



Certified

Expert

Technical Artist:

Rigging &

Animation

試験概要

Unity認定エキスパート
テクニカルアーティスト
リギング & アニメーション

役割

リギング & アニメーションのテクニカルアーティストは、ゲームのアーティストとプログラマーとの間の重要な橋渡し役として、芸術的なビジョンとプラットフォームの制限を尊重しながら、ゲームに体系的に統合するためのアセットを提供します。

さらに、ゲーム開発チームは、オーディオとアニメーションを含む複雑なアニメーションやゲームオブジェクトをスクリプト化するためのリギング&アニメーションテクニカルアーティストのコアスキルに依存しています。アーティストはまた、さまざまなプラットフォームでこれらのアセットを最適化するパイプラインツールを作成してサポートし、それらがゲームのテクニカルデザインドキュメント(TDD)に準拠していることを保証します。

この役割の役職名

- テクニカルアーティスト
- リガー
- テクニカルアニメーター
- テクニカルキャラクターデザイナー

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、少なくとも2つのタイトルをリリースした経験
- C++、C#、またはUnityScriptなどの言語を使用したスクリプティング/コーディングに関する知識
- 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の経験
- リギング/キャラクターセットアップとアニメーションの経験
- Python、MEL、MaxScriptなどのデジタルコンテンツ作成 (DCC) で使用されるスクリプティング言語に精通
- 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
- Adobe Creative Suite、Substance Designer、Substance Painter、Quixel Suite、Autodesk Maya、3ds Max、Pixologic ZBrushおよびMotion Builderなどのアセット作成ツールを使用

コアスキル

エキスパートテクニカルアーティスト リギング&アニメーション認定資格は、リグとアニメーションアセットをゲームに効果的に統合するために必要なスキルを受験者が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

- ゲームデザインドキュメント (GDD) を検討して、アニメーションにフォーカスされたエディタツール群を決定し、デザインチームの他のメンバーがゲームを構築しつつも「デザインを守る」ことを実施および評価
- ベストプラクティスを開発する目的でプロトタイプを作成し、評価する。および各プラットフォーム毎にパフォーマンス仕様を策定することでテクニカルデザインドキュメント (TDD) を改良
- アニメーションやリギングの問題を解決するための技術的な解決策を評価し、推奨

パイプライン

- アセットのインポートのカスタマイズと自動化
- ゲームオブジェクトの属性を手順通りに変更
- 複数のゲームオブジェクトのパラメータを制御
- ビヘイビアとアニメーションを手順通りに実装

ゲームオブジェクトの準備

- ゲームの実装のためのLevel of Detail (LOD) を持ったアセットプレハブの準備
- HumanoidとGenericリグのアニメーションタイプの実装と対応付け
- プレハブとして配置されるJoint、Cloth、Rigidbody、およびPhysicsコンポーネントを使用したリグとスクリプトの複雑な実装
- ゲームプレイを遊びやすくするためのカスタムPhysicマテリアルの作成とテスト
- メッシュコライダーおよびPhysicマテリアルが複雑に絡み合ったオブジェクトの評価と最適化

アニメーションの準備

- ステートマシンにブレンドツリーを作成
- 複雑なスクリプト (アクティブ状態で複数レイヤー)、ステート マシンの高速動作
- ステートマシンレイヤーを作成し、レイヤーアニメーションする
- ブレンドシェイプの重み付けとトランジションのためのステートとビヘイビアの作成
- 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/ FXを設定

パフォーマンスと最適化

- ターゲットプラットフォームの仕様と制限を理解
- FKとIKリグの違いと、それらの相対的なパフォーマンスへの影響を理解
- プラットフォーム要件ごとに複雑な組み込みをテストし最適化
- シーンのパフォーマンスを評価し、プロファイラーを使用してボトルネックを特定
- リグの複雑さ、バッチ処理、および頂点シェーダメソッドの観点からCPUとGPUの最適化を評価

認定試験トピック

ツーリングとパイプライン

- エディタのカスタマイズ
 - アセットのカスタマイズ
 - カスタムツールを使用したプロセス自動化
-

プロトタイピング

- リギングとアニメーションのためのプロトタイプ作成
-

アニメーションとリギング

- アニメーションシステムステートマシンとアニメーションイベントの設定
 - リグのセットアップとアニメーション
 - ダイナミックアニメーションのためのPhysicsコンポーネント
-

パフォーマンス

- シーンの最適化

サンプル問題

問題1

コンバットゲームのゲームデザインドキュメント (GDD) は、ノンプレイヤーキャラクター (NPC) がアイドル、ランニング、アタック、ディフェンドなどの一連のアニメーションを有していると記述されています。NPCはすべての戦闘において武装および両手剣を使用します。

GDDは、NPCがポーズレイヤーによって決定された異なるポーズを持ってゲームの外観を変えるように指定します。ブルートポーズ(野獣のようなポーズ)のアニメーションクリップにはキャラクターが肩を前に突き出して猫背の姿勢が含まれています。ヒーローポーズのアニメーションクリップには、キャラクターが肩と胸を広げた誇らしげな姿勢が含まれています。ゲームをテストプレイしたところ、NPCにポーズレイヤーを適用すると両手に剣の柄が整列しないことがわかりました。

