



**Certified**

**Expert**

**Gameplay  
Programmer**

# 시험 목표

Unity 전문가 인증  
게임플레이 프로그래머

# 역할

게임플레이 프로그래밍 전문가는 게임 디자인의 실행에 주력합니다.

게임플레이 프로그래머는 아트 팀에서 제작한 에셋에 스크립트를 추가해 게임에 생명력을 불어넣습니다. 사용자 인터페이스(UI), 환경, 캐릭터, 오브젝트를 구현해야 하며, 게임 메카닉이 실행될 수 있도록 최적화와 스크립팅도 담당해야 합니다. 또한 게임 내의 플레이어 이외의 캐릭터(NPC) 기능도 구현해야 합니다.

게임플레이 프로그래머는 게임의 현실감을 불어 넣기 위해 게임 디자인 문서(GDD)를 참고하며 디자이너와 밀접하게 일해야 합니다. (또는 자신이 디자이너가 될 수도 있습니다.) 프로그래머는 고도의 기술을 보유하고 있으며 아이디어를 구현하는데 필요한 것이 무엇인지 이해하고 있습니다. 이는 Unity 컴포넌트를 언제, 어떻게 사용해야 하는지, 자체적으로 컴포넌트를 쓰고, 프리팹(Prefab)을 구성하고 각 씬에 적절하게 배치해야 하는지 알고 있는 것도 포함됩니다. 기술 설계 문서(TDD)를 고려하여 모든 작업을 수행합니다.

## 이 업무의 직함

- 게임플레이 프로그래머
- 게임 스크립터/엔지니어
- 레벨 디자이너

# 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 근무 경력
- 몰입 게임, 웹/모바일 게임 및 디지털 게임 디자인 경력
- **C#** 언어로 스크립팅/프로그래밍에 대한 해박한 지식
- 초기 게임 컨셉부터 출시까지 게임의 전체 주기 경험
- 게임 디자인(레벨, 플레이어 이외의 캐릭터(**NPC**), 게임 메카닉)의 모든 측면에 대한 전문적인 이해
- **Unity Services**의 디자인 및 구현 경력
- 캐릭터 및 환경 설정을 포함한 게임 에셋 및 애니메이션 파이프라인에 대한 이해
- 파일 구조, 작명 규칙 및 기타 확립된 프로토콜을 준수하는 믿음직한 체계화 기술
- 최고의 게임을 만들기 위해 프리팹(**Prefab**) 라이브러리를 생성 및 사용하는 능력

# 핵심 기술

전문가 게임플레이 프로그래머 인 증은 지원자가 게임을 효과적으로 조합하고 최적화하는 데 필요한 기술을 갖추고 있는지 확인합니다. 합격한 지원자는 다음 분야에서 전문적인 역량을 갖추어야 합니다.

## 프로토타입 제작

- 빠른 반복을 위한 핵심 게임플레이 프로토타입 제작 및 평가
- 게임 메카닉 기능을 증명하기 위한 게임 오브젝트(GameObjects) 설계 및 구현
- 아트 팀에서 생성한 에셋을 사용하여 게임 설계 문서(GDD)를 기반으로 프리팹(Prefab) 라이브러리 제작
- 프리팹(Prefab) 라이브러리의 폴더 구조 구성 및 설정

## 레벨 설계

- 콜라이더(Collider) 및 리지드바디(Rigidbody) 컴포넌트를 사용하여 상호작용 레벨 프리팹(Prefab) 설계 및 제작
- 상호작용 프리팹(Prefab)에서 물리 마스킹에 적합한 레이어 설정
- 런타임 시 생성되는 프리팹(Prefab) 설정 및 동적 게임플레이 구현
- 콜라이더(Collider)로 발동하는 스크립트된 이벤트와 이벤트 트리거 설정 및 구현
- 게임 오브젝트(GameObjects)가 스테이트 머신(State Machine)을 씬의 상호작용/트리거(Trigger)에 연결할 수 있게 하는 커스텀 로직 컴포넌트 제작 및 스크립트
- 스태틱(Static) 및 런타임 시 생성되는 배치를 포함하여 씬 전체에 효과 배치 및 구성
- 플랫폼 사양별 디테일 레벨(LOD) 구성
- 플랫폼 사양별 게임 오브젝트(GameObject) 배치 및 종속성 평가
- 스트리밍 및 스태틱 씬 로딩에 대한 씬 평가
- 여러 씬으로 게임 레벨 구성
- 게임플레이 향상을 위한 시네마틱 설정

