- Utiliser les Unity Services : Cloud Build, multiplayer, monetization, analytics.
- Comprendre et appliquer les mathématiques 3D : algèbre linéaire, matrices, quaternions.
- Travailler avec l'art pipeline : import workflow, materials, shaders custom.
- Cycle de vie complet : concept, dev, QA, release, post-launch.

L'examen évalue spécifiquement sept familles de compétences essentielles à tout Unity Programmer professionnel :

- Programming core interactions — gameplay, input, state.
- Working in the art pipeline — collaboration avec artistes.
- Developing application systems — architecture, save, services.
- Programming for scene and environment design — spawning, navigation.
- Optimization of performance and platforms — profiling, mobile, console.
- Working in software development teams — version control, unit testing.
- Common C# programming constructs — patterns, generics, async.

3. Prérequis et public cible OpenCertif

Unity Technologies recommande aux candidats de disposer de :

- **2 ans ou plus d'expérience pratique** en programmation de jeux ou de 3D interactif avec Unity.
- **2 ans ou plus d'expérience pratique** en programmation, y compris C#.
- Expérience du cycle de développement logiciel complet, du concept à la livraison.
- Compréhension des applications professionnelles de Unity : game dev, interactive entertainment, design visualization.
- Compréhension basique du pipeline visual / 3D asset et animation dans Unity, y compris setups character et environment.
- Compréhension des pratiques professionnelles de team software development, y compris **unit testing** et **version control**.
- Connaissance des Unity Services : collaboration, monetization, live operations, multiplayer.

- Compréhension des mathématiques critiques au 3D interactif : **algèbre linéaire** et **opérations matricielles**.

Public cible OpenCertif

- Unity Developers en poste depuis 2+ ans en studio AAA, indie ou serious games.
- Software Engineers / Software Developers travaillant sur Unity.
- Mobile Application Developers utilisant Unity pour mobile games / apps.
- Gameplay Programmers et Systems Programmers.
- Profils ayant obtenu la certification UCA Programmer ou équivalent.
- Freelances Unity confirmés souhaitant valider officiellement leur niveau.
- Lead Developers et Tech Leads en environnement Unity professionnel.

4. Domaines de compétences mesurées

L'examen est structuré autour de 7 grands domaines de compétences. Le tableau ci-dessous indique le poids relatif de chaque domaine dans l'évaluation finale (version 2026 — aligné sur Unity 2022 LTS). Les pondérations sont des estimations issues du guide officiel Unity / Certiport.

Domaine	Intitulé	Pondération
1	Programming core interactions	15 — 20 %
2	Working in the art pipeline	10 — 15 %
3	Developing application systems	15 — 20 %
4	Programming for scene and environment design	10 — 15 %
5	Optimization of performance and platforms	15 — 20 %
6	Working in software development teams	10 — 15 %
7	Common C# programming constructs	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 UCP Programmer, en s'appuyant sur les Objective Domains publiés par Certiport et Unity Technologies (version 2026 — aligné sur Unity 2022 LTS).

1 Programming core interactions

15 — 20 %

1.1 Input et gameplay

- ▶ Gérer le nouveau Input System et le Legacy Input.
- ▶ Implémenter le mapping multi-plateforme (clavier, gamepad, mobile, VR).
- ▶ Créer des Action Maps complexes avec Input System.
- ▶ Gérer le débouncing et le throttling des inputs.

1.2 Physique et collisions avancées

- ▶ Configurer Rigidbody avec Continuous Collision Detection.
- ▶ Utiliser les joints : Hinge, Fixed, Spring, Configurable Joint.
- ▶ Effectuer des raycasts complexes : SphereCast, BoxCast, CapsuleCast.
- ▶ Paramétrer Physics Materials et friction custom.
- ▶ Comprendre Physics 2D vs Physics 3D.

1.3 State management

- ▶ Implémenter des Finite State Machines (FSM) en code.
- ▶ Utiliser Animator State Machines pour la logique d'état.
- ▶ Gérer les transitions et conditions d'état.
- ▶ Pattern State design en C#.

2 Working in the art pipeline

10 — 15
%

2.1 Collaboration avec les artistes

- ▶ Comprendre les workflows d'import de modèles riggés.
- ▶ Gérer les animations clips et animation events.
- ▶ Connaître les Avatar et humanoid rigs.
- ▶ Paramétrer les import settings pour les besoins du projet.

2.2 Materials et shaders custom

- ▶ Créer des shaders custom en HLSL ou Shader Graph.
- ▶ Communiquer avec les artistes sur les material parameters.
- ▶ Exposer des properties dans l'Inspector pour les artistes.

