



**Certified**

**Expert**

**Gameplay  
Programmer**

# 試験概要

Unity認定エキスパート  
ゲームプレイプログラマー

# 役割

ゲームプレイプログラミングの専門家は、ゲームデザインの遂行に重点を置いています。ゲームプレイプログラマーは、アートチームによって作成されたアセットにスクリプトを追加することで、ゲームをより楽しくすることができます。彼らは、ゲームのメカニズムを遂行するために必要な最適化とスクリプティングだけでなく、ユーザーインターフェイス (UI)、環境、キャラクター、オブジェクトの実装も担当しています。また、ゲームに参加するノンプレイヤーキャラクター (NPC) にも機能をもたらします。

ゲームプレイプログラマーは、デザイナーと一緒に作業し(あるいはデザイナー自身でもよい)、ゲームを実現するためにゲームデザインドキュメント (GDD) を参照します。これらの人々は非常に技術的であり、アイデアを生かすために必要なものを理解しています。これには、Unity コンポーネントを使用するタイミングと方法、独自のコンポーネントの記述、プレハブの構成、各シーンの適切な整理などが含まれます。彼らは、テクニカルデザインドキュメント (TDD) を念頭に置いて、このすべてを行います。

## この役割の役職名

- ゲームプレイプログラマー
- ゲームスクリプター/エンジニア
- レベルデザイナー

# 前提条件

このエキスパート認定は、この分野で数年の経験があり、次のような高度で実用的なさまざまなアプリケーションの経験を積んだ方を推奨します。

- ビデオゲーム開発スタジオでの経験
- 没入型ゲーム、ウェブ/モバイルゲーム、デジタルゲームの設計経験
- C# を使用したスクリプティング/プログラミングに対する優れた知識
- 初期のコンセプトから発売まで、ゲームのライフサイクル全体に関わった経験
- ゲームデザイン(レベル、ノンプレイヤーキャラクター[NPC]、ゲームメカニズム)のすべての側面に対する専門知識
- Unityサービスの設計と実装経験
- キャラクターと環境の設定を含む、ゲームアセットとアニメーションパイプラインの知識
- ファイル構造、命名規則、その他の確立された手続きに準拠した強固な組織スキル
- 可能な限り最高のゲームを作るためにプレハブのライブラリを作成し、使用する能力

# コアスキル

エキスパート ゲームプレイプログラマー認定資格は、受験者が効果的にゲームを組み立てて最適化するために必要なスキルを持っていることを証明します。合格した受験者は、次の分野で高度な能力を備えています。

## プロトタイピング

- 迅速なイテレーションのためのコアゲームプレイのプロトタイプの実装と評価
- ゲームメカニズムの機能を裏付けするゲームオブジェクトの実装と実装
- アートチームが作成したアセットを使用して、GDDに基づいてプレハブライブラリを作成
- プレハブライブラリフォルダ構造を整理およびセットアップ

## レベルデザイン

- コライダーとリジッドボディコンポーネントを使用してインタラクティブなレベルプレハブを実装および作成
- インタラクティブなプレハブのPhysicsマスクに適切なレイヤーを設定
- 実行時に生成されるプレハブを実装し、動的なゲームプレイを実装
- コライダーをトリガーとして使用して、イベントトリガーとスクリプトイベントを実装および実装
- ゲームオブジェクトがステートマシンとリンクしてシーン内でインタラクションやトリガーとして機能するように、カスタムロジックコンポーネントを実装およびスクリプト化
- 静的および動的生成の両方を含む、シーン全体にエフェクトを実装および実装
- 各プラットフォーム仕様に合わせたLevel of detail(LOD)を設定
- 各プラットフォーム仕様に対しゲームオブジェクトの実装と依存関係の評価
- ストリーミングシーンと静的なシーンロードを実装
- 複数のシーンでゲームのレベルを実装
- ゲームプレイをさらに高めるためにシネマティクス(カットシーン)を実装

## ノンプレイヤーキャラクター (NPC) デザイン

- NPCロジックと人工知能 (AI) スクリプトの設計と作成
- NavMeshe、NavMesh Agent、NavMesh Obstacle、Off-Meshリンクを使用して、NPCのナビゲーションと経路探索を実装
- NavMeshエリアタイプとコストを設定
- ナビメッシュエリアを有効/無効にするようにトリガーを設定
- NavMeshAgentの回避と群衆シミュレーションの実装
- プラットフォーム仕様におけるNPCの配置と依存関係を評価
- アニメーションクリップでフレームベースのオーディオとエフェクトを設定

