UIMotionSlide // inherits from CanvasProperties

Properties

defaultCurve	an animation curve that will be used if not supplied any in method calls

Static Methods

SlideLeft	slides UI element to the left
SlideRight	slides UI element to the right
SlideUp	slides UI element to up
SlideDown	slides UI element to down

UIMotionSnap // inherits from CanvasProperties

Properties

defaultCurve	an animation curve that will be used if not supplied any in method calls

Static Methods

SnapLeft	snaps UI element to the left border of the screen
SnapRight	snaps UI element to the right border of the screen
SnapUp	snaps UI element to the top border of the screen
SnapDown	snaps UI element to the bottom border of the screen
SnapOutLeft	snaps out UI element to the left border of the screen
SnapOutRight	snaps out UI element to the right border of the screen
SnapOutUp	snaps out UI element to the top border of the screen
SnapOutDown	snaps out UI element to the bottom border of the screen

UIMotionScale

Properties

defaultCurve	an animation curve that will be used if not supplied any in method calls

Static Methods

ScaleToOne	scales UI element to one, brings it back to original size
ScaleToZero	scales UI element to zero, makes it disappear
ScaleBy	slides UI element to up

UIMotionSlide.SlideLeft

```
public static void SlideLeft(RectTransform content);
public static void SlideLeft(RectTransform content, float slideRateOverScreenWidth);
public static void SlideLeft(RectTransform content, AnimationCurve curve);
public static void SlideLeft(RectTransform content, float slideRateOverScreenWidth, AnimationCurve curve);
public static void SlideLeft(RectTransform content, UnityEvent OnSlideEnd);
public static void SlideLeft(RectTransform content, float slideRateOverScreenWidth, UnityEvent OnSlideEnd);
public static void SlideLeft(RectTransform content, AnimationCurve curve, UnityEvent OnSlideEnd);
public static void SlideLeft(RectTransform content, float slideRateOverScreenWidth, AnimationCurve curve, UnityEvent OnSlideEnd);
```

Parameters

content	content that will be slided
slideRateOverScreenWidth	how much screen width-long that slide will be
curve	animation curve to animate the slide
OnSlideEnd	event to be called after sliding process ends

UIMotionSlide.SlideRight

```
public static void SlideRight(RectTransform content);
public static void SlideRight(RectTransform content, float slideRateOverScreenWidth);
public static void SlideRight(RectTransform content, AnimationCurve curve);
public static void SlideRight(RectTransform content, float slideRateOverScreenWidth, AnimationCurve curve);
public static void SlideRight(RectTransform content, UnityEvent OnSlideEnd);
public static void SlideRight(RectTransform content, float slideRateOverScreenWidth, UnityEvent OnSlideEnd);
public static void SlideRight(RectTransform content, AnimationCurve curve, UnityEvent OnSlideEnd);
public static void SlideRight(RectTransform content, float slideRateOverScreenWidth, AnimationCurve curve, UnityEvent OnSlideEnd);
```

Parameters

content	content that will be slided
slideRateOverScreenWidth	how much screen width-long that slide will be
curve	animation curve to animate the slide
OnSlideEnd	event to be called after sliding process ends

UIMotionSlide.SlideUp

public static void **SlideUp**(RectTransform **content**);

public static void **SlideUp**(RectTransform **content**, float **slideRateOverScreenHeight**);

public static void **SlideUp**(RectTransform **content**, AnimationCurve **curve**);

public static void **SlideUp**(RectTransform **content**, float **slideRateOverScreenHeight**, AnimationCurve curve);

public static void **SlideUp**(RectTransform **content**, UnityEvent **OnSlideEnd**);

public static void **SlideUp**(RectTransform **content**, float **slideRateOverScreenHeight**, UnityEvent **OnSlideEnd**);

public static void **SlideUp**(RectTransform **content**, AnimationCurve **curve**, UnityEvent **OnSlideEnd**);

public static void **SlideUp**(RectTransform **content**, float **slideRateOverScreenHeight**, AnimationCurve **curve**, UnityEvent **OnSlideEnd**);

Parameters

content	content that will be slided
slideRateOverScreenHeight	how much screen height-long that slide will be
curve	animation curve to animate the slide
OnSlideEnd	event to be called after sliding process ends