## 플레이어 이외의 캐릭터(NPC) 설계

- NPC 로직과 인공지능(AI) 스크립트 설계 및 제작
- 네브메쉬(NavMesh), 네브메쉬에이전트(NavMeshAgent), 네브메쉬 장애물(NavMesh Obstacle) 및 오프-메시 링크(Off-Mesh Link)를 사용하여 NPC의 내비게이션 및 경로 찾기 구현
- 네브메쉬(NavMesh) 영역 유형 및 비용 설정
- 네브메쉬(NavMesh) 영역을 활성화/비활성화하는 트리거(Trigger) 설정
- 네브메쉬에이전트 회피(NavMeshAgent Avoidance) 및 군중 시뮬레이션 구현
- 플랫폼 사양별 NPC 배치 및 종속성 평가
- 애니메이션 클립에서 프레임 기반의 오디오 및 효과 설정

## 사용자 인터페이스(UI) 및 게임 메카닉 설계

- 애니메이터 컨트롤러(Animator Controller), 스테이트 머신(State Machine) 및 게임 메카닉 로직 스크립트에 애니메이션 시스템 사용
- 게임 오브젝트(GameObjects) 상호작용을 위해 콜라이더(Collider), 리지드바디(Rigidbody) 컴포넌트, 물리 머티리얼 평가 및 최적화
- HUD 디스플레이, 미니맵, 레이더 시스템, 체력 바 및 기타 데이터 기반 요소 등 게임플레이 관련 사용자 인터페이스(UI) 구현

## 성능 최적화 및 대상 플랫폼

- 게임 레벨에서 에셋번들(AssetBundle) 다운로드 및 배치 구현
- 다양한 플랫폼 및/또는 가상 현실(VR)에 대한 입력 및 컨트롤러 스키마 설계 및 수정
- 플랫폼 사양별 런타임 및 스토리지 최적화를 위해 게임 오브젝트(GameObjects) 및 씬 분석
- 씬 전체에서 오클루전 컬링(Occlusion Culling) 최적화
- 런타임에서 게임 레벨 테스트 및 디버그

## Unity Services 적용: Ads, IAP, Analytics

- 간결하고 보상을 수령할 수 있는 Unity Ads 적용
- Unity IAP 적용
- 게임 설계 문서(GDD) 및 기술 디자인 문서(TDD)에서 Unity Analytics 통합 설계
- Analytics를 사용하여 플레이어 행동을 모니터링하는 커스텀 데이터 이벤트 모니터링 설정
- Analytics 데이터를 기반으로 기존 레벨을 분석 및 평가하고 수정 사항 권장
- Unity Performance Reporting을 사용해서 플랫폼별 게임 빌드를 수정 및 최적화

# 인증 시험 주제

---

## 프로토타입 제작 (핵심 게임플레이를 위한 신속한 피드백 및 수정)

- 핵심 게임플레이 프로토타입 제작
- 프로토타입 단계에서 일어나는 충돌 및 솔루션

---

## 레벨 디자인 프로그래밍

- 물리(physics) 설정
- 레이캐스팅(Raycasting)
- 런타임 중에 스크립트로 생성된 프리팹(Prefab)
- 레벨 로직 및 행동
- 파티클 시스템(Particle System) 및 효과로 레벨 채우기
- 플랫폼 최적화
- 씬 로딩 및 언로딩
- 시네마틱 표시 방법

