

VRoidキャラクター演技システム

開発ロードマップ

プロジェクト: RPG_defete

作成日: 2025-12-30

概要

目的

VRoidキャラクターをストーリーエンジンと連携し、
表情・アニメーション・IK・カメラ・エフェクトを統合制御

主要機能

機能	説明
表情コントローラ	制御・ブレンド・遷移
アニメーション	ポーズ再生、NavMesh連携
IKシステム	視線・手の制御
カメラワーク	フォーカス、ズーム、シェイク
エフェクト	パーティクル演出
ラグドール	やられ時の演出

既存基盤

要素	状況
ストーリーエンジン	✓ 実装済み
VRM表情フィールド	✓ 設計済み
UniVRM	✓ 導入済み
CombatSystem	✓ 実装済み

SkillData.cs に表情連携の基盤あり

Phase 1: 基盤システム

VRM表情コントローラ

- `VRMExpressionController.cs` 新規作成
- VRM10標準表情の制御
- 表情ブレンド・遷移機能
- ストーリーエンジンAPI

Phase 1: ストーリーコマンド

コマンド	説明
expression	表情変更
pose	アニメーション再生
move	キャラクター移動
lookat	視線制御
camera	カメラワーク
effect	エフェクト発生

Phase 2: アニメーション

VRMアニメーションコントローラ

- Animator動的制御
- AnimationClip再生
- ブレンドツリー対応

アニメーション取得先

- **Mixamo** - 無料Humanoid
- **Asset Store** - 有料/無料パック
- **BOOTH** - VRM/VRMA形式

Phase 2: NavMesh連携

CharacterNavigator

- NavMeshAgentベースの移動
- パス追従アニメーション
- 目的地到達イベント

Phase 3: IKシステム

必要パッケージ

```
com.unity.animation.rigging
```

視線IK (Look At)

- Multi-Aim Constraint使用
- 頭と目の回転制御

手のIK

- Two Bone IK Constraint使用
- 肩→肘→手のチェーン

Phase 4: 演出システム

カメラワーク

- キャラクターフォーカス
- ズームイン/アウト
- シェイク効果

エフェクト連携

- 既存 `SkillData.effectPrefab` 活用
- パーティクル生成
- タイミング制御

Phase 5: ラグドール

方式	特徴	推奨
Ragdoll Wizard	標準・無料	★★★
Active Ragdoll	部分制御可	★★★★
Puppet Master	高品質・有料	★★★★★

ファイル構成

```
Assets/Scripts/  
├── Character/  
│   ├── VRMExpressionController.cs  
│   ├── VRMAnimationController.cs  
│   ├── CharacterNavigator.cs  
│   ├── CharacterIKController.cs  
│   └── RagdollController.cs  
├── Camera/  
│   └── StoryCameraController.cs  
└── Story/Commands/  
    └── (各コマンドクラス)
```

開発スケジュール

Phase	内容	期間
1	基盤システム	8日
2	アニメーション	8日
3	IKシステム	8日
4	演出システム	7日
5	ラグドール	8日

優先度

● High

1. VRM表情コントローラ
2. ストーリーコマンド拡張
3. Animation Rigging導入

● Medium

4. アニメーションコントローラ
5. 視線IK / カメラワーク

● Low

6. 手のIK / ラグドール

技術要件

パッケージ	用途	状態
UniVRM	表情制御	✓ 導入済み
Animation Rigging	IK	◆ 要導入
AI Navigation	移動	✓ 導入済み

調査結果

VRoid標準モーション

- 直接エクスポート 不可
- BlendShape（表情）は 保持される
- アニメーションはMixamo等から取得

Unity IK

- Multi-Aim: 視線制御
- Two Bone IK: 手の制御
- ランタイム変更可能

まとめ

1. 既存基盤を活用 - SkillData, StoryEngine
2. **Phase 1**から着手 - 表情コントローラ
3. 段階的に実装 - 5フェーズ構成

Next Steps

1. Animation Riggingパッケージ導入
2. VRMExpressionController実装開始
3. ストーリーコマンド拡張設計

質問・確認事項があればお知らせください