UIMotionSlide.SlideDown

```
public static void SlideDown(RectTransform content);
public static void SlideDown(RectTransform content, float slideRateOverScreenHeight);
public static void SlideDown(RectTransform content, AnimationCurve curve);
public static void SlideDown(RectTransform content, float slideRateOverScreenHeight, AnimationCurve curve);
public static void SlideDown(RectTransform content, UnityEvent OnSlideEnd);
public static void SlideDown(RectTransform content, float slideRateOverScreenHeight, UnityEvent OnSlideEnd);
public static void SlideDown(RectTransform content, AnimationCurve curve, UnityEvent OnSlideEnd);
public static void SlideDown(RectTransform content, float slideRateOverScreenHeight, AnimationCurve curve, UnityEvent OnSlideEnd);
```

Parameters

content	content that will be slided
slideRateOverScreenHeight	how much screen height-long that slide will be
curve	animation curve to animate the slide
OnSlideEnd	event to be called after sliding process ends

UIMotionSnap.SnapLeft

```
public static void SnapLeft(RectTransform content);

public static void SnapLeft(RectTransform content, float offsetRateOverScreenWidth);

public static void SnapLeft(RectTransform content, AnimationCurve curve);

public static void SnapLeft(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve);

public static void SnapLeft(RectTransform content, UnityEvent OnSnapEnd);

public static void SnapLeft(RectTransform content, float offsetRateOverScreenWidth, UnityEvent OnSnapEnd);

public static void SnapLeft(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);

public static void SnapLeft(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenWidth	how much screen width-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapRight

```
public static void SnapRight(RectTransform content);
public static void SnapRight(RectTransform content, float offsetRateOverScreenWidth);
public static void SnapRight(RectTransform content, AnimationCurve curve);
public static void SnapRight(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve);
public static void SnapRight(RectTransform content, UnityEvent OnSnapEnd);
public static void SnapRight(RectTransform content, float offsetRateOverScreenWidth, UnityEvent OnSnapEnd);
public static void SnapRight(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);
public static void SnapRight(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenWidth	how much screen width-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapUp

```
public static void SnapUp(RectTransform content);
public static void SnapUp(RectTransform content, float offsetRateOverScreenHeight);
public static void SnapUp(RectTransform content, AnimationCurve curve);
public static void SnapUp(RectTransform content, float offsetRateOverScreenHeight, AnimationCurve curve);
public static void SnapUp(RectTransform content, UnityEvent OnSnapEnd);
public static void SnapUp(RectTransform content, float offsetRateOverScreenHeight, UnityEvent OnSnapEnd);
public static void SnapUp(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);
public static void SnapUp(RectTransform content, float offsetRateOverScreenHeight, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenHeight	how much screen height-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapDown

```
public static void SnapDown(RectTransform content);
public static void SnapDown(RectTransform content, float offsetRateOverScreenHeight);
public static void SnapDown(RectTransform content, AnimationCurve curve);
public static void SnapDown(RectTransform content, float offsetRateOverScreenHeight, AnimationCurve curve);
public static void SnapDown(RectTransform content, UnityEvent OnSnapEnd);
public static void SnapDown(RectTransform content, float offsetRateOverScreenHeight, UnityEvent OnSnapEnd);
public static void SnapDown(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);
public static void SnapDown(RectTransform content, float offsetRateOverScreenHeight, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenHeight	how much screen height-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapOutLeft

```
public static void SnapOutLeft(RectTransform content);

public static void SnapOutLeft(RectTransform content, float offsetRateOverScreenWidth);

public static void SnapOutLeft(RectTransform content, AnimationCurve curve);

public static void SnapOutLeft(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve);

public static void SnapOutLeft(RectTransform content, UnityEvent OnSnapEnd);

public static void SnapOutLeft(RectTransform content, float offsetRateOverScreenWidth, UnityEvent OnSnapEnd);

public static void SnapOutLeft(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);

public static void SnapOutLeft(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenWidth	how much screen width-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapOutRight

