Program na přípravu obrázků pro Voxel Print

Nejdříve je potřeba vytvořit data a zazipovat je do souboru `data.zip`, ten si nyní nahrajeme.

Smazání starých dat

```
!rm -r -f /content/*
```

Nahrání vstupních dat

```
from google.colab import files
from PIL import Image
import os

uploaded = files.upload()
```

Zvolit soubory Soubor nevybrán Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving data.zip to data.zip

```
for fn in uploaded.keys():
    print('Nahrál se soubor "{name}" s délkou {length} B'.format(name=fn, length=len(uploa
```

```
Nahrál se soubor "data.zip" s délkou 36078904 B
```

Dekomprimuj zip

```
!unzip /content/data.zip -d /content/data/
```

Převod všech barev pouze na níže definované barvy

```
def cga_quantize(image):
    pal_image= Image.new("P", (1,1))
    pal_image.putpalette( (
          0,0,0,        # Černé pozadí
          0,137,166,    # VeroCY-V
          198,0,88,     # VeroMGT-V
          240,197,0     # VeroYL-V
        ) + (255,255,255)*251)
    return image.convert("RGB").quantize(palette=pal_image)
```

Spuštění převodu a uložení do složky

```
dir = "/content/data/"
```

```
save_dir = "/content/converted/"
!mkdir -p /content/converted/
!rm -f /content/converted/*


for fn in os.listdir(dir):
    if fn.endswith(".png"):
        image = Image.open(dir + fn).convert('RGB')
        converted = cga_quantize(image)
        converted.save(save_dir + fn)
```

Pozadí je bílé --> vnitřní pixely změníme na jinou barvu.

Ve zvoleném obrázku změň všechny pixely dané barvy na barvu jinou, pokud má daný pixel více než n sousedních pixelů jiné než původní barvy

```
def colorIfSurround(image_to_transform, ignore_color, mark_color, n):
  # n = minimální počet sousedů s jinou než zvolenou barvou
    output_image=image_to_transform.copy().convert('RGB')
    for x in range(output_image.width):
        for y in range(output_image.height):
            pocet_sousedu = 0
            # pozor na krajní hodnoty abychom nešahali mimo obrázek
            if (output_image.getpixel((x,y)) == ignore_color):
                if (1 < x < output_image.width-1) and (1 < y < output_image.height-1):
                    if output_image.getpixel((x-1,y  )) != ignore_color:
                        pocet_sousedu += 1
                    if output_image.getpixel((x  ,y-1)) != ignore_color:
                        pocet_sousedu += 1
                    if output_image.getpixel((x+1,y  )) != ignore_color:
                        pocet_sousedu += 1
                        if pocet_sousedu >= n:
                            output_image.putpixel((x,y), mark_color)
                            continue
                    if output_image.getpixel((x   ,y+1)) != ignore_color:
                        pocet_sousedu += 1
# pokud je počet obarvených sousedů větsi než "n", změň tento pixel na mark_color barvu
                    if pocet_sousedu >= n:
                        output_image.putpixel((x,y), mark_color)
    return output_image



bila = (255,255,255)
zelena = (0,255,0)  # mark_color

!mkdir -p /content/colored/
!rm -f /content/colored/*


for fn in os.listdir("/content/converted/"):
    if fn.endswith(".png"):
      image = Image.open("/content/converted/" + fn).convert('RGB')
```

```
      converted = colorIfSurround(image,bila,zelena,3)
      converted.save("/content/colored/" + fn)
```

Změna všech černých pixelů na mark color

```
def black(image_to_transform, old_color, new_color):
  output_image = image_to_transform.copy().convert('RGB')
  width = output_image.size[0]
  height = output_image.size[1]
  for x in range(0,width):
    for y in range(0,height):
        pixel = output_image.getpixel((x,y))
        if (pixel == old_color):
            output_image.putpixel((x,y),new_color)

  return output_image


!mkdir -p /content/black/
!rm -f /content/black/*


cerna = (0,0,0)
zelena = (0,255,0)

for fn in os.listdir("/content/colored/"):
    if fn.endswith(".png"):
      image = Image.open("/content/colored/" + fn).convert('RGB')
      converted = black(image,cerna,zelena)
      converted.save("/content/black/" + fn)
```

Přidání okrajového rámečku

```
def border(image_to_transform, border_color):
  output_image = image_to_transform.copy().convert('RGB')
  width = output_image.size[0]
  height = output_image.size[1]
  for x in range(0,width):
    output_image.putpixel((x,0),border_color)
    output_image.putpixel((x,height-1),border_color)

  for y in range(0,height):
    output_image.putpixel((0,y),border_color)
    output_image.putpixel((width-1,y),border_color)

  return output_image


!mkdir -p /content/border/
!rm -f /content/border/*


cervena = (255,0,0)
```

```
for fn in os.listdir("/content/black/"):
    if fn.endswith(".png"):
      image = Image.open("/content/black/" + fn).convert('RGB')
      converted = border(image,cervena)
      converted.save("/content/border/" + fn)
```

Zdvojení pixelů v ose X/Y

```
def zdvojenaOsa(image_to_transform):
    width, height = image_to_transform.size
    output_image=image_to_transform.copy().convert('RGB').resize((width*2,height),resample

    return output_image


!mkdir -p /content/zdvojenaOsa/
!rm -f /content/zdvojenaOsa/*


for fn in os.listdir("/content/border/"):
    if fn.endswith(".png"):
      image = Image.open("/content/border/" + fn).convert('RGB')
      print("Colored image size = ", image.size)
      converted = zdvojenaOsa(image)
      print("Zdvojen image size = ", converted.size)
      converted.save("/content/zdvojenaOsa/" + fn)
```

