



Certified
Programmer

測驗 目標

Unity 認證程式
設計師

角色

Unity 程式設計專家使用 Unity 來開發互動式內容。Unity 程式設計師會協同開發團隊的其他成員（例如，音訊及美術專家），使用 Unity Editor 的功能搭配由軟體開發團隊的其他成員所建立的視覺及音訊資源，讓應用程式的願景成真。Unity 程式設計師是一個通才，擅長解決困難的程式碼編寫問題，並負責協助解決各式各樣的潛在技術問題，包括整合美術資源、編寫使用者介面程式碼、編寫使用者互動和遊戲系統規則的腳本、實作應用程式狀態邏輯、模擬物理效果、偵錯程式碼及最佳化效能。

「Unity 認證程式設計師」是針對入門等級至中級程式設計師，以及要在各行各業尋求程式設計職務之大專畢業學生適用的專業人員認證。這項認證對潛在的僱主表明了認證持有人具備下列能力：

- 能在專業軟體開發過程中使用程式設計技能來建立並維護使用 Unity 建構的應用程式
- 具備技術流程能力傾向、思考以邏輯導向且能隨機應變
- 可受託獨立處理中級程式設計任務的例行作業，並能與更資深的工程師合作解決複雜的技術挑戰

此角色的職務名稱

- 遊戲程式設計師
- 軟體工程師
- 軟體開發人員
- Unity 開發人員
- 手機應用程式開發人員

先決條件

本認證是針對下列對象而設立：就讀遊戲程式設計、電腦科學或相關領域的大學應屆畢業之程式設計師；已經完成 2 年以上程式設計方面之大專同等學歷或具備工作經驗的獨立學習者；或早期職業生涯到中期職業生涯在工作環境中會使用 Unity 的專業人士。考生應試前應具備使用 Unity 進行獨力開發，或參與跨職能團隊編寫互動式應用程式的實際經驗，擁有完整的雛形設計或技術示範能力。

必備經驗：

- 2 年以上使用 Unity 設計遊戲或 3D 互動式程式之實際經驗
- 2 年以上電腦程式設計方面之實際經驗，包括 C#
- 曾親身參與從概念到完成的完整軟體開發過程
- 了解如何使用專業應用程式與 Unity 結合進行軟體開發，包括遊戲開發、互動式娛樂及設計視覺化
- 對 Unity 中的視覺/3D 資源與動畫製作流程，包括角色與環境設定，具備基本的了解
- 了解專業團隊軟體開發實務，包括單元測試及版本控制
- 熟知 Unity 服務的共同作業、創造營收、即時作業及多人遊戲特性
- 了解對 3D 互動式開發中至關重要的數學，包括線性代數和矩陣運算

注意：本項認證是針對 Unity 2017.3 版所開發。

核心技能

本工作領域的核心技能著重在解決專案從概念到完成上市以及其他方面的技術執行問題。

程式設計核心互動

- 實作及設定遊戲物件行為和物理效果
- 實作及設定輸入及控制
- 實作及設定攝影機視圖和動作

處理美術製作流程

- 了解材質、貼圖和著色器 (Shader)，並撰寫可與 Unity 的渲染 API 互動的腳本
- 了解光源，並撰寫可與 Unity 的光源 API 互動的腳本
- 了解 2D 和 3D 動畫，並撰寫可與 Unity 的動畫 API 互動的腳本
- 了解粒子系統和效果，並撰寫可與 Unity 的粒子系統 API 互動的腳本

開發應用程式系統

- 解釋應用程式介面流程腳本，例如選單系統、UI 瀏覽和應用程式設定
- 解釋使用者控制之自訂化的腳本，例如角色產生器、物品清單、商店及 APP 內購
- 使用 Unity Analytics 和 PlayerPrefs 等技術分析使用者進度功能的腳本，例如計分、關卡和遊戲內經濟
- 分析 2D 疊加圖的腳本，例如抬頭顯示器 (HUD)、小地圖和廣告
- 確認儲存及擷取應用程式及使用者資料的腳本
- 認識並評估線上多人遊戲功能的影響

場景及環境設計之程式設計

- 判斷用於實作音訊資源的腳本
- 確認用於實作 `GameObject` 實例化、摧毀及管理的方法
- 判斷使用 `Unity` 導航系統尋徑的腳本

效能及平台最佳化

- 使用工具 (例如 `Unity Profiler`) 評估錯誤及效能問題
- 確認可透過哪些最佳化措施滿足特定建構平台及 / 或硬體配置的要求
- 判斷 `XR` 平台常見的 `UI` 預設用途及最佳化

