TabletopKit / TabletopAction

Protocol

TabletopAction

A protocol for objects that describe an action in a tabletop game.

visionOS 2.0+

```
protocol TabletopAction
```

Topics

Getting the player

```
var playerID: PlayerIdentifier?
```

The player performing the action.

Required

Getting game-specific information

```
var context: UInt64
```

An integer value that your game uses.

Required

Moving equipment

```
static func moveEquipment(some Equipment, childOf: any Equipment, order
: MoveEquipmentAction.Order?, pose: TableVisualState.Pose2D?, context:
UInt64) -> Self
```

```
static func moveEquipment(matching: EquipmentIdentifier, childOf: EquipmentIdentifier, order: MoveEquipmentAction.Order?, pose: TableVisualState.Pose2D?, context: UInt64) -> Self
```

Changing equipment state properties

```
static func updateEquipment<E>(E, faceUp: Bool?, seatControl: ControllingSeats?, pose: TableVisualState.Pose2D?, boundingBox: Rect3D?, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, rawValue: UInt64?, seatControl: ControllingSeats?, pose: TableVisualState.Pose2D?, boundingBox: Rect3D?, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, seatControl: ControllingSeats?, pose: TableVisualState.Pose2D?, boundingBox: Rect3D?, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, state: E.State, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, state: E.State, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, state: E.State, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, state: E.State, context: UInt64) -> Self
```

```
static func updateEquipment<E>(E, value: Int?, seatControl: ControllingSeats?, pose: TableVisualState.Pose2D?, boundingBox: Rect3D?, context: UInt64) -> Self
```

Taking turns

```
static func setTurn(forSeat: some TableSeat, context: UInt64) -> Self
```

```
static func setTurn(forSeats: some Sequence, context: UInt64) -> Self
```

```
static func setTurn(forSeats: some Sequence<any TableSeat>, context: UInt64) -> Self
```

```
static func setTurn(matching: TableSeatIdentifier, context: UInt64) -> Self
static func setTurn(matching: [TableSeatIdentifier], context: UInt64) -
> Self
```

Keeping score

```
static func updateCounter(ScoreCounter, context: UInt64) -> Self
```

```
static func updateCounter(matching: ScoreCounter.Identifier, value:
Int64, context: UInt64) -> Self
```

Creating bookmarks

```
static func createBookmark(StateBookmark, context: UInt64) -> Self
```

```
static func createBookmark(id: StateBookmarkIdentifier, context: UInt64
) -> Self
```

Adding actions

```
static func customAction(some CustomAction, context: UInt64) -> Self
```

Relationships

Conforming Types

CreateBookmarkAction
MoveEquipmentAction
SetTurnAction
UpdateCounterAction
UpdateEquipmentAction

See Also

Actions

```
struct MoveEquipmentAction
```

An action that moves a piece of equipment on the table or changes the grouping.

`struct UpdateEquipmentAction`

An action that updates properties of equipment on the table.

`struct SetTurnAction`

An action that sets the current seats participating in the current turn.

`struct UpdateCounterAction`

An action that updates the game counter.

`struct CreateBookmarkAction`

An action that takes a snapshot of the game.

`protocol CustomAction`

A protocol that represents an action whose behavior is implemented outside of TabletopKit. A custom action that can be applied to a `TableState`.