---

## NPC 설계 프로그래밍

- NPC 로직 및 행동
- 내비게이션 및 경로 찾기
- NPC 생성 및 배치

---

## 사용자 인터페이스(UI) 구현

- UI 좌표 시스템 및 UI 스크립팅

---

## 대상 플랫폼 성능 최적화

- 렌더링 최적화
- 에셋번들(AssetBundle) 다운로드 및 설정
- 게임플레이 디버깅
- 플랫폼들의 차이점 및 게임플레이에 미치는 영향

---

## Unity Services 적용

- Unity Ads
- Unity IAP
- Unity Analytics
- Unity Cloud Build

# 문항 예제

## 문항 1

게임플레이 프로그래머가 횡스크롤 슈팅 게임 프로토타입을 제작 중입니다. 플레이어는 비행기이며 적들은 다양한 종류의 **UFO**입니다. 플레이어가 **UFO**를 파괴하면 **UFO**가 있던 곳에 폭발 효과가 나타납니다. 플레이어가 사망하면 특수 플레이어 기체 폭발 효과가 나타납니다.

비행기는 화면에서 최대 **64**개의 총알을 쏠 수 있습니다. 화면에 표시될 수 있는 모든 **UFO**는 최대 **128**대이며 **UFO** 총알 수는 최대 **1024**개, 그리고 폭발 효과 개수는 최대 **128**개입니다.

게임플레이 프로그래머는 **Editor**에 어떠한 게임 오브젝트(**GameObjects**)를 배치하고, 어떠한 게임 오브젝트(**GameObjects**)를 런타임 시 생성할지 (예: 오브젝트 풀) 결정해야 합니다.

어떤 방법으로 이를 수행할 수 있습니까?

**A** 플레이어, **UFO**, 플레이어 폭발 효과 및 **UFO** 폭발 효과를 **Editor**에 배치합니다. 플레이어 총탄 및 **UFO** 총탄은 런타임 시 생성되게 합니다.

**B** 플레이어, 플레이어 총탄, **UFO** 및 **UFO** 총탄을 **Editor**에 배치합니다. 플레이어 폭발 효과 및 **UFO** 폭발 효과는 런타임 시 생성되게 합니다.

**C** 플레이어 및 **UFO**를 **Editor**에 배치합니다. 플레이어 총탄, **UFO** 총탄, 플레이어 폭발 효과 및 **UFO** 폭발 효과는 런타임 시 생성되게 합니다.

**D** 플레이어 및 플레이어 폭발 효과를 **Editor**에 배치합니다. **UFO**, **UFO** 총탄, 플레이어 총탄 및 **UFO** 폭발 효과는 런타임 시 생성되게 합니다.

## 문항 2

3인칭 오픈 월드 게임에 대한 게임 설계 문서(GDD)를 다음과 같이 작성했습니다. 게임 설계 문서(GDD)는 주인공 게임플레이의 두 가지 유형을 명시하고 있습니다.

1. 도보로 전 세계 돌아다니기
2. 오토바이 운전하기

각 모드의 높이는 상대적으로 비슷하지만, 플레이어가 오토바이를 타면 도보로 걸을 때보다 훨씬 빠르게 이동할 수 있습니다.

게임플레이 프로그래머는 걷기와 타기를 전환할 때 수정해야 할 게임의 여러 구역을 결정합니다.

- 카메라 시야각(Field of View)
- 디테일 레벨(LOD) 거리
- 레벨 스트리밍

플레이어가 오토바이를 타고 있을 때, 게임플레이 프로그래머는 이 게임 설계 문서(GDD)의 각 구역을 처리하기 위해 다음 중 어떠한 전략을 활용해야 합니까?

- A** 카메라 시야각(FOV) 값을 줄입니다. 레벨이 더 빨리 스트리밍됩니다.
- B** 카메라 시야각(FOV) 값을 늘립니다. 레벨이 더 천천히 스트리밍됩니다.
- C** 디테일 레벨(LOD) 거리 값을 늘립니다. 레벨이 더 천천히 스트리밍됩니다.
- D** 카메라 시야각(FOV) 값을 늘립니다. 레벨이 더 빨리 스트리밍됩니다.



