



Certified

Expert

**Technical Artist:
Shading & Effects**

시험 목표

Unity 인증 전문
테크니컬 아티스트:
그림자 및 효과

역할

그림자 및 효과 테크니컬 아티스트는 게임의 뒤에서 시각적인 의도를 전달하는데 중점을 둡니다. 그림자 및 효과 기술과 역량을 갖춘 아티스트는 종종 다른 테크니컬 아티스트 및 효과 아티스트와 협력하여 에셋을 준비하거나 준비된 에셋을 향상시킵니다. 그림자 및 효과 아티스트는 또한 게임의 모습, 스타일, 테마 및 예술을 구현해야 합니다.

이러한 핵심 기술을 갖춘 아티스트는 실시간으로 베이킹된 라이팅을 구현하고, 셰이더와 렌더링 시스템을 제작해서 사용자 지정하며, 다른 에셋과 상호작용하는 파티클 시스템 및 효과를 생성합니다.

이 업무의 직함

- 셰이더(Shader) 작가
- 라이트 제작자
- 효과 아티스트

필수 조건

이 전문가 인증 시험은 이 분야에서 수년 동안 일했으며, 다음과 같이 다양하고 실용적인 고급 응용 프로그램 경험을 쌓은 사람들에게 권장됩니다.

- 비디오 게임 개발 스튜디오 경험(출시된 타이틀 두 개 이상 포함)
- 물리 기반 라이팅 기술 및 작업 흐름에 대한 명확한 지식
- 물리 기반 렌더링 파이프라인의 머티리얼 제작에 대한 전문가 수준의 이해
- 색상 보정 및 사후 효과에 대한 전문가 수준의 이해
- 사진 개념에 대한 명확한 지식
- HLSL, CgFX 또는 기타 셰이딩 언어로 셰이더 작성 경험
- C++, C# 또는 Unityscript 언어로 스크립팅/코딩에 대한 해박한 지식
- 파티클 시스템, 동적 시뮬레이션 및 Alembic과 같은 교환 형식에 대한 명확한 지식
- Adobe Creative Suite, Substance Designer, Substance Painter, Quixel Suite 등 에셋 제작 도구의 능숙한 사용
- 2D 및 3D 수학 개념에 대한 명확한 이해

핵심 기술

그림자 및 효과 테크니컬 아티스트 인증 시험은 지원자가 게임의 모습, 스타일, 테마 및 예술을 효과적으로 구현하는 데 필요한 기술을 갖추고 있는지 확인합니다. 합격한 지원자는 다음 분야에서 고급 능력을 갖추게 될 것입니다.

프로토타입 제작

- 머티리얼 및 셰이더 프로토타입 제작 및 평가

셰이더 및 머티리얼

- 다음 목적으로 커스텀 셰이더를 구성 및 테스트합니다.
 - 현상 시뮬레이션
 - 게임플레이 이벤트에 따라 동적으로 변경
 - 표준 셰이더의 기능을 확장해서 모양 개발 작업 흐름 지원
 - 커스텀 라이팅 모델 및 비사실적(NPR) 모양 구현
- 씬 설계와 입력에 맞춰 조정되는 절차적 머티리얼 및 머티리얼 효과를 설계, 구성 및 구현
- ShaderGUI를 사용해서 커스텀 머티리얼 UI 구현
- OnInspectorGUI()를 사용해서 커스텀 인스펙터(Inspector) 제작
- 게임 설계 문서(GDD)에서 참조된 특정 영화 촬영과 일치하도록 사후 효과 (예: 피사계 심도, 색상 보정, 블룸, 스크린 공간 반사, 모션 블러 및 안개) 구현
- 실시간 반사를 관리하기 위해 렌더 텍스처(Render Texture) 사용 스크립트

렌더링 및 라이팅

- 다양한 유형의 라이트와 성능 영향 이해
- 다양한 유형의 그림자와 성능 영향 이해
- 정방향 렌더링 경로와 지연 렌더링 경로의 차이점 이해
- 플랫폼별 렌더링 API 요구 사항 및 제약 조건 결정
- Unity API, 명령 버퍼, 그래픽 라이브러리를 사용해서 렌더링 파이프라인 조정 및 연장

파티클 시스템

- 다중 파티클 시스템을 이용한 대기현상 시뮬레이션
- 화재, 폭발, 연기 및 물과 같은 일반적인 게임 효과 구현
- 서브-이미터(Sub-Emitter), 라인(Line), 트레일 렌더러(Trail Renderer)
파티클 시스템을 포함한 복합 파티클 효과 제작
- 플레이어 및 NPC 행동 및 기타 런타임 이벤트에 따라 게임플레이 중 발생하는 파티클 시스템 이벤트 스크립트
- 외부에서 생성된 시뮬레이션 데이터 가져오기 및 렌더링
- 콜라이더(Collider) 및 트랜스폼(Transform) 데이터를 동적으로 평가하여 파티클 시스템과의 상호작용 구현

