note @4002 **567** views

Python program to generate solid pixel art

Below is a Python 3 script that I made that can read in an image file an convert it into commands for A4Q3. I am not sure if it is allowed to post it (hence the private), but here is the code and two examples with the three images being the

1. Python generated resulting image
2. A4Q3 output
3. Source file

It does not create the best images when they are using colours far away from the red, green and blue that the code does, but it is still fun to play around with.

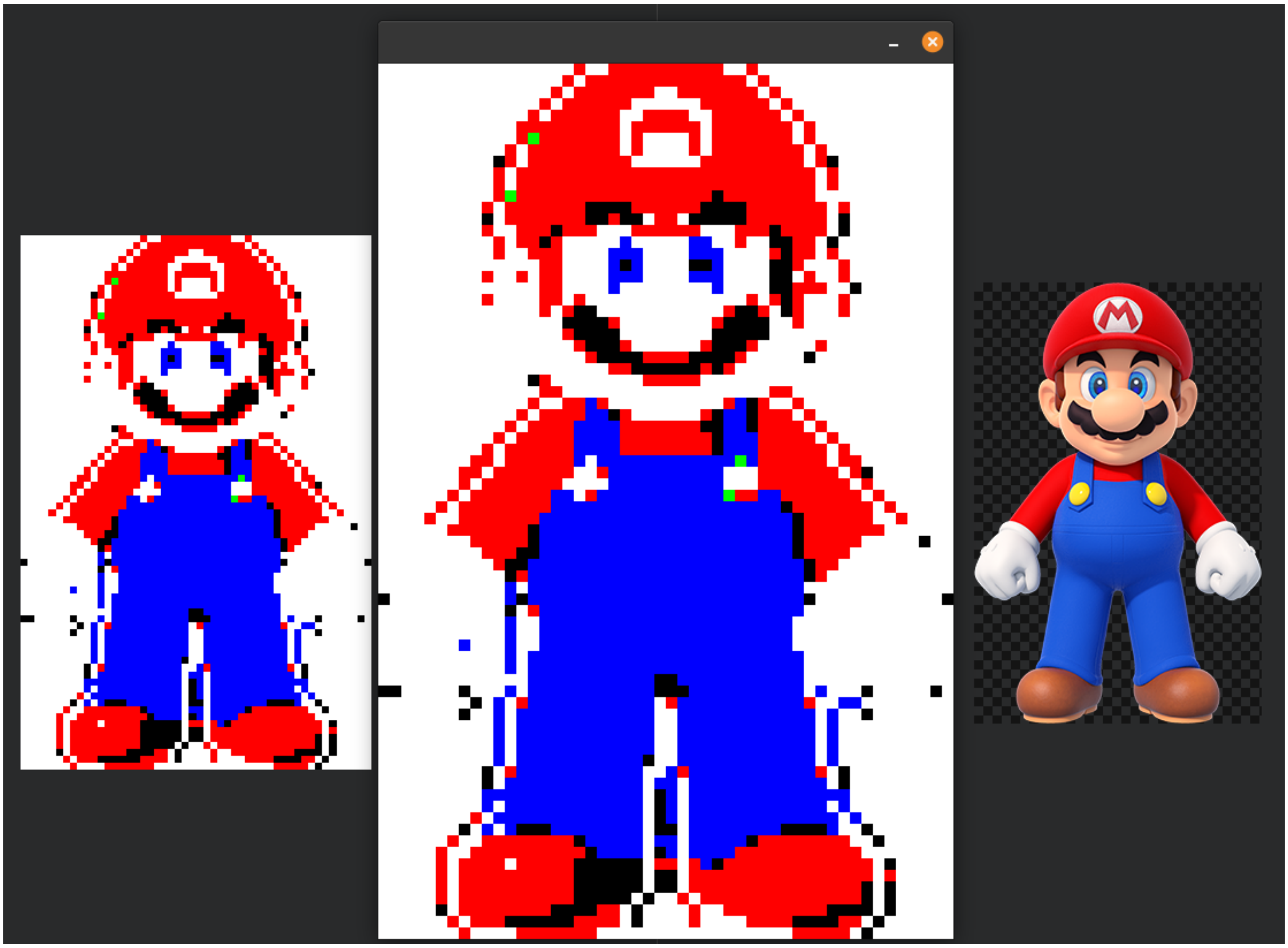
The "output" file that is generated contains the commands and the "output.png" contains the python interpretation of the image.