```
public static void SnapOutRight(RectTransform content);
public static void SnapOutRight(RectTransform content, float offsetRateOverScreenWidth);
public static void SnapOutRight(RectTransform content, AnimationCurve curve);
public static void SnapOutRight(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve);
public static void SnapOutRight(RectTransform content, UnityEvent OnSnapEnd);
public static void SnapOutRight(RectTransform content, float offsetRateOverScreenWidth, UnityEvent OnSnapEnd);
public static void SnapOutRight(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);
public static void SnapOutRight(RectTransform content, float offsetRateOverScreenWidth, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenWidth	how much screen width-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapOutUp

```
public static void SnapOutUp(RectTransform content);
public static void SnapOutUp(RectTransform content, float offsetRateOverScreenHeight);
public static void SnapOutUp(RectTransform content, AnimationCurve curve);
public static void SnapOutUp(RectTransform content, float offsetRateOverScreenHeight, AnimationCurve curve);
public static void SnapOutUp(RectTransform content, UnityEvent OnSnapEnd);
public static void SnapOutUp(RectTransform content, float offsetRateOverScreenHeight, UnityEvent OnSnapEnd);
public static void SnapOutUp(RectTransform content, AnimationCurve curve, UnityEvent OnSnapEnd);
public static void SnapOutUp(RectTransform content, float offsetRateOverScreenHeight, AnimationCurve curve, UnityEvent OnSnapEnd);
```

Parameters

content	content that will be snapped
offsetRateOverScreenHeight	how much screen height-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionSnap.SnapOutDown

public static void **SnapOutDown**(RectTransform **content**);

public static void **SnapOutDown**(RectTransform **content**, float **offsetRateOverScreenHeight**);

public static void **SnapOutDown**(RectTransform **content**, AnimationCurve **curve**);

public static void **SnapOutDown**(RectTransform **content**, float **offsetRateOverScreenHeight**, AnimationCurve **curve**);

public static void **SnapOutDown**(RectTransform **content**, UnityEvent **OnSnapEnd**);

public static void **SnapOutDown**(RectTransform **content**, float **offsetRateOverScreenHeight**, UnityEvent **OnSnapEnd**);

public static void **SnapOutDown**(RectTransform **content**, AnimationCurve **curve**, UnityEvent **OnSnapEnd**);

public static void **SnapOutDown**(RectTransform **content**, float **offsetRateOverScreenHeight**, AnimationCurve **curve**, UnityEvent **OnSnapEnd**);

Parameters

content	content that will be snapped
offsetRateOverScreenHeight	how much screen height-long space will be left between content and target
curve	animation curve to animate the snap
OnSnapEnd	event to be called after snapping process ends

UIMotionScale.ScaleToOne

```
public static void ScaleToOne(RectTransform content);
public static void ScaleToOne(RectTransform content, AnimationCurve curve);
public static void ScaleToOne(RectTransform content, UnityEvent OnScaleEnd);
public static void ScaleToOne(RectTransform content, AnimationCurve curve, UnityEvent OnScaleEnd);
```

Parameters

content	content that will be scaled
curve	animation curve to animate the scale
OnScaleEnd	event to be called after scaling process ends

UIMotionScale.ScaleToZero

```
public static void ScaleToZero(RectTransform content);

public static void ScaleToZero(RectTransform content, AnimationCurve curve);

public static void ScaleToZero(RectTransform content, UnityEvent OnScaleEnd);

public static void ScaleToZero(RectTransform content, AnimationCurve curve, UnityEvent OnScaleEnd);
```

Parameters

content	content that will be scaled
curve	animation curve to animate the scale
OnScaleEnd	event to be called after scaling process ends

UIMotionScale.ScaleBy

```
public static void ScaleBy(RectTransform content, float scaleFactor);

public static void ScaleBy(RectTransform content, float scaleFactor, AnimationCurve curve);

public static void ScaleBy(RectTransform content, float scaleFactor, UnityEvent OnScaleEnd);

public static void ScaleBy(RectTransform content, float scaleFactor, AnimationCurve curve, UnityEvent OnScaleEnd);
```

Parameters

content	content that will be scaled
scaleFactor	size multiplier for scaling
curve	animation curve to animate the scale
OnScaleEnd	event to be called after scaling process ends