## ユーザーインターフェースおよびゲームメカニクスデザイン

- アニメーター・コントローラー、ステートマシン、およびゲームメカニクスのロジックスクリプトにアニメーションシステムを使用
- インタラクティブなゲームオブジェクトのためのコライダー、リジッドボディのコンポーネントと物理特性マテリアルの評価および最適化
- ヘッドアップディスプレイ(HUD)、ミニマップ、レーダーシステム、体力ゲージ、その他のデータ駆動型要素など、ゲームプレイに関するユーザーインターフェイスを実装

## パフォーマンスの最適化とターゲットプラットフォーム

- アセットバンドルによるゲームレベルのダウンロードおよび配置を実装
- 異なるプラットフォームやVirtual Reality(VR) のための入力およびコントローラスキーマの設計および変更
- プラットフォーム仕様ごとのランタイムおよびストレージの最適化のためのゲームオブジェクトおよびシーンの分析
- シーン全体のオクルージョンカリングの最適化
- ランタイム中のゲームレベルのデバッグおよびテスト

## Unityサービス実装: Ads、In-App Purchasing、Analytics

- Unity Adsの実装(リワード型およびインタースティシヤル型)
- Unity IAP(In-App Purchasing:アプリ内課金) を実装
- GDDとTDDにおいて Unity Analyticsを統合した設計
- Analyticsを使用してプレーヤの振る舞いをモニタリングするカスタムデータイベントを設定
- 既存のレベルを分析および評価し、Analyticsのデータに基づいて変更を推奨
- UnityのPerformance Reportingを使用して、プラットフォームごとにゲームのビルドを変更および最適化

# 認定試験トピック

---

## プロトタイピング (迅速なイテレーションのためのコアゲームプレイ)

- コアゲームプレイプロトタイピング
  - プロトタイピング段階における葛藤と解決
- 

## レベルデザインプログラミング

- Physics コンフィグレーション
  - レイキャスト
  - ランタイム中でのスクリプトによるプレハブの生成
  - レベルロジックと振る舞い
  - パーティクルシステムとエフェクトを備えたレベルの発生
  - プラットフォーム最適化
  - シーンのロードとアンロード
  - シネマティクス(カットシーン)を表示する方法
- 

## NPCデザインプログラミング

- NPCのロジックと振る舞い
  - ナビゲーションと経路探索
  - NPCの生成と配置
- 

## ユーザーインターフェイスの実装

- UI 座標システムと UI スクリプト
- 

## パフォーマンスの最適化とターゲットプラットフォーム

- レンダリングの最適化
  - アセットバンドルのダウンロードと設定
  - ゲームプレイのデバッグ
  - プラットフォームの違いとゲームプレイへの影響
- 

## Unity サービスの実装

- Unity Ads
- Unity In-App Purchasing
- Unity Analytics
- Unity Cloud Build

# サンプル問題

## 問題 1

ゲームプレイプログラマーは、ゲームがサイドスクロール弾丸シューティングゲームであるプロトタイプを作成しています。主人公は飛行機であり、敵はさまざまな種類のUFOです。プレイヤーがUFOを破壊すると、UFOがあったところに爆発エフェクトが表示されます。プレイヤーが死亡すると、特別なプレイヤーの飛行機の爆発エフェクトが表示されます。

飛行機はスクリーン上で最大64個の弾丸を撃てます。画面に表示できるUFOの最大数は128です。画面上のUFOの最大弾数は1024です。画面上の爆発エフェクトの最大数は128です。

ゲームプレイプログラマーは、どのゲームオブジェクトをエディタに配置すべきか、またどのゲームオブジェクトを実行時に生成すべきか(すなわち、オブジェクトプールを利用するか)を決めなければなりません。

どのように実現するのが良いでしょうか。

- A** プレイヤー、UFO、プレイヤーの爆発エフェクト、UFOの爆発エフェクトはエディターで配置し、プレイヤーの弾丸とUFOの弾丸は実行時に生成する。
- B** プレイヤー、プレイヤーの弾丸、UFO、UFOの弾丸はエディターで配置し、プレイヤーの爆発エフェクトとUFOの爆発エフェクトは実行時に生成する。
- C** プレイヤーとUFOはエディターで配置し、プレイヤーの弾丸、UFOの弾丸、プレイヤーの爆発エフェクト、およびUFOの爆発エフェクトを実行時に生成する。
- D** プレイヤーとプレイヤーの爆発効果はエディターで配置し、UFO、UFOの弾丸、プレイヤーの弾丸、UFOの爆発エフェクトは、実行時に生成する。

