

Discover Packages > Standard library > image > draw 

draw







package


standard library

Version: [go1.20.1](#) **Latest** | Published: Feb 14, 2023 | License: [BSD-3-Clause](#) | Imports: 3 |

Imported by: 5,970

Details

 Valid [go.mod](#) file   Redistributable license   Tagged version 

 Stable version 

[Learn more](#)

Repository

cs.opensource.google/go/go

Links

 [Report a Vulnerability](#)

 Documentation 

<> Documentation

Overview

Package draw provides image composition functions.

See "The Go image/draw package" for an introduction to this package:

https://golang.org/doc/articles/image_draw.html

Index

`func Draw(dst Image, r image.Rectangle, src image.Image, sp image.Point, op Op)`

`func DrawMask(dst Image, r image.Rectangle, src image.Image, sp image.Point, ...)`

`type Drawer`

`type Image`

`type Op`

`func (op Op) Draw(dst Image, r image.Rectangle, src image.Image, sp image.Point)`

`type Quantizer`

`type RGBA64Image`

Examples

[Drawer \(FloydSteinberg\)](#)

Constants

This section is empty.

Variables

This section is empty.

Functions

func Draw

```
func Draw(dst Image, r image.Rectangle, src image.Image, sp image.Point, op Op)
```

Draw calls DrawMask with a nil mask.

func DrawMask

```
func DrawMask(dst Image, r image.Rectangle, src image.Image, sp image.Point, mask image.Image, mp image.Point, op Op)
```

DrawMask aligns r.Min in dst with sp in src and mp in mask and then replaces the rectangle r in dst with the result of a Porter-Duff composition. A nil mask is treated as opaque.

Types

type Drawer

added in go1.2

```
type Drawer interface {  
    // Draw aligns r.Min in dst with sp in src and then replaces the  
    // rectangle r in dst with the result of drawing src on dst.  
    Draw(dst Image, r image.Rectangle, src image.Image, sp image.Point)  
}
```

Drawer contains the Draw method.

► Example (FloydSteinberg)

```
var FloydSteinberg Drawer = floydSteinberg{}
```

FloydSteinberg is a Drawer that is the Src Op with Floyd-Steinberg error diffusion.

type Image

```
type Image interface {  
    image.Image  
    Set(x, y int, c color.Color)  
}
```

Image is an image.Image with a Set method to change a single pixel.

type Op

```
type Op int
```

Op is a Porter-Duff compositing operator.

```
const (  
    // Over specifies "(src in mask) over dst".  
    Over Op = iota  
    // Src specifies "src in mask".  
    Src  
)
```

func (Op) [Draw](#)

added in go1.2

```
func (op Op) Draw(dst Image, r image.Rectangle, src image.Image, sp image.Point)
```

Draw implements the Drawer interface by calling the Draw function with this Op.

type [Quantizer](#)

added in go1.2

```
type Quantizer interface {  
    // Quantize appends up to cap(p) - len(p) colors to p and returns the  
    // updated palette suitable for converting m to a paletted image.  
    Quantize(p color.Palette, m image.Image) color.Palette  
}
```

Quantizer produces a palette for an image.

type [RGBA64Image](#)

added in go1.17

```
type RGBA64Image interface {  
    image.RGBA64Image  
    Set(x, y int, c color.Color)  
    SetRGBA64(x, y int, c color.RGBA64)  
}
```

RGBA64Image extends both the Image and image.RGBA64Image interfaces with a SetRGBA64 method to change a single pixel. SetRGBA64 is equivalent to calling Set, but it can avoid allocations from converting concrete color types to the color.Color interface type.



Source Files

[View all](#) [↗](#)

[draw.go](#)

Why Go

[Use Cases](#)

[Case Studies](#)

Get Started

[Playground](#)

[Tour](#)

[Stack Overflow](#)

[Help](#)

Packages

[Standard Library](#)

[About Go Packages](#)

About

[Download](#)

[Blog](#)

[Issue Tracker](#)

[Release Notes](#)

[Brand Guidelines](#)

[Code of Conduct](#)

Connect

[Twitter](#)

[GitHub](#)

[Slack](#)

[r/golang](#)

[Meetup](#)

[Golang Weekly](#)

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)



Google