Aby byly různé barvy ve 3d modelu vidět udělám různé variace pruhů, kde se plné bavy proloží průhlendou výplní

```
def pruhyDiagonal(image_to_transform, ignore_color, mark_color, sirka_pruhledna, sirka_bar
    output_image=image_to_transform.copy().convert('RGB')
    for x in range(output_image.width):
        for y in range(output_image.height):
            if (output_image.getpixel((x,y)) != ignore_color):
              if((x+(y*posun)) % (sirka_pruhledna + sirka_barevna) <= sirka_pruhledna ):
                output_image.putpixel((x,y), mark_color)

    return output_image

def pruhyDiagonalRev(image_to_transform, ignore_color, mark_color, sirka_pruhledna, sirka_
  output_image=image_to_transform.copy().convert('RGB')
  for x in range(output_image.width):
      for y in range(output_image.height):
          if (output_image.getpixel((x,y)) != ignore_color):
            if((x-(y*posun)) % (sirka_pruhledna + sirka_barevna) <= sirka_pruhledna ):
              output_image.putpixel((x,y), mark_color)

  return output_image

def pruhyHor(image_to_transform, ignore_color, mark_color, sirka_pruhledna, sirka_barevna)
```

```
  output_image=image_to_transform.copy().convert('RGB')
  for x in range(output_image.width):
      for y in range(output_image.height):
          if (output_image.getpixel((x,y)) != ignore_color):
            if((x) % (sirka_pruhledna + sirka_barevna) <= sirka_pruhledna ):
              output_image.putpixel((x,y), mark_color)

  return output_image

def pruhyVer(image_to_transform, ignore_color, mark_color, sirka_pruhledna, sirka_barevna)
  output_image=image_to_transform.copy().convert('RGB')
  for x in range(output_image.width):
      for y in range(output_image.height):
          if (output_image.getpixel((x,y)) != ignore_color):
            if((y) % (sirka_pruhledna + sirka_barevna) <= sirka_pruhledna ):
              output_image.putpixel((x,y), mark_color)

  return output_image
```

Vytvorime slozku pro prevedene obrázku s **diagonálními pruhy**

```
!mkdir -p /content/pruhy_diagonal/
!rm -f /content/pruhy_diagonal/*


for fn in os.listdir("/content/colored/"):
    if fn.endswith(".png"):
      image = Image.open("/content/colored/" + fn).convert('RGB')
      converted = pruhyDiagonal(image,bila,zelena,20,5,1)
      converted.save("/content/pruhy_diagonal/" + fn)
```

Vytvorime slozku pro prevedene obrázku s **pruhy HOR**

```
!mkdir -p /content/pruhy_hor/
!rm -f /content/pruhy_hor/*


for fn in os.listdir("/content/colored/"):
    if fn.endswith(".png"):
      image = Image.open("/content/colored/" + fn).convert('RGB')
      converted = pruhyHor(image,bila,zelena,20,20)
      converted.save("/content/pruhy_hor/" + fn)
```

Vytvorime slozku pro prevedene obrázku s **pruhy VER**

```
!mkdir -p /content/pruhy_ver/
!rm -f /content/pruhy_ver/*


for fn in os.listdir("/content/colored/"):
```

```
    if fn.endswith(".png"):
      image = Image.open("/content/colored/" + fn).convert('RGB')
      converted = pruhyVer(image,bila,zelena,20,5)
      converted.save("/content/pruhy_ver/" + fn)
```

Vytvorime slozku pro prevedene obrázku s **mřížkou**

```
!mkdir -p /content/mrizka/
!rm -f /content/mrizka/*
```

```
for fn in os.listdir("/content/colored/"):
    if fn.endswith(".png"):
      image = Image.open("/content/colored/" + fn).convert('RGB')
      converted = pruhyHor(image,bila,zelena,3,3)
      converted = pruhyVer(converted,bila,zelena,3,3)
      converted.save("/content/mrizka/" + fn)
```

Vytvorime slozku pro prevedene obrázku s **tečky**

```
!mkdir -p /content/tecky/
!rm -f /content/tecky/*
```

```
for fn in os.listdir("/content/colored/"):
    if fn.endswith(".png"):
      image = Image.open("/content/colored/" + fn).convert('RGB')
      converted = pruhyDiagonal(image,bila,zelena,4,3,1)
      converted = pruhyDiagonalRev(converted,bila,zelena,4,3,1)
      converted.save("/content/tecky/" + fn)
```

Vytvoření souboru zip ze složek

```
!rm -f hotovo.zip
```

```
!zip hotovo.zip /content/converted/*
!zip hotovo.zip /content/colored/*
!zip hotovo.zip /content/mrizka/*
!zip hotovo.zip /content/pruhy_diagonal/*
!zip hotovo.zip /content/pruhy_hor/*
!zip hotovo.zip /content/pruhy_ver/*
!zip hotovo.zip /content/tecky/*
!zip hotovo.zip /content/zdvojenaOsa/*
#!zip hotovo.zip /content/black/*
#!zip hotovo.zip /content/border/*
```

Stažení zkonvertovaných dat

```
files.download('hotovo.zip')
```