2.3 Animation programming

- ▶ Piloter Animator par code : SetBool, SetFloat, SetTrigger.
- ▶ Gérer Animation Events callbacks.
- ▶ Utiliser Playables API pour l'animation runtime.
- ▶ Comprendre IK (Inverse Kinematics) et FK (Forward Kinematics).

3 Developing application systems

15 — 20
%

3.1 Save / Load et persistance

- ▶ Implémenter des systèmes de sauvegarde robustes (JSON, binary).
- ▶ Utiliser PlayerPrefs pour les paramètres utilisateur.
- ▶ Gérer la sérialisation custom.
- ▶ Implémenter le cloud save (Unity Cloud Save).

3.2 Scene management avancé

- ▶ Charger des scènes additivement (LoadSceneAsync, Additive).
- ▶ Gérer multiple scenes dans un même projet.
- ▶ Persister des données entre scènes (DontDestroyOnLoad).
- ▶ Streaming de niveaux pour des mondes ouverts.

3.3 Unity Services intégration

- ▶ Intégrer Unity Authentication.
- ▶ Utiliser Unity Cloud Save / Cloud Code.
- ▶ Configurer Unity Analytics.
- ▶ Implémenter Unity IAP (In-App Purchases).
- ▶ Gérer Unity Ads (LevelPlay / Vector AI).

3.4 Networking et multiplayer

- ▶ Comprendre Netcode for GameObjects (NGO).
- ▶ Implémenter de la synchronisation client/server.
- ▶ Gérer Relay et Lobby pour le matchmaking.

4 Programming for scene and environment design

10 — 15
%

4.1 Spawn et instantiation

- ▶ Implémenter Object Pooling pour optimiser l'instantiation.
- ▶ Gérer le spawn dynamique d'entités.
- ▶ Utiliser des spawn points et règles de placement.

4.2 Navigation et IA

- ▶ Configurer NavMesh runtime avec NavMeshSurface.
- ▶ Utiliser NavMeshAgent et NavMeshObstacle.
- ▶ Créer des Off-Mesh Links scripted.
- ▶ Implémenter des comportements d'IA avec Behavior Trees ou FSM.

4.3 Procedural generation (notions)

- ▶ Créer du contenu procedural simple.
- ▶ Utiliser Perlin Noise pour terrain procedural.
- ▶ Gérer la seed-based generation pour la reproductibilité.

5 Optimization of performance and platforms

15 — 20
%

5.1 Profiling avancé

- ▶ Utiliser Unity Profiler : CPU, GPU, Memory, Rendering.
- ▶ Utiliser Frame Debugger pour optimiser les draw calls.
- ▶ Identifier les bottlenecks et les solutions.
- ▶ Memory Profiler pour traquer les fuites mémoire.

5.2 Optimisation rendering

- ▶ Optimiser les draw calls : Static / Dynamic Batching, GPU Instancing.
- ▶ Utiliser LOD (Level of Detail) approprié.
- ▶ Occlusion Culling et Frustum Culling.
- ▶ Texture compression et atlas.
- ▶ Shader complexity et perf shaders.

5.3 Memory management

- ▶ Gérer le Garbage Collector et réduire les allocations.
- ▶ Object Pooling pour éviter le GC.
- ▶ Utiliser les Burst Compiler et Jobs System.
- ▶ DOTS / ECS pour les performances extrêmes (notions).

5.4 Optimisation multi-plateforme

- ▶ Adapter les graphics settings par plateforme (Mobile, Console, PC, WebGL).
- ▶ Profiler sur device réel (mobile, casque VR).
- ▶ Gérer les Quality Settings et Render Pipelines.
- ▶ Optimiser les Asset Bundles et Addressables.

6 Working in software development teams

10 — 15
%

6.1 Version control

- ▶ Utiliser **Git** avec un projet Unity (.gitignore, .gitattributes).
- ▶ Configurer **Unity Version Control** (Plastic SCM).
- ▶ Paramétrer Asset Serialization en Force Text.
- ▶ Gérer les merges sur scènes et prefabs.

6.2 Unit testing

- ▶ Écrire des tests unitaires avec Unity Test Framework.
- ▶ Créer des tests PlayMode et EditMode.
- ▶ Mocker des dépendances.
- ▶ Intégrer les tests dans CI/CD.

6.3 Code reviews et standards

- ▶ Suivre des coding standards de team.
- ▶ Effectuer des code reviews constructives.
- ▶ Documenter le code (XML doc comments).
- ▶ Utiliser des linters et formatters.

6.4 CI/CD avec Unity

- ▶ Configurer Unity Cloud Build.
- ▶ Mettre en place des pipelines CI avec GitHub Actions ou GitLab CI.
- ▶ Automatiser les builds multi-plateformes.
- ▶ Distribuer les builds (TestFlight, App Center).

