



**認證**

專業

技術美術：  
著色和特效

# 考試目標

Unity 認證專業  
技術美術：  
著色和特效

# 職責

著色器和特效技術美術專注於實現遊戲背後的視覺意圖。在著色器和特效方面具備出色技能和能力的美術常常與其他技術美術和粒子特效美術攜手合作，以準備資源或改進所準備的資源。著色器和特效美術師負責實現遊戲的外觀、風格、主題和美感。

具備這些核心技能的美術要實現實時照明和預設照明，建立和自訂著色器和渲染系統，並建立與其他資源互動的粒子系統和效果。

## 此角色的職位頭銜

- 著色器程式設計師
- 照明師
- 特效美術

# 資質要求

此專業認證適合在此領域工作多年、積累了深厚實踐經驗的專業人員。相關資質要求包括：

- 具備遊戲開發工作室從業經驗，至少參與製作過兩款已上市的遊戲
- 在基於物理的照明技巧和工作流方面具備深厚知識
- 在為基於物理的渲染管線製作材質方面有深入理解
- 對顏色校正和後期效果有深入理解
- 具備深厚的攝影理論知識
- 具備使用 HLSL、CgFX 或其他著色語言編寫著色器的經驗
- 具備使用 C++、C# 或 Unityscript 等語言編寫腳本/程式的知識
- 在粒子系統、動態模擬和 Alembic 等互換格式方面具備深厚知識
- 熟練掌握 Adobe Creative Suite、Substance Designer、Substance Painter、Quixel Suite 等資源製作工具
- 對 2D 和 3D 數學理論有深入理解

# 核心技能

「Unity 認證專業技術美術：著色和特效」認證會考查考者是否具備所需的技能，從而有效地實現遊戲的外觀、風格、主題和美感。只有在下列領域具備豐富經驗的考者才能通過認證。

## 原型設計

- 製作並評估材質和著色器原型

## 著色器和材質

- 構建並測試自訂著色器，以實現下列目標：
  - 模擬各種現象
  - 根據遊戲事件進行動態變更
  - 擴展標準著色器的功能，以支援視效開發 workflow
  - 實現自訂照明模型和非擬真 (NPR) 外觀
- 設計、構建並實現可根據場景設計和輸入而變化的程序化材質和材質效果
- 使用 ShaderGUI 實現自訂材質 UI
- 使用 OnInspectorGUI() 建立自訂檢視面板 (Inspector)
- 實現後製效果 (如景深、顏色校正、泛光、屏幕空間反射、動態模糊和霧效)，以匹配 GDD 中提及的特定電影攝影技巧
- 為渲染材質的使用編寫腳本，以管理實時反射效果

## 渲染和照明

- 瞭解不同類型的光源及其對效能的影響
- 瞭解不同類型的陰影及其對效能的影響
- 瞭解正向和延遲渲染路徑之間的差異
- 確定各個平台對渲染 API 的要求和其局限性
- 使用 Unity API、命令緩衝區和圖形庫來調整和擴展渲染管線

## 粒子系統

- 使用多個粒子系統來模擬大氣現象
- 實現火焰、爆炸、煙霧和水體等典型遊戲效果
- 建立複雜粒子效果，包括具有子噴射器 ( Sub-Emitters ) 的粒子系統以及線條和軌跡渲染器
- 為粒子系統事件編寫腳本，以便在遊戲過程中根據玩家行為、NPC 行為及其他運行時事件觸發此類事件
- 導入和渲染外部產生的模擬數據
- 動態評估碰撞體和變換數據，以實現與粒子系統的互動

## 效能和優化

- 瞭解目標平台的規格和限制
- 優化著色器、粒子系統、後期效果、照明、霧效、陰影等，使其能夠在目標平台上流暢運行
- 瞭解該在何時使用各種優化技巧並在需要時解決出現的問題  
( Billboard、Alpha 排序問題、draw calls、填充率問題、與CPU/GPU相關的性能問題 )
- 使用 Frame Debugger 和各平台專用的幀捕捉工具來分析和評估渲染問題

# 認證 考試主題

---

## 工具和流程管理

- 資源客製化
  - 通過自訂工具和編輯器來改進流程
- 

## 渲染

- 渲染管線
  - 後製特效
  - Unity 鏡頭
- 

## 著色器

- 著色器構建、原型設計和自訂
  - 渲染設定著色器知識
  - 與著色器相關的腳本知識
- 

## 粒子和效果

- 粒子系統自訂和擴展
  - 特效開發技巧
- 

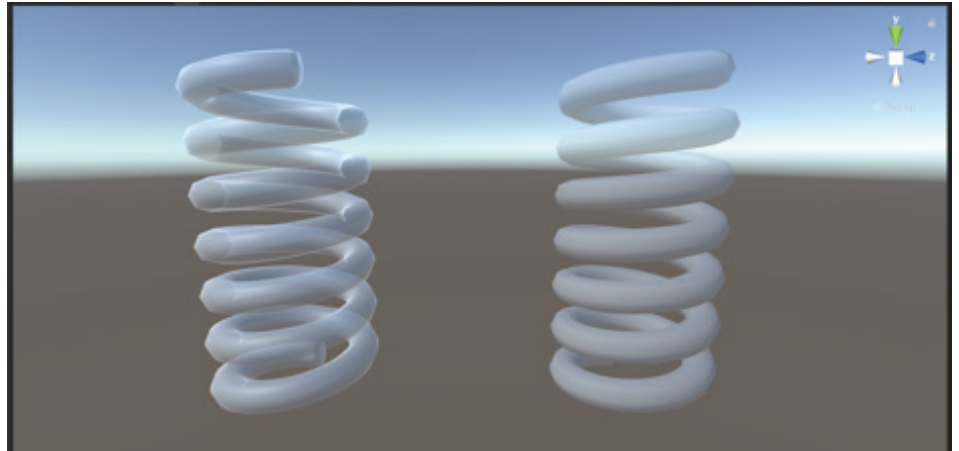
## 效能

- 場景優化

# 範例問題

## 問題 1

請參見圖示：



「渲染模式」( Rendering Mode ) 設定為「透明」( Transparent ) 的標準著色器已應用於單個網格。一些面重疊並且使用了朝向相同方向的法線。左邊的模型渲染不正確。概念圖要求到達右側圖像中的效果。