## 問題 2

このゲームデザインドキュメント (GDD) は三人称視点のオープンワールドゲームです。GDDにはメインプレイヤーに2種類のモードが記載されています。

1. ウォーキング: 足で世界中を歩く
2. ライディング: オートバイを運転する

各モードの相対的な目線の高さは同じぐらいですが、プレイヤーがオートバイに乗っているときは、歩くよりもはるかに速く移動することができます。

ゲームプレイプログラマーは、ウォーキングとライディングの切り替え中にゲームのいくつかのエリアを微調整する必要があると判断しました。

- カメラ視野 (FOV)
- Level of Detail (LOD) 距離
- レベルストリーミング

**プレイヤーがオートバイに乗っているとき、ゲームプレイプログラマーはこのGDDの各エリアに対処するためにどの戦略を使用すべきですか。**

**A** カメラのFOVは、より小さい値でなければなりません。  
レベルはより速くストリーミングする必要があります。

**B** カメラのFOVは、より大きな値でなければなりません。  
レベルは、より遅くストリーミングする必要があります。

**C** LOD距離はより大きな値でなければなりません。  
レベルは、はるかに遅くストリーミングする必要があります。

**D** カメラのFOVはより大きな値でなければなりません。  
レベルはより速くストリーミングする必要があります。

# 問題 3

このGDDでは大規模な都市に設置されており、プレイヤーは歩き回って市の誰とでも交流できるように記載されています。プレイヤーは市民のためのさまざまなランダムクエストを遂行することができます。プレイヤーは5つのアクティブなクエストを持つことができます。プレイヤーが都市の一端から他端まで歩いた場合、およそ4時間かかります。

このゲームにはヘッドアップディスプレイ(HUD)にトップダウンのミニマップが含まれており、以下の2Dアイコンが表示されます。

- プレイヤーの位置
- すべてのアクティブなクエスト
- 市内を歩いている様々な市民全員

ミニマップはズームイン/ズームアウトすることができます。最大までズームインするとプレイヤーのアイコンのみが表示されます。最大までズームアウトすると、街全体の25%が見えます。プレイヤーがミニマップ上で最大までズームインした場合のみスローダウンが発生します。

**スローダウンの原因として考えられる一番の原因は何ですか。**

- A** メモリからすべての市民アイコンのテクスチャのアンロードコールが多すぎるため。
- B** アクティブなクエストアイコンは、現在アクティブな数を取得するために頻繁にクエリを実行しているため。
- C** ミニマップのカリングアルゴリズムが都市の大部分を選び出すのに時間がかかりすぎているため。
- D** HUDが複雑すぎて、フレーム毎に再作成されているため。

# 問題 4

このGDDは厳しいメモリ制約の中で動作するモバイルゲームです。このゲームでは、スムーズなパフォーマンスを確保するために アセットバンドルに読み込まれているのセクションがあります。各セクションは独立しているため、他のセクションに影響を与えずに アセットバンドルをロードおよびアンロードできます。

ただし、ピンクのテクスチャが表示されることがあります。

**この問題の原因は何ですか。**

- A** AssetBundle.mainAsset が壊れているため。
- B** あまりにも早くAssetBundle.Unload() が呼び出されているため。
- C** AssetBundle.LoadAllAssets () が間違ったタイプで呼び出されているため。
- D** AssetBundle.LoadAssetWithSubAssets () が、間違った文字列とタイプで呼び出されているため。

# 問題 5

このGDDは300以上のレベルがあるモバイルマッチ3Dパズルゲームです。このゲームには各レベルにおいて次のAnalyticsカスタムイベントが含まれています。

1. “levelStarted”:プレイヤーがレベルを開始したことを通知
2. “levelCompleted”:プレイヤーがレベルを完了したことを通知

開発チームはゲームの中で最も難しい上位5つのレベルがどれかを判断したいと考えています。なおレベルには時間制限があります。

**上位5つを決定するためには、さらにどのようなカスタムイベントが必要でしょうか。**

- A** レベルで費やされた時間を通知するために “levelTime” を追加
- B** オペレーティングシステムによってモバイルアプリが終了したことを通知するために “levelRestarted” を追加
- C** Aプレイヤーが失敗したり、レベルを早期に終了したことを通知するために “levelFailed” を追加
- D** モバイルアプリケーションがバックグラウンドプロセスから再開したことを通知するために “levelResumed” を追加

---

正解: D, D, C, B, C