この問題の解決策は何でしょうか。

- A** `OnAnimatorIK()` を使用して、両手に武器を移動するための位置、回転、および関連するウェイトを設定します。
- B** それぞれのアバターにおいて、両手に適切に武器を移動させるためにAvatar InspectorのMusclesタブのPer-Muscle Settingsにて胸(chest)の設定をします。
- C** `OnStateMove()` をオーバーライドするStateMachineBehaviourを使用し、`animator.MatchTarget()` を呼び出して手の位置を調整します。
- D** それぞれのアバターにおいて、オプションのボーンを追加設定することで、手に武器を近づけたり遠ざけたりすることができます。

問題2

ゲームデザインドキュメント(GDD)には次の要件があります。

- 武器は持ち替えることができるようにする。
- 武器はPropWeaponという名前のジョイントを親とする。
- PropWeaponは、キャラクターのrootの子とする。

アニメーションチームはデジタルコンテンツ作成(DCC)パッケージにて PropWeapon ジョイントにコンストレイントを使用して、武器の切り替え手をアニメーション化しました。アニメーションはすべてのフレームをベイクし、FBXを使用してエクスポートしました。テクニカルアーティストは、エディターでゲームをプレイしたときに、DCCパッケージに含まれていない不安定な動き(jittery)をすることに気付きました。

不安定なアニメーションの原因は何ですか？

- A** 反対側の手で再生しているアニメーションは、PropWeaponジョイントに影響を与えるため。
- B** The position error on the compression settings is too low.
- C** 武器の取り付けジョイントは、直接手の親となるものではないため。
- D** Animatorのルートモーションは、PropWeaponジョイントの位置に影響するため。

問題3

これは人間が操作するキャラクターがセキュリティロボットによって狩られていることを特徴とするサバイバルゲームのためのゲームデザインドキュメント(GDD)です。ロボットは外骨格が硬く、主要な関節部分に見える油圧ピストンが特徴です。アニメーションディレクターは、ロボットがより硬く、より機械的な動作をすることを要求しています。人間とロボットのキャラクターには、歩行アニメーションを再利用するためにHumanoid Rig Animationタイプとしています。

テクニカルアーティストは、モーションスタイルの違いを強調するキャラクターのアバターをどのように設定する必要がありますか？

- A** Per-Muscle Settingsを使用して、ロボットの動作範囲を制限し、人間の設定をデフォルトのままにします。
- B** ロボットのアバターをAポーズに設定し、人間をTポーズとして設定します。
- C** ロボットの上腕と上肢の骨の長さを増やし、人間の上腕と上肢の骨の長さを元のままにしておきます。
- D** ロボットのオプションボーンを使用して動きの範囲を広げ、人間にはオプションのボーンを有効にしない。

問題4

テクニカル・アーティストは、ヒューマノイド・キャラクターが移動中に正確に足を地面に付けるためのシステムに取り組んでいます。この解決策は、さまざまなサイズのキャラクターに適用されます。このシステムでは、2つのコリジョンメッシュに対して処理が必要です。1つは移動用ヒューマノイドキャラクターのカプセル、もう1つはより正確に足を地面に接地するためのコリジョンメッシュです。

システムの一部において、足の位置を地面に投影する必要があり、そのためのコードを以下に示しています。

```
Vector3 ProjectPositionOnGround(Vector3 position)
{
    Vector3 ret = position;

    RaycastHit hitInfo = new RaycastHit();
    if (Physics.Raycast(position + new Vector3(0, 0.5f, 0), new Vector3(0,
-1, 0), out hitInfo, 1.0f, m_LayerMask))
    {
        ret = hitInfo.point;
    }

    return ret;
}
```

テクニカルアーティストは、一部のキャラクターにおいてIKによる接地が期待通りに機能しないことに気づきました。この問題を解決するには、コードをどのように変更する必要がありますか？

- A** レイヤーを動的に設定して、すべてのキャラクターが適切なコリジョンメッシュに対してレイキャスティングされるようにします。
- B** キャラクターサイズに基づいてオフセットを使用して、Raycastのoriginを変更します。
- C** 必ずコライダーに届くようにRaycastの方向ベクトルをスケールします。
- D** RaycastのoriginとmaxDistanceプロパティに、キャラクターサイズに基づくオフセットをセットします。

問題5

堅いコートを持つキャラクターのアニメーション中に頭と顎(あご)が突き抜けないようにデジタルコンテンツクリエイション(DCC)パッケージにてリグを設定します。希望する関節運動を達成するために、コートのセットアップは、物理的に駆動しない12本の追加のボーンを使用します。ゲームのすべてのキャラクターはヒューマノイドリグとしてインポートし、同じアニメーションセットが再利用されます。

アニメーションデータは、ターゲットビルドプラットフォームにおいて非常に多くのスペースを占有しています。

既存のアニメーションの動作を一致させながらアニメーションを最適化する最も効率的な方法は何ですか？

- A** DCCリグを編集して、コートの襟のジョイント数を減らす。
- B** DCCリグを編集し、BlendShapesでコートの襟のジョイントの設定を置き換える。
- C** アニメーションを読み込むときに12個のコートの襟ボーンをマスクし、Unity Physicsコンポーネントを使用して襟の衝突動作を再現する。
- D** アニメーションを読み込むときに12個のコートの襟ボーンをマスクし、DCCリグの設定に一致する動作を実装するスクリプトを作成する。

正解: A, B, A, D, D