Vector/Vector2

Contents

Methods

create

This is the default constructor for the Vector2 class and returns a Vector2 object.

Syntax

```
vector2 Vector2 ( mixed vectorOrX [, float y ] )
```

Required Arguments

- **vectorOrX**: Vector2, Table or Float indicating vector's coordinates

Optional Arguments

- **y**: if *vectorOrX* is a float, this will indicate the vector's Y coordinate

Example

This example checks if the player is using a low resolution.

client

```
addEventHandler ( "onClientResourceStart", resourceRoot, function()
        local screenSize = Vector2(guiGetScreenSize())
        if screenSize.x < 1360 and screenSize.y < 768 then
                outputChatBox ("You are running on a low resolution")
        end
end)
```

normalize

This function normalizes the vector

Syntax

```
bool Vector2.normalize ( vector2 vector )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Normalizes a vector*
> **Method**: *Vector2:normalize(...)*

Required Arguments

- **vector**: Vector2 to normalize

Example

This example demonstrates how to draw a line that is 200 pixels long and red. The line is drawn from the center of the screen to the mouse cursor's position.

client

```
local screenW, screenH = guiGetScreenSize() -- get screen size

local centerVec = Vector2(screenW / 2, screenH / 2) -- get screen center point
local mouseVec = Vector2() -- predefined variable that will change later

local lineLength = 200 -- sets length of the drawn line
local lineColor = tocolor(255, 0, 0) -- set line color

showCursor(true) -- show cursor

addEventHandler('onClientRender', root, function()
        local curX, curY = getCursorPosition()
        mouseVec.x, mouseVec.y = curX * screenW, curY * screenH -- get absolute cursor position

        local dVec = Vector2(mouseVec - centerVec) -- get the distance between the mouse position and the center of the screen
        local scaleFactor = lineLength / dVec.length

        mouseVec = centerVec + (dVec * scaleFactor)

        dxDrawLine(centerVec.x, centerVec.y, mouseVec.x, mouseVec.y, color)
end)
```

getX

This function gets the X coordinate of a vector.

Syntax

```
float Vector2.getX ( vector2 vector )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Gets X coordinate of a vector*
> **Method**: *Vector2:getX(...)*
> **Variable**: *.x*

Required Arguments

- **vector**: Vector2 to get X coordinate from

getY

This function gets the Y coordinate of a vector.

Syntax

```
float Vector2.getY ( vector2 vector )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Gets Y coordinate of a vector*
> **Method**: *Vector2:getY(...)*
> **Variable**: *.y*

Required Arguments

- **vector**: Vector2 to get Y coordinate from

setX

This function sets the X coordinate of a vector.

Syntax

```
bool Vector2.setX ( vector2 vector, float x )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Sets X coordinate of a vector*
> **Method**: *Vector2:setX(...)*
> **Variable**: *.x*

Required Arguments

- **vector**: Vector2 to set X coordinate from
- **x**: New X coordinate

setY

This function sets the Y coordinate of a vector.

Syntax

```
bool Vector2.setY ( vector2 vector, float y )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Sets Y coordinate of a vector*
> **Method**: *Vector2:setY(...)*
> **Variable**: *.y*

Required Arguments

- **vector**: Vector2 to set Y coordinate from
- **Y**: New Y coordinate

getNormalized

This function gets a normalized vector.

Syntax

```
vector2 Vector2.getNormalized ( vector2 vector )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Gets a normalized vector*
> **Method**: *Vector2:getNormalized(...)*
> **Variable**: *.normalized*

Required Arguments

- **vector**: Vector2 to get normalized version of

getLength

This function gets the length of a vector.

Syntax

```
vector2 Vector2.getLength ( vector2 vector )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Gets the length of a vector*
> **Method**: *Vector2:getLength(...)*
> **Variable**: *.length*

Required Arguments

- **vector**: Vector2 to get length from

getSquaredLength

This function gets the squared length of a vector.

Syntax

```
vector2 Vector2.getSquaredLength ( vector2 vector )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Gets the squared length of a vector*
> **Method**: *Vector2:getSquaredLength(...)*
> **Variable**: *.squaredLength*

Required Arguments

- **vector**: Vector2 to get squared length from

dot

This function gets the dot product of two vectors.

Syntax

```
vector2 Vector2.dot ( vector2 vectorOne, vector2 vectorTwo )
```

OOP Syntax Help! I don't understand this!

> **Note**: *Gets the dot product of two vectors*
> **Method**: *Vector2:dot(...)*

Required Arguments

- **vectorOne**: First Vector2 to calculate the dot product of
- **vectorTwo**: Second Vector2 to calculate the dot product of