

[TabletopKit](#) / BaseEquipmentState

Structure

BaseEquipmentState

A state for equipment that contains no equipment-specific data.

visionOS 2.0+

```
struct BaseEquipmentState
```

Topics

Creating an equipment state

```
init(parentID: EquipmentIdentifier, seatControl: ControllingSeats, pose : TableVisualState.Pose2D, boundingBox: Rect3D)
```

Creates a base state for equipment using a parent, location, and player interactions.

```
init(parentID: EquipmentIdentifier, seatControl: ControllingSeats, pose : TableVisualState.Pose2D, entity: Entity)
```

Getting the parent equipment

```
var parentID: EquipmentIdentifier
```

The identifier for the parent equipment that holds or contains this equipment.

Rendering the equipment

```
var boundingBox: Rect3D
```

A 3D bounding box that encloses the equipment.

```
var pose: TableVisualState.Pose2D
```

The 2D position and rotation of the equipment relative to the equipment parent, or table.

Controlling the equipment

```
var lockedBy: PlayerIdentifier?
```

The identifier for the player who exclusively controls the equipment.

```
var seatControl: ControllingSeats
```

The seats that can manipulate or interact with the equipment.

Relationships

Conforms To

BitwiseCopyable

Copyable

Equatable

EquipmentState

MutableEquipmentState

Sendable

SendableMetatype

See Also

Equipment

{ } Implementing playing card overlap and physical characteristics

Add interactive card game behavior for a pile of playing cards with physically realistic stacking and overlapping.

```
protocol Equipment
```

A protocol for equipment that players directly interact with in a game.

```
struct EquipmentCollection
```

A collection of equipment whose state can be inspected and modified.

`protocol EntityEquipment`

A protocol for equipment in a game that you render using RealityKit.

`struct EquipmentIdentifier`

A unique identifier for equipment.

`protocol EquipmentState`

A protocol for the equipment data that TabletopKit syncs between players.

`struct EquipmentStateCollection`

A collection of equipment states that can be inspected and modified.

`protocol CustomEquipmentState`

A specialized protocol for the equipment state that allows to accommodate custom data that TabletopKit syncs between players.

`protocol MutableEquipmentState`

A protocol for equipment data that TabletopKit syncs between players, and that can be mutated.

`struct CardState`

A state for cards that contains face up and down information.

`struct DieState`

A state for dice that contains the current value.

`struct RawValueState`

A state for equipment that contains a game-specific value.

`enum ControllingSeats`

The seats that can manipulate or interact with the equipment.