

Leuphana University Lüneburg

“Leuphana Pixelart”

Stylized Pixelart with Turtle in Python

Report for the seminar

“Creative Applications of Python and Machine Learning”

Module “Agents and Interfaces”

by Sarah Haq

Summer semester 2023

written by Jessica Maier

Jessica.Maier@stud.leuphana.de

3044981

Digital Media

4th Semester

Initial Research and Brainstorming

First, I needed to decide what kind of artwork I wanted to create. I was really stuck from the beginning on the idea that I want to work with the code we learned from you, about creating pixel art. So I looked at a lot of images and tried to find inspiration for my own artwork. I really like the way reality is portrayed in a minimalistic and yet aesthetically fun and easy-looking way. And even though reality is broken down in little squares, you can still identify objects and themes. That is very fascinating conceptually. From a more practical perspective, I decided to find a reasonable amount of pixels, I could possibly note down for one image. But when I started sketching images with 50 x 50 pixels, I had difficulties with the minimalism, because now everything was too abstract. I actually draw a lot in my freetime, but never before in this pixel art style. So in the end, I decided to go for the more detailed 100 x 140 pixels, which was in retrospect maybe a bit too much. For the theme of the artwork, I did not want to go realistic, but rather cartoonish or like graffiti. I wanted it to feel like landscape art without distracting from the Leuphana building at the center. Also, I wanted to include the Leuphana building in the image, because I thought it would be fitting, in case this code would be demonstrated next semester and nonetheless I really like how the building looks and how it is hopefully recognizable even in this pixel state. I wanted the clouds and stars to add a dream-like appeal to the image. The black outlines and white highlights helped make the picture more 1990s cartoonish looking. That the sun looks a bit like a skull and like it melts over the clouds came from a mistake while drawing the original picture, but I liked the spooky vibe it added, so I left it like this. And at first, I wanted the picture to look like a landscape at night, but in the end, it feels more like a combination of sundown and night. The colors were supposed to be very bright and eye-catching.

Methodology and References

The majority of my code is from our third lecture, when you showed us how to create squares and rows with Turtle in Python. More specifically, I used the code from your github repository:

https://github.com/shaq31415926/creative-programming/blob/main/03Lecture/pixel_art_demo.py (last visited 13.09.2023).

So my first step was to go through the code we did in class again and get an understanding of the functions and commands. Then I decided on a color scheme. I only wanted to use a limited amount of colors that also Turtle supported, to not get confused, when noting them all down later. I chose orange, yellow, red, black and white. So what I did afterwards was to open a drawing software, called Procreate, on my iPad and create a canvas with the same size and pixel size I wanted to use for my Turtle Screen. With these settings I drew a picture with an Apple Pencil, based on the sketches I did during the research and brainstorming phase, placing one pixel after another, while simultaneously writing down every color I used for every row. So I essentially wrote down what you can now see in the `leuphana_pixelcolors` script as the matrix and it took a really long time. Then I edited the colors into lists and copied them row by row into PyCharm, creating the matrix in the end. I treated the matrix more as multiple lists in a bigger list and applied the same things to it as I would to a list. Now I just needed to add a function, so that I can call the colors essentially for my code later without having this really long list obstructing the simplicity of the main code. Now I could set up my Turtle and start creating the pixels and rows like we did in class. And I edited the iterations in the last part of the code, according to my usage of the color matrix. Then the code was finally completed and running with the desired outcome. There was only one problem left: The Turtle was extremely slow, it could take up to 12 minutes for the Turtle to complete the picture. So I looked for solutions to make it run through the picture faster, but even the fastest speed was still too slow to wait for it. So I researched how to completely turn off the animation and I found a short article, explaining `turtle.tracer()` and also `turtle.update()`. The article can be found here:

<https://medium.com/@kingkongcoder123/what-the-heck-the-turtle-tracer-and-turtle-update-90c9e91335d2> (last visited 13.09.2023).

These commands then really helped me to get more in control with the Turtle's speed. Based on this article, I tried out multiple things, the placement of the `turtle.update()` command and different speed levels and ended with the current

solution. One thing, I don't quite understand, is why the animation gets continuously slower towards the end of the image. In the end, I decided to leave it that way, as it still seemed to be the best option. This project was very fun and even though it took quite some time, I think the result is good, too. Next time I would try to find a way to not add the colors manually by hand but maybe automatically, maybe with the help of machine learning. Then I would be able to have even better and more detailed results.

So after I wrote the code, I checked if everything in this code was understandable and well readable. Then I went on to push it on github and edit the README file. It was very important to me to explain every part of the code and how to use it accordingly, because I then hope, if you or someone else tries out the code, it will all run smoothly and they can see the image, as it is supposed to be.