## 문항 3

이 게임 설계 문서(GDD)는 플레이어가 걸어 다니며 도시 내의 사람들과 상호작용할 수 있는 대도시로 설정되어 있습니다. 플레이어는 시민들에게 받은 다양한 무작위 퀘스트를 완료할 수 있습니다. 플레이어는 한 번에 5개의 퀘스트만 수행할 수 있습니다. 플레이어가 도시의 한쪽 끝에서 다른 쪽 끝까지 걷는다면 약 4시간이 소요됩니다.

게임 HUD의 탐 다운 시점 미니맵에는 다음 요소들이 2D 아이콘으로 표시됩니다.

- 플레이어 위치
- 진행 중인 퀘스트
- 도시를 활보하는 다양한 시민들

미니맵은 확대하거나 축소할 수 있습니다. 완전히 확대하면 플레이어 아이콘만 표시됩니다. 완전히 축소하면 전체 도시의 25%만 볼 수 있습니다. 플레이어가 미니맵 화면을 끝까지 확대할 경우, 게임이 느려집니다.

### 이 느려짐 현상의 원인은 다음 중 무엇입니까?

**A** 메모리에서 모든 시민 아이콘의 텍스처에 대한 언로드 요청이 너무 많습니다.

**B** 진행 중인 퀘스트 아이콘이 너무 자주 쿼리되어 현재 몇 개가 진행 중인지 불러올 수 없습니다.

**C** 미니맵의 컬링(Culling) 알고리즘이 너무 오래 걸려 도시 대부분이 제한됩니다.

**D** HUD가 너무 복잡하며 매 프레임마다 다시 생성됩니다.

## 문항 4

이 게임 설계 문서(GDD)는 빠듯한 메모리 용량에서 운영되는 모바일 게임입니다. 이 게임에는 원활한 성능을 보장하기 위해 에셋번들(AssetBundle)을 로드하는 게임 섹션이 있습니다. 각 섹션은 다른 섹션과 독립적이므로, 에셋번들(AssetBundle)은 다른 섹션에 영향을 미치지 않고 로드 및 언로드할 수 있습니다.

하지만 때로는 분홍색 텍스처가 나타납니다.

**이 문제의 원인은 다음 중 무엇입니까?**

- A** AssetBundle.mainAsset이 손상되었습니다.
- B** AssetBundle.Unload()가 너무 빨리 호출 중입니다.
- C** AssetBundle.LoadAllAssets()이 잘못된 유형으로 호출 중입니다.
- D** AssetBundle.LoadAssetWithSubAssets()이 잘못된 문자열 및 유형으로 호출 중입니다.

## 문항 5

이 게임 설계 문서(GDD)는 레벨이 300개가 넘는 모바일 매치 3D 퍼즐 게임입니다. 게임에는 각 레벨마다 다음 **Analytics** 커스텀 이벤트가 포함되어 있습니다.

1. **“levelStarted”**: 플레이어가 레벨을 시작할 때 발동
2. **“levelCompleted”**: 플레이어가 레벨을 완료할 때 발동

이 팀은 게임에서 가장 어려운 레벨 5개가 무엇인지 확인하고 싶어합니다. 레벨에는 시간제한이 있습니다.

**상위 5개를 결정하기 위해 어떤 추가 커스텀 이벤트가 필요합니까?**

**A** 레벨에서 소요한 시간을 전송하기 위해 **“levelTime”**을 추가합니다.

**B** 운영 체제가 모바일 앱을 종료할 때 실행되는 **“levelRestarted”**를 추가합니다.

**C** 플레이어가 실패하거나 레벨을 일찍 종료할 때 발동하는 **“levelFailed”**를 추가합니다.

**D** 모바일 앱이 백그라운드 프로세스에서 다시 시작할 때 실행되는 **“levelResumed”**를 추가합니다.

---

정답: D, D, C, B, C