與專業軟體開發團隊合作

- 使用如 `Unity Collaborate` 之類的技術，認識與版本控制的使用和影響相關之概念
- 示範開發人員的測試知識及其對軟體開發流程的影響，包括 `Unity Profiler` 及傳統偵錯和測試技術
- 認識可實現模塊化、可讀性和可重複使用性的各種腳本建構技術

認證測驗主題

程式設計核心互動

- 實作 `GameObject` 和環境的行為及互動
- 確認實作輸入及控制的方法
- 確認實作攝影機視圖及動作的方法

處理美術製作流程

- 熟知材質、貼圖及著色器 — Unity 渲染 API
- 熟知光源 — Unity 光源 API
- 熟知 2D 及 3D 動畫 — Unity 動畫 API
- 熟知粒子系統 — Unity 粒子 API

開發應用程式系統

- 應用程式介面流程，例如選單系統、UI 瀏覽和應用程式設定
- 使用者控制之自訂化，例如角色產生器、物品清單、商店及 APP 內購
- 使用 `Unity Analytics` 等工具實作使用者進度功能，例如計分、關卡和遊戲內經濟
- 實作 2D 疊加圖，例如抬頭顯示器 (HUD)、小地圖和廣告
- 儲存及擷取應用程式及使用者資料
- 認識線上多人遊戲功能的價值和影響

場景及環境設計之程式設計

- 判斷用於實作音訊資源的腳本
- 確認用於實作 **GameObject** 實例化、摧毀及管理的方法
- 判斷使用 **Unity** 導航系統尋徑的腳本

效能及平台最佳化

- 使用工具 (例如 **Unity Profiler**) 評估錯誤及效能問題
- 確認可透過哪些最佳化措施滿足特定建構平台及/或硬體配置的要求
- 判斷 **XR** 平台常見的 **UI** 預設用途及最佳化

與軟體開發團隊合作

- 版本控制：Unity Collaborate 等工具的影響及用途
- 測試及其對軟體開發流程的影響
- 認識可實現模塊化、可讀性和可重複使用性的各種腳本建構技術

範例問題

問題 1

程式設計師必須實作 UI 選單系統。每個選單包含一個 UI 面板及一個或多個 UI 按鈕，這些全部都是 UI Canvas 物件的子物件。整個 UI 選單系統將在另外載入的單獨場景中建立。

面板和按鈕的美術樣式應保持一致 (色彩、貼圖、按鈕轉換類型等)，但是美術總監尚未敲定這些決策。她希望能在程式設計師進行 UI 作業的同時處理這些設定。她的變更會在場景中所有新增及現有物件上生效。

問題：對於程式設計師而言，要使用 **Unity** 功能輕鬆建立可用的選單系統，並且符合美術總監同時 (且獨立) 調整外觀和質感的需求，可採用的最佳方式是什麼？

- A** 建立 `UI.Button` 和 `UI.Panel` 的子類別，再以程式設計方式設定外觀和質感。
- B** 建立新的按鈕和面板材質，再將它們指派給場景中的所有按鈕和面板。
- C** 對按鈕和面板使用 `Prefab`，再由美術總監修改 `Prefab`。
- D** 撰寫腳本，根據美術總監的意見在場景檔案中搜尋/取代值。

問題 2

一款 3D 跑酷遊戲將場景設定在鐵路調車站的多條平行列車軌道上。玩家永遠在軌道上向前奔跑，且必須跳過迎面而來的列車或跳到相鄰的軌道上來避開列車。

每列新增至軌道的列車都會新增至該軌道上所有其他列車後面。不過，因為列車是以不同的速度朝著玩家移動，或完全不會移動，列車有時候會彼此重疊，這個問題必須修正。

問題：要避免新列車與已在相同軌道上的列車重疊，最有效的方式是什麼？

- A** 在軌道上產生列車時，利用新列車和最後放置在該軌道上之列車的速度，判斷可避免問題的產生位置，以及在軌道上最後放置列車的位置點超過玩家位置點時取消產生列車。
- B** 當列車移動時，從列車頭的方向往前投射光線，並將光線照射到的任何列車，以較快列車的速度向前推動。
- C** 在軌道上產生列車時，在其中加入 **Rigidbody**，再用力來移動列車。
- D** 在軌道上產生列車時，使用長度與列車速度成正比的 **BoxCast**，確保在其進入攝影機後方之前，不會與其他列車相撞。