성능 및 최적화

- 대상 플랫폼의 사양 및 제약 이해
- 대상 플랫폼에서 실행되도록 셰이더, 파티클 시스템, 사후 효과, 라이팅, 안개 및 그림자 등 최적화
- 최적화 기술 및 문제 해결(빌보드, 알파 정렬 문제, 그리기 호출, 채우기 속도 문제, CPU/GPU 경계 시나리오)이 필요한 경우 및 사용 시기 이해
- Frame Debugger 및 플랫폼별 프레임 캡처 도구를 사용해서 렌더링 문제 분석 및 평가

인증 시험 주제

도구 및 파이프라인

- 에셋 사용자 지정
 - 커스텀 도구 및 Editor 사용자 정의를 통한 프로세스 개선
-

렌더링

- 렌더링 파이프라인
 - 사후 처리 효과
 - Unity의 카메라
-

셰이더

- 셰이더 구성, 프로토타입 제작 및 사용자 지정
 - 렌더링 설정 셰이더 지식
 - 셰이더와 관련된 스크립트 지식
-

파티클 및 효과

- 파티클 시스템 사용자 정의 및 확장
 - 효과 기술
-

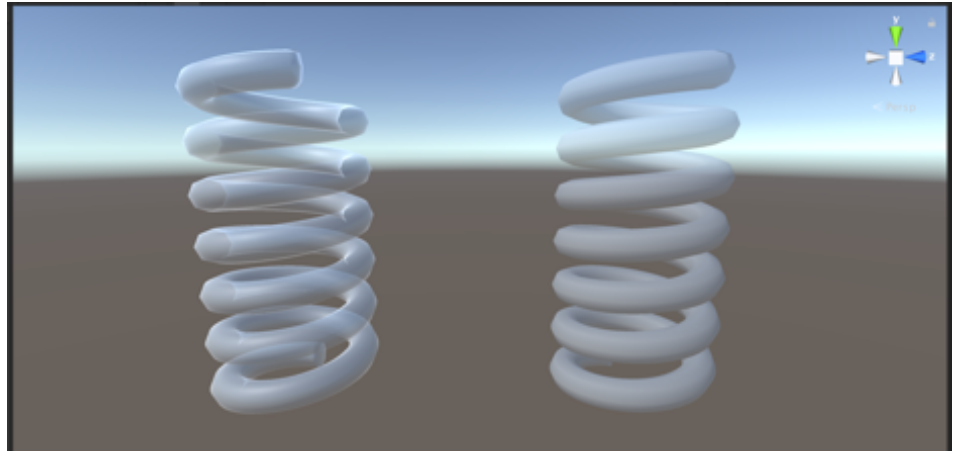
성능

- 씬(Scene) 최적화

예제 문항

문항 1

다음 이미지 참고:



렌더링 모드가 투명(Transparent)으로 설정된 스탠다드 셰이더가 단일 메시에 적용되었습니다. 오버랩되는 면들의 일부는 같은 방향을 향하는 노멀(Normal)을 포함하게 됩니다. 왼쪽 모델은 잘못 렌더링되어 있습니다. 컨셉 아트에서 요구하는 것은 오른쪽 이미지를 생성하는 효과입니다.

테크니컬 아티스트는 이 효과를 어떻게 구현해야 할까요?

- A 커스텀 컬링 모드(Culling Mode)를 사용한다.
- B 커스텀 렌더 큐를 사용한다.
- C 커스텀 멀티 패스(multi-pass) 셰이더를 작성한다.
- D 그랩패스(GrabPass)를 사용해서 커스텀 셰이더를 작성한다.

문항 2

모바일 전략 게임에 대한 게임 설계 문서(GDD)에는 OpenGL ES 3.0을 지원하는 플랫폼이 최저 사양으로 지정되어 있고, 런타임 시 금광이 생성되어 순서대로 배치되는 평면 지도가 기술되어 있습니다. 금광의 일부는 지하에 묻혀 있지만 일부는 지상에 노출되어 있습니다.

다음 중 금광 입구를 가장 효율적으로 드러내는 방법은 무엇입니까?

- A 런타임 시 지도 메시지를 순서대로 생성한다.
- B 지도에서 컴퓨트 셰이더(Compute Shader)를 사용한다.
- C 지도에서 페럴랙스 셰이더(Parallax Shader)를 사용한다.
- D 스텐실 마스크(Stencil Mask)가 있는 셰이더를 사용한다.

문항 3

1인칭 어드벤처 게임은 동적 조명이 적용되는 산간 목초지라는 제한된 지리 영역을 배경으로 합니다. 테크니컬 아티스트는 순차적으로 표현되는 하늘 색상 위에 순서대로 구름 시스템을 추가하고, 닿을 수 없지만 멀리서 계속 보이는 산을 추가해야 합니다. 씬에서 산에는 동적 광원과 안개의 영향을 받아야 합니다.

