RECT INTERFACE

View online: https://www.construct.net/en/make-games/manuals/addon-sdk/reference/geometry-interfaces/rect

---

The `Rect` interface represents an axis-aligned rectangle in the SDK. It can also be constructed independently as a general-purpose geometry class.

Constructor

```
new SDK.Rect();
new SDK.Rect(left, top, right, bottom);
```

A `Rect` can be constructed with no parameters, which defaults all co-ordinates to zero, or with given positions for the left, top, right and bottom positions.

Methods

---

set(left, top, right, bottom)

Set all sides of the rectangle in one call.

---

copy(rect)

Set all sides of the rectangle by copying another `SDK.Rect`.

---

clone()

Return a new instance of an `SDK.Rect` with identical values to this one.

---

setLeft(left)
setTop(top)
setRight(right)
setBottom(bottom)

Set the position of each side of the rectangle individually.

---

getLeft()
getTop()
getRight()
getBottom()

Get the positition of each side of the rectangle individually.

---

width()
height()

Get the width or height of the rectangle. Note if the right edge is to the left of the left edge, or the bottom edge above the top edge, this will return a negative size.

---

midX()
midY()

Return the average of the left and right, or top and bottom, positions.

---

offset(x, y)

Add `x` to the left and right positions, and `y` to the top and bottom positions, offsetting the entire rectangle.

---

inflate(x, y)
deflate(x, y)

Expand or shrink the rectangle using the given offsets. Inflating subtracts from the left and top edges and adds to the right and bottom edges, and deflating does the opposite.

---

multiply(x, y)
divide(x, y)

Multiply or divide each position by a given factor on each axis.

---

clamp(left, top, right, bottom)

Clamp each position in the rectangle to a given value, ensuring the rectangle does not extend beyond the bounds of the passed rectangle.

---

normalize()

Normalize the rectangle positions, swapping the left-right positions if the right position is on the left, and swapping the top-bottom positions if the bottom position is on the top. This ensures the width and height are positive.

---

intersectsRect(rect)

Test for an intersection with another `SDK.Rect`, returning a boolean indicating if it intersects.

---

containsPoint(x, y)

Test if the given point is inside the bounds of this rectangle, returning a boolean.