問題 3

程式設計師正在設計一個黑暗又神祕的房間，必須製作一個閃爍的火炬，在牆壁、地板和天花板上投射出舞動、怪異的陰影。程式設計師為附加至火炬的 `MonoBehaviour` 編寫了這些函式：

```
void Start()
{
    Light light = GetComponent<Light>();
    light.lightMapBakeType = LightMapBakeType.Mixed;
    light.type = LightType.Area;
    light.shadows = LightShadows.Soft;
    light.range = 5f;
}
void Update()
{
    GetComponent<Light>().intensity = Mathf.PerlinNoise(Time.time, 0);
}
```

火炬在執行期間並沒有投射任何光線或陰影。在 `Unity Editor` 中，已將光源設定為預設值。

問題：程式設計師應該如何修改此程式碼，才能滿足設計要求？

- A** 將 `light.lightBakeType` 設定為 `LightmapBakeType.Realtime`
- B** 將 `light.range` 設定為 10
- C** 將 `light.shadows` 設定為 `LightShadows.Hard`
- D** 將 `light.type` 設定為 `LightType.Point`

問題 4

程式設計師正在開發一種採礦模擬遊戲，玩家可在其中挖掘地面尋找礦藏。玩家可以在其中一個現場建立與現有洞穴系統相交的隧道。根據設計文件規定，在目前洞穴和新隧道中發出的任何聲音都應該有部分混響 (Reverb) 效果。程式設計師必須確定使用者始終位於最接近洞穴的 **ReverbZone** (混響區) 裡。

問題：程式設計師該如何操縱 **AudioReverbZone** 屬性，才能滿足這些要求？

- A** 增加反射以適應新區域。
- B** 增加兩個 **ReverbZone** 的 **maxDistance**，使兩者能在新的連接區域內接觸。
- C** 增加混響效果以適應新區域。
- D** 在新區域上增加 **decayTime**。

問題 5

在編寫載入函式時，程式設計師收到編譯錯誤：

```
error CS1624: The body of `CustomAnalytics.LevelLoading()' cannot be an iterator block because `void' is not an iterator interface type
```

```
void LevelLoading(){
    AsyncOperation async = SceneManager.LoadSceneAsync("Level_01");
    while (!async.isDone)
    {
        yield return null;
    }
}
```

問題：程式設計師應如何修正此錯誤？

- A** 將 `yield return null` 變更為 `yield return WaitForSeconds(0)`
- B** 將 `void LevelLoading()` 變更為 `IEnumerator LevelLoading`
- C** 將 `SceneManager.LoadSceneAsync("Level_01")` 變更為 `Application.LoadLevelAdditiveAsync("Level_01")`
- D** 將 `while (!async.isDone)` 變更為 `while (!async.allowSceneActivation)`

問題 6

一款模擬駕駛遊戲的輸入系統使用水平輸入軸來控制轉向。在測試過程中發現，有些搖桿裝置即使搖桿已經置中，還是會記錄轉向輸入。

問題：應該對輸入系統中的輸入軸進行何種變更，才能解決這個問題？

- A** 增加 Gravity (重力)
- B** 將 Snap 設定為 true
- C** 增加 Deadzone
- D** 減少 Sensitivity (靈敏度)

問題 7

在一款以外星球為場景的冒險遊戲中，玩家必須消滅各種生命形式。玩家的得分會隨著每一次殺敵增加。根據設計文件的說明，得分必須與玩家的帳戶連結以方便日後檢索，即使玩家是在不同的遊戲工作階段或在不同裝置上玩，也是如此。

問題：程式設計師若要儲存得分資料，哪一種方法最可靠？

- A** 在 `GameObject` 上使用 `DontDestroyOnLoad()` 來保存得分資料，並在退出應用程式之前將資料上傳至伺服器。
- B** 得分每次更新時儲存至 `PlayerPrefs`，並在退出應用程式之前上傳至伺服器。
- C** 使用靜態值來儲存得分資料，下一個遊戲工作階段便能取得該資料。
- D** 使用資料序列化以便持續儲存得分資料，並將其上傳至伺服器。

正確答案：C、A、D、B、B、C、D