테크니컬 아티스트는 산과 구름의 인터랙션(interaction)을 어떻게 방지해야 하나요?

- A CameraEvent.BeforeSkybox를 사용해서 산을 추가한다.
- B 씬에 낮은 해상도의 지오메트리와 디테일 수준(LOD)으로 산을 추가한다.
- C CameraEvent.AfterEverything을 사용해서 산을 추가한다.
- D 각 캔버스(Canvas)에 설정된 산을 월드 공간 UI 이미지(World Space UI Image)로 추가한다.

문항 4

테크니컬 아티스트는 지구에 내리는 비를 구현해야 합니다. 데이터는 날씨 서버에서 실시간으로 검색됩니다. 날씨 데이터는 지구의 규칙적인 디퓨즈 텍스처와 일치하는 유색의 강우 이미지로 표시됩니다. 강우량을 측정하기 위해 지구의 한 위치 표본을 얻는 기능이 제공됩니다. 게임 설계 문서(GDD)에서는 TV 뉴스의 날씨 지도처럼 비가 떨어지는 빗방울의 형태로 방출되도록 요구합니다.

테크니컬 아티스트가 런타임 동안 강우 분포 및 강우량 데이터를 이미터(Emitter)로 전달하는 가장 효율적인 방법은 무엇입니까?

A

1. 셰이프(Shape) 모듈 세트를 사용해서 지구 메시의 확대/축소된 인스턴스 아래쪽으로 방출합니다.
2. 파티클 머티리얼에 강우 텍스처를 적용해서 비가 오지 않는 지역의 파티클이 보이지 않거나 잘리지 않도록 합니다.

B

1. 스크립트된 업데이트에서 지구상의 보이는 지역에서 무작위 포인트를 얻고 강우 데이터를 가져옵니다.
2. 이 데이터를 방출 함수의 스타일 파티클에 적용합니다.

C

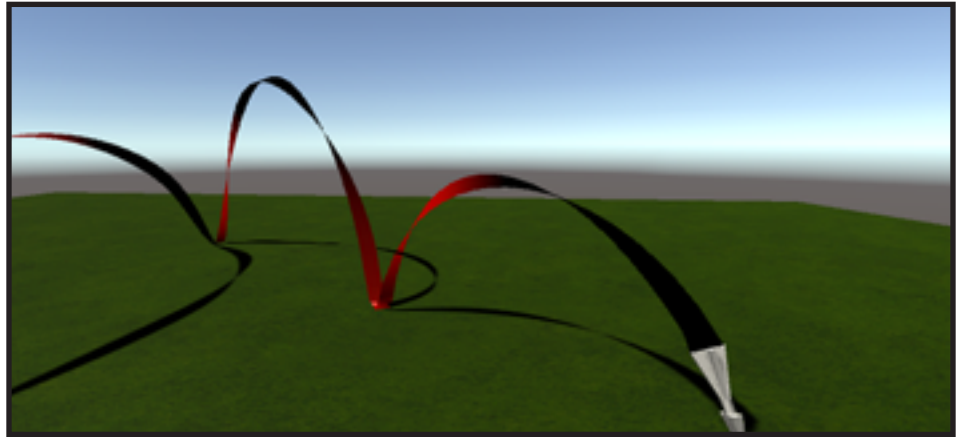
1. 셰이프(Shape) 모듈 세트를 사용해서 스피어의 아래쪽으로 방출합니다.
2. 파티클 머티리얼에 강우 텍스처를 적용해서 비가 오지 않는 지역의 파티클이 보이지 않거나 잘리지 않도록 합니다.

D

1. 스크립트된 업데이트에서 구체 이미터로부터 활성화된 파티클 목록을 가져옵니다.
2. 액티브 파티클의 위치를 사용하여 강우 데이터를 검색하고, 비가 내리지 않는 지역의 파티클을 제거하고, 유효한 비 파티클의 강도를 업데이트합니다.

문항 5

다음 이미지 참고:



게임 프로젝트에 플레이어가 제어하는 무기가 포함되어 있습니다. 이 무기는 축을 중심으로 회전하며 튕기는 발사체를 발사하고, 꼬인 형태의 리본을 남겨 궤적을 표시합니다. 이미지에서 보여지는 것처럼 리본은 썬의 조명에 올바르게 반응해야 합니다.

다음 중 궤적 효과를 생성하는 가장 효율적인 방법은 무엇입니까?

- A** 게임 오브젝트에 발사체의 부모가 되는 트레일 렌더러(Trail Renderer)를 생성한다.
- B** 게임 오브젝트에 발사체의 부모가 되는 스크립트 기반의 라인 렌더러(Line Renderer)를 생성한다.
- C** 메시 클래스를 사용하여 발사체에 효과를 구현하는 스크립트를 생성한다.
- D** SkinnedMeshRenderer로 리깁된 리본 모델을 운용하는 스크립트를 생성한다.

정답: C, D, A, B, C