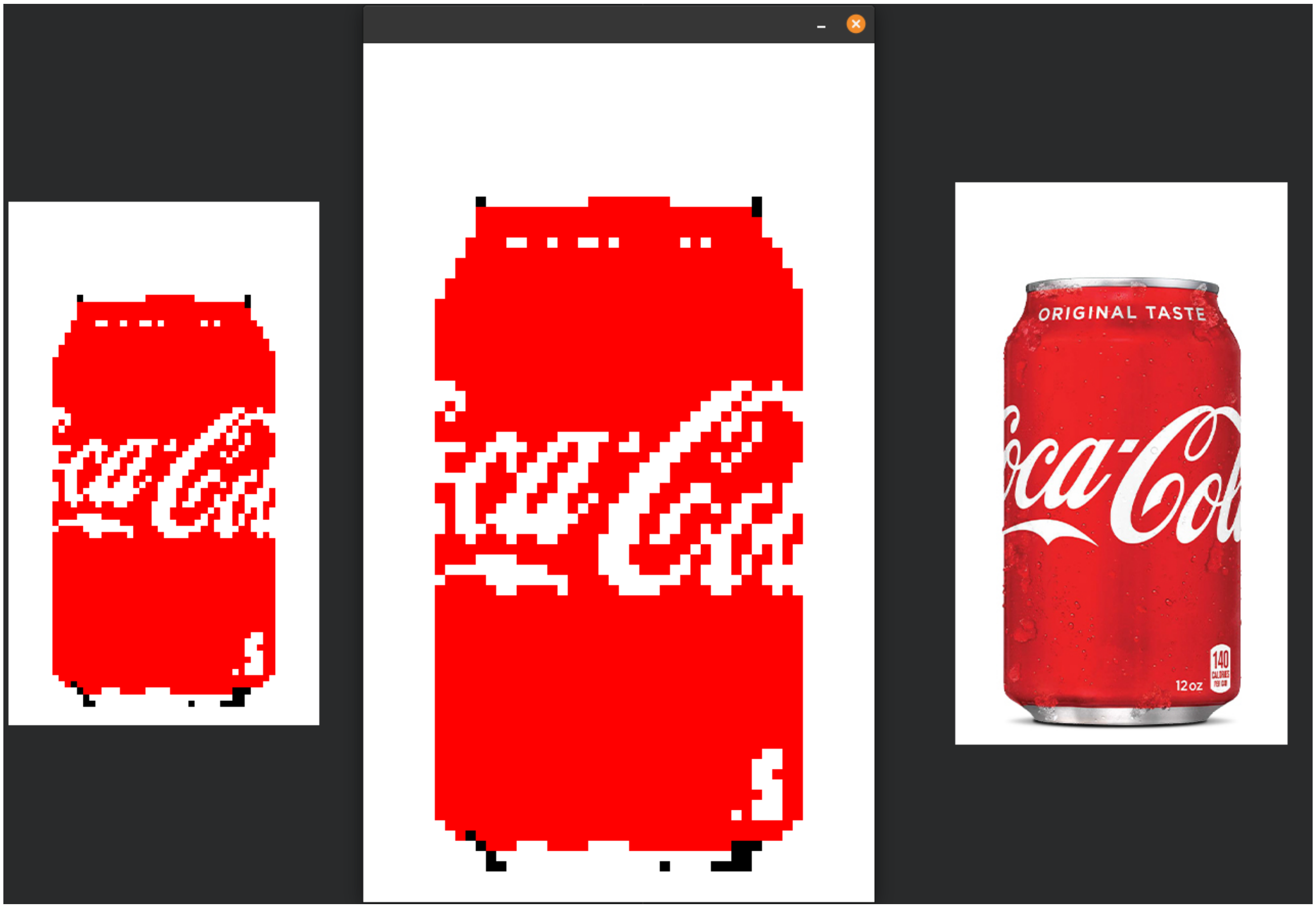
```
# =============== CONSTANTS ===============
FILENAME = "Mario.png" # Filename to read in
WIDTH = 50 # Max width of the output file - needs to be small or otherwise will generate tens
or thousands of commands
AVAILABLE=[(255,0,0), (0,255,0), (0,0,255), (255,255,255), (0,0,0)] # Colors to choose from -
must include black
OUTPUT=['a', 'A', '1', ' ', '!'] # Letter assignment to color
# =========================================
from PIL import Image
from math import sqrt

im = Image.open(FILENAME)
im = im.convert('RGBA')
width_scale = (WIDTH/float(im.size[0]))
height = int((float(im.size[1])*float(width_scale)))
im = im.resize((WIDTH,height), Image.ANTIALIAS)
im.save('scaled.png')
f = open("output", "w")
f.write(f"addgraphics 0 {height-1} 0 {WIDTH-1}\n")

pix = im.load()
for row in range(im.size[0]):
    for col in range(im.size[1]):
        lst = []
        for avail in AVAILABLE:
            diff = sqrt((avail[0]-pix[row,col][0])**2+(avail[1]-pix[row,col][1])**2+(avail[2]-
pix[row,col][2])**2)
            lst.append(diff/sqrt((255)**2+(255)**2+(255)**2))
        if pix[row,col] == 0:
            pix[row,col]=0
        elif pix[row,col][3] == 0:
            pix[row,col]=(255,255,255)
        else:
            pix[row,col]=AVAILABLE[lst.index(min(lst))]
            if lst.index(min(lst)) != 3:
                f.write(f"filledbox {col} {col} {row} {row} {OUTPUT[lst.index(min(lst))]}\n")
```

```
f.write("render\n")
f.close()
im.save('output.png')
```

run code snippet

Visit ‘Manage Class’ to disable runnable code snippets ×





Edit: Fixed minor error in script

assignments assignments/a4 assignments/a4/dd2

~ An instructor (Caroline Kierstead) thinks this is a good note ~

Updated 14 days ago by Oliver Odendaal and Caroline Kierstead

followup discussions *for lingering questions and comments*

● Resolved ○ Unresolved

Caroline Kierstead 15 days ago
Sweet!

good comment | 6

Caroline Kierstead 14 days ago And lets make it public, so everyone can appreciate this awesome work!

good comment | 3

● Resolved ○ Unresolved

Peter Jiang 14 days ago
😮 time to animate bad apple on A4Q3

helpful! | 14

Bilal Khan 14 days ago time to print every frame of the bee movie through Xwindows helpful! | 12

Stephen Cao (Anon. Gear to classmates) 13 days ago dun dun dun dudududun dun dun

helpful! | 1

Aryan Sureka 10 days ago Time to Rick roll the entire class in pixels

helpful! | 1