

[RealityKit](#) / Character control, skeletons, and inverse kinematics

API Collection

Character control, skeletons, and inverse kinematics

Direct the movements and animation of models.

Overview

Games and immersive game-like experiences often rely on animated character models to represent the player or non-player characters. The `CharacterControllerComponent` simplifies the process of moving a character around a scene. It handles basic movement, including navigating up and down stairs and slopes and jumping. It also allows you to sync the character movement with specific animations.

To further animate models in the scene, you may need to define a [SkeletalPose](#) or adopt a full inverse kinematics solution with [IKComponent](#).

Topics

Character control

`struct CharacterControllerComponent`

A component that manages character movement.

`struct Collision`

A container that holds collision state for the character controller.

`struct CollisionFlags`

An option set that specifies which parts of the character capsule have collided with other objects.

```
struct CharacterControllerStateComponent
```

A component that represents the state of a character controller.

Skeletons

```
struct SkeletalPosesComponent
```

A component that exposes the collection of named animation skeletal poses.

```
struct SkeletalPose
```

A container that holds the position and orientation of each joint in a single animation skeleton.

```
struct SkeletalPoseSet
```

A collection of named skeletal poses.

Inverse kinematics components

```
struct IKComponent
```

A component that allows you to procedurally animate a skeletal model using a full body inverse kinematics solver.

```
class Joint
```

The update stage object that lets you read and update the current settings of a single joint in an IK solver.

```
struct JointCollection
```

Ordered dictionary like container with fixed size.

```
class Solver
```

The update stage object that lets you read and update the current settings of a single solver instance.

```
struct SolverCollection
```

Ordered dictionary like container with fixed size.

```
class Constraint
```

The update stage object that lets you read and update the current settings of a single constraint in an IK solver.

```
struct ConstraintCollection
```

Ordered dictionary like container with fixed size.

```
class IKResource
```

A reference counted immutable resource which contains one or more inverse kinematics solver rigs.

```
struct IKSolverDefinition
```

A container describing a solver instance.

Inverse kinematics rigs

```
struct IKRig
```

A full body inverse kinematics rig definition for a single skeleton.

```
struct Joint
```

A definition of a rig joint and its IK solver settings.

```
struct JointCollection
```

Ordered dictionary-like container with a fixed size.

```
struct Constraint
```

A definition of a rig constraint.

```
struct ConstraintsCollection
```

Ordered dictionary-like container.

See Also

Game development

☰ Gaming sample code projects

Explore a collection of projects relating to game development.

☰ Entity animations

Dynamically move, rotate, and scale entities at runtime.