7 Common C# programming constructs

15 — 20
%

7.1 OOP avancé

- ▶ Maîtriser héritage multiple via interfaces.
- ▶ Utiliser les abstract classes et virtual methods.
- ▶ Comprendre SOLID principes.
- ▶ Encapsulation avec properties et auto-properties.

7.2 Generics et collections

- ▶ Créer des classes et méthodes génériques (T, U, K, V).
- ▶ Utiliser les contraintes de type (where T : class, struct, new()).
- ▶ Maîtriser System.Collections.Generic : List, Dictionary, HashSet, Queue, Stack.
- ▶ Utiliser LINQ pour filtrer, transformer et agréger des collections.

7.3 Async et concurrence

- ▶ Utiliser async / await pour les opérations asynchrones.
- ▶ Comprendre les Task et CancellationToken.
- ▶ Différence entre Coroutines et async/await.
- ▶ Gérer la concurrence avec lock et ConcurrentDictionary.

7.4 Design patterns courants

- ▶ Implémenter Singleton, Observer, Factory, Strategy.
- ▶ Utiliser Object Pool, Command, MVC/MVVM.
- ▶ Event-driven architecture.
- ▶ Dependency Injection (notions).

7.5 Mathématiques 3D

- ▶ Manipuler Vector3, Vector2 et les opérations vectorielles.
- ▶ Comprendre les Quaternion et les rotations.
- ▶ Utiliser les matrices de transformation (Matrix4x4).
- ▶ Mathématiques d'algèbre linéaire pour le 3D.

6. Modalités pédagogiques OpenCertif

OpenCertif accompagne les candidats au UCP Programmer à travers un parcours blended-learning complet, combinant ressources e-learning interactives, projets pratiques en C# avancé, Unity API, ECS/DOTS, optimisation, mathématiques 3D, version control et services Unity et accompagnement tutoré.

Format de la formation

Durée recommandée	300 à 400 heures de pratique Unity professionnelle (pas une formation OpenCertif standard — cette certification s’adresse à des professionnels déjà en poste)
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 UCP Programmer en centre OpenCertif (CATC Certiport)

Parcours d’apprentissage proposé

- **Module 1** : Architecture logicielle Unity professionnelle.
- **Module 2** : Input System avancé et rebinding runtime.
- **Module 3** : Physique avancée — joints, raycasts complexes, soft bodies.
- **Module 4** : Finite State Machines et patterns gameplay.
- **Module 5** : Art pipeline integration — rigs, animations, shaders custom.
- **Module 6** : Save / Load et persistance robuste.
- **Module 7** : Unity Services — Auth, Cloud Save, IAP, Ads, Analytics.
- **Module 8** : Networking — Netcode for GameObjects, Relay, Lobby.
- **Module 9** : Object Pooling et scene management avancé.

- **Module 10** : Navigation runtime — NavMeshSurface, Behavior Trees.
- **Module 11** : Profiling — CPU, GPU, Memory, Frame Debugger.
- **Module 12** : Optimisation rendering — batching, GPU Instancing, LOD.
- **Module 13** : Memory management et Burst Compiler.
- **Module 14** : DOTS / ECS — fondamentaux.
- **Module 15** : Version control avancé — Git, Plastic SCM, merges.
- **Module 16** : Unit testing — Test Framework, mocking, CI.
- **Module 17** : CI/CD avec Cloud Build et GitHub Actions.
- **Module 18** : C# avancé — generics, async/await, LINQ.
- **Module 19** : Design patterns et SOLID en Unity.
- **Module 20** : Mathématiques 3D — quaternions, matrices, algèbre linéaire.
- **Module 21** : 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) — pathways avancés.
- Unity Scripting API (docs.unity3d.com/ScriptReference).
- PDF officiel :
Unity_Certified_Professional_Programmer_2021_LTS_Exam_Objectives_document.
- UCP Programmer Courseware (Unity Education Store) — 15+ coding challenges, 8h de vidéo, 4 cours.
- Official Practice Test UCP Programmer (Unity Education Store).
- Documentation officielle Netcode for GameObjects.
- Documentation DOTS / ECS (Unity DOTS Stack).
- Documentation Unity Cloud Build et Cloud Services.
- Documentation Unity Test Framework.
- Microsoft Learn C# avancé (learn.microsoft.com/dotnet/csharp).
- Chaîne YouTube officielle Unity — Made with Unity, Tech Talks.
- Page officielle Unity : unity.com/products/unity-certifications/professional-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 UCP 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 UCP Programmer, dans sa version applicable (version 2026 — aligné sur Unity 2022 LTS). 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, Unity Cloud, Unity Asset Store, DOTS (Data-Oriented Technology Stack), ECS (Entity Component System), Burst Compiler, Jobs System, Netcode for GameObjects, Addressables et Unity Certified Professional sont des marques déposées de Unity Technologies. C# et .NET sont des marques déposées 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