ASCII Art

I Faire un dessin avec Golang : transformer des images (rgb) en caractères

Vidéo explicative : <https://www.youtube.com/watch?v=vY5Q_T_ouk4&t=58s>

Unicode block : <https://en.wikipedia.org/wiki/Block_Elements>

* u2001 : unicode de l’espace : laissera un bloc vide
* Niveaux de gris codés en uint8 ( 1 octet : 8bits : 0 à 255)
* RGBa : uint16 (2 octets : 16bits : 0 à 65.535) stocké dans un uint32 (4 octets : 32bits 0 à 4.294.967.295)
* uint8(r/257) : valeur comprise entre 0 et 255
* (r\*0.299) + (g\*0.587) + (b\*0.114) : ce qui se rapproche le plus d’une conversion fidèle à l’image en terme de luminosité d’après l’internet

1. Créer un dossier

* Il contient
* ***main.go***
* l’image à transformer en caractères
* Dans ce dossier on en crée un autre ASCIIconvert qui contient
* ***asciiconvert.go***

1. Déclarer le package de main.go

* *package main*

1. Créer une fonction main pour déclarer/décoder l’image pour créer un objet image

* *func main () {*

*file, \_ := os.Open(‘’image.jpg’’)*

*defer file.Close()*

*Img, \_ := jpeg.Decode(file)*

*}*

1. Déclarer le package asciiConvert dans le fichier du même nom

* *package asciiconvert*

1. Créer la fonction convertToGrey qui a pour paramètre un objet c qui appartient au package image color

* *func convertToGrey(c color.Color) uint8 {*

*r,g,b, \_ := c.RGBA()*

*red := float32(uint8(r/257))\*0.299*

*green := float32(uint8(g/257))\*0.587*

*blue := float32(uint(g/257))\*0.114*

*grey := red + green + blue*

*return unint8(grey)*

*}*

1. Créer une function asciiInvert dans le fichier asciiconvert.go pour associer une chaine de charactères à grey en fonction de sa luminosité et une fonction asciiRegular qui sera son opposé

* *func asciiInvert(grey uint8) string {*

*switch {*

*case grey < 51 : return ‘’\u2001’’*

*case grey >= 51 && grey < 102 : return ‘’\u2591’’*

*case grey >= 102 && grey < 153 : return ‘’\u2592’’*

*case grey >= 153 && grey < 204 : return ‘’\u2593’’*

*case grey >= 204 : return ‘’\u2588’’*

*default : return ‘’ ‘’*

*}*

*}*

* *func asciiRegular(grey uint8) string {*

*switch {*

*case grey < 51 : return ‘’\u2588’’*

*case grey >= 51 && grey < 102 : return ‘’\u2593’’*

*case grey >= 102 && grey < 153 : return ‘’\u2592’’*

*case grey >= 153 && grey < 204 : return ‘’\u2591’*

*case grey >= 204 : return ‘’\u2001’’*

*default : return ‘’ ‘’*

*}*

*}*

* Dans ce cas là on utilise 5 caractères différents pour 5 niveaux de gris différents, c’est pourquoi on a divisé 255 par 5 ce qui donne 51, 102, 153, 204, 255 mais pour être plus précis on pourrait en mettre beaucoup plus
* Missing return : le switch ne sait pas qu’on à défini tous les cas existants, il faut lui déclarer par un default : return ‘’ ‘’, une chaîne vide

1. Créer la fonction asciiToConsole dans asciiconvert.go : adapter les chaines de caractères à l’image

* *func AsciiToConsole (img image.Image, reduc int) {*

*for h := 0; h < img.Bounds().Dy(); h += reduc {*

*for w := 0; w < img.Bounds().Dx(); w += reduc {*

*pixel := img.At(w,h)*

*character := asciiInvert(convertToGrey(pixel))*

*fmt.Printf(‘’%s%s’’, character, character)*

*}*

*Fmt.Print(‘’\n’’)*

*}*

*}*

* Image en pixel (carré) et les chaines de caractères sont des rectangles
* Donc on va remplacer un pixel par 2 caractères pour que ces caractères forment un rectangle
* Faire une boucle for ***h*** (height) ***h := 0*** avec count ***h*** ***< img.Bounds().Dy()*** et ***h += reduc*** (reduc = 1 <-> h++ sinon on va compter de 2 en 2 : pixel 1-3-5-7..)
* Réduire arithmétiquement l’itération sur l’image (nb de pixels)
* Faire une autre boucle for ***w*** (width) ***w := 0***, avec count ***w < img.Bounds().Dx()*** et w += reduc
* ***pixel := img.At(w, h)***, à chaque itération on récupère le pixel qui lui correspond, pixel renvoie à un objet ***color***
* ***character := asciiInvert(convertToGrey(pixel))*** on récupère le caractère issu du package asciiInvert qui va convertir les pixels en échelle de gris
* ***fmt.Printf(‘’%s%s’’, character, character)***: %s%s (2 caractères sous forme de string)
* ***fmt***.Print(‘’\n’’) : créer une nouvelle ligne

1. Importer le package sciiconvert dans main.go

* *import asc ‘’./asciiconvert’’*

1. Ajouter la fonction AsciiToConsole dans la fonction main

* *func main () {*

*file, \_ := os.Open(‘’image.jpg’’)*

*defer file.Close()*

*Img, \_ := jpeg.Decode(file)*

*Asc.AsciiToConsole(img, 2)*

*}*

* 2 : facteur de réduction

1. Créer fonction AsciiToFile dans asciiconvert.go

* *func AsciiToFile(img, image.Image, reduc int, filename string ) {*

*name := fmt.Sprintf(“%s.text”, filename)*

*file, \_ := os.Create(name)*

*defer file.Close()*

*for h := 0; h < img.Bounds().Dy(); h += reduc {*

*for w := 0; w < img.Bounds().Dx(); w += reduc {*

*pixel := img.At(w,h)*

*character := asciiInvert(convertToGrey(pixel))*

*asciiChar := fmt.Sprintf(‘’%s%s’’, character, character)*

*file.WriteString(asciiChar)*

*}*

*File.WriteString(‘’\n’’)*

*}*

*}*

* ***name := fmt.Sprintf(“ %s.txt “, filename)*** : nom de fichier de destination
* ***file, \_ := os*** : créer le fichier de destination