技術美術應該如何達到這個效果？

- A 使用自訂剔除 ( Culling ) 模式。
- B 使用自訂渲染隊列。
- C 編寫自訂多通道著色器。
- D 使用 GrabPass 編寫自訂著色器。

## 問題 2

使用支援 OpenGL ES 3.0 的最小平台的手遊戰略遊戲，其遊戲設計文件 ( GDD ) 定義了一個在執行時按照程序放置產生的金礦的平面地圖。金礦一部分在地下，一部分在地面。

**顯示金礦入口最有效的方法是什麼？**

- A** 在執行時用程式產生地圖網格。
- B** 在地圖上使用計算著色器 ( Compute Shader ) 。
- C** 在地圖上使用視差著色器 ( Parallax Shader ) 。
- D** 使用帶 stencil mask 的著色器。



## 問題 3

第一人稱冒險遊戲將場景設定在具有動態照明的範圍受限的多山草地上。技術美術需要在程式化的天空顏色上添加一個程式化的雲系統，並在遠處添加連綿不斷、無法觸及的山脈。場景中的山地必須受動態光霧影響。

技術美術應該如何避免山脈與雲之間的相互干擾？

- A 使用 `CameraEvent.BeforeSkybox` 增加山脈。
- B 將山脈作為低解析度幾何體和細節等級 ( LOD ) 添加到場景中。
- C 使用 `CameraEvent.AfterEverything` 增加山脈。
- D 將山脈作為 `World Space UI Images` 添加，每個圖層位於各自的「畫布」 ( Canvas ) 上。

# 問題 4

技術美術需要模擬地球上的降雨。數據是從天氣伺服器即時檢索的。天氣數據顯示為彩色降雨圖像，圖像與全球常規漫反射紋理一致。現有一個函數可以用來抽取地球上的某個位置來獲得降雨強度。遊戲設計文件（GDD）要求降雨要呈現雨滴降落形式，類似於電視上的新聞天氣圖。

技術美術在執行期間將降雨分佈和強度數據傳遞給發射器（Emitter）的最有效方法是什麼？

**A**

1. 使用 Shape 模組設定為從全球網格的縮放實例向下發射。
2. 將降雨紋理應用到粒子材料上使無雨地區不可見或被省略。

**B**

1. 在腳本更新中，獲取全球可見區域的隨機點並檢索降雨數據。
2. 將這些數據應用於 emit 函數中的樣式粒子。

**C**

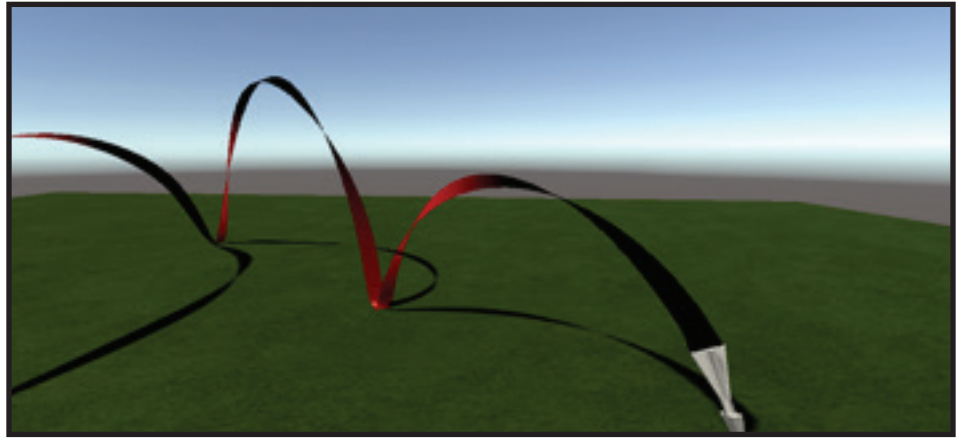
1. 使用 Shape 模組設定為從球體向下發射。
2. 將降雨紋理應用到粒子材料上使無雨地區不可見或被省略。

**D**

1. 在腳本更新中，獲取球形發射器的活動粒子列表。
2. 利用它們的位置檢索降雨數據，去除無雨區域的粒子，並更新有效雨粒的強度。

# 問題 5

請參見圖示：



一個遊戲專案包括一個玩家控制的武器，該武器可發射彈跳發射物，這些發射物在軸上旋轉並留下一條顯示拖痕的扭曲的色帶。如圖所示，色帶必須對場景照明正確反應。

**建立拖痕效果的最有效的解決方案是什麼？**

- A** 在與發射物關聯的 GameObject 上建立拖痕渲染器 ( Trail Renderer ) 。
- B** 在與發射物關聯的 GameObject 上建立腳本驅動的線渲染器 ( Line Renderer ) 。
- C** 使用 Mesh 類別建立一個腳本來實現發射物效果。
- D** 建立一個腳本來驅動使用 SkinnedMeshRenderer 裝配的色帶模型。

---

正確答案：C、D、A、B、C