Queering Code Manual

Winnie Soon

Code poem

```
#!/bin/bash

echo unsettling structures
awk 'BEGIN {
while (structures -e power)
print "destabilising " binary++
}'
```

Code poem read aloud

shebang bin bash
echo unsettling structures
awk(ward) begin
while structures equal power
print destablising binary plus plus

Description

Queering Code is written in Bash, which is a command line interface and a scripting language developed in 1989 as a piece of free software for Unix Systems (Linux and Mac OS but it has been also ported to Windows). The bash script is a plain text file that contains a series of commands for computer execution. Queering Code is both a work of "software art" (software as artwork, not software to make an artwork) and a "codework"

(where the source code and critical writing operate together) produced to embody "queer code" as a form of code poetry. The piece explores the material and linguistic tensions of writing, reading and printing computer code.

<p style="text-align:center">Run me</p>

This section shows how to run a Bash script in your computer. For those who have bash and terminal/CLI experience, you can simply download the script [here](#)

1. Open your terminal

2. Type `mkdir queercode` (make a directory called queercode)

3. Type `cd queercode` (so you are now at the queercode directory)

4. Type `nano queercode.sh` (editing mode)

5. Copy and paste the script below (use mouse click copy and paste, but not shortcuts)

```
#!/bin/bash

echo unsettling structures
awk 'BEGIN {
while (structures -e power)
print "destabilising " binary++
}'
```

6. Press control+X (to save the file)

7. Type `Y`, then press Enter

8. In the terminal: Type `ls` (list the file in the directory)

9. Type `bash queercode.sh` (to run the bash script)

10. Press control+z (to stop the program)

You should see the text displays on the terminal like the below image:

Bash syntax in a nutshell

- `#!/bin/bash` : to instruct the computer system to use bash as a default shell, which is the LINUX command interpreter, to execute a defined script. (#! refers to shebang, which is the combination of the # (pound key) and ! (exclamation mark))

- `echo unsettling structures` : echo is a bash syntax which is used to instruct the computer to display a line of text/string on the screen. Here the text is "unsettling structures".

- `awk 'BEGIN {}` : awk and BEGIN are bash syntax. The intention for using both the syntax in the script is more to take the semantic dimension, and it reads as: awk(ward) begin. Functonally speaking, the two syntax are used for pattern (especially text parsing in sentences/files) recognization and processing.

- `while (structures -e power)` : while() is a conditional loop structure. This means that while the condition holds true, the program will keep executing the code. It usually comes with do and done syntax. Both structures and power are null variables. -e is an operator, that means equal (==), a comparing condition of the two variales: structures and power. The whole line reads as "while structures equal power."

- `print "destabilising " binary++` : this statement is within the while loop structure, and binary++ is a variable that will increment by 1 for each iteration. Therefore, the screen continuous and endlessly prints the line destabilising 1, destabilising 2, destabilising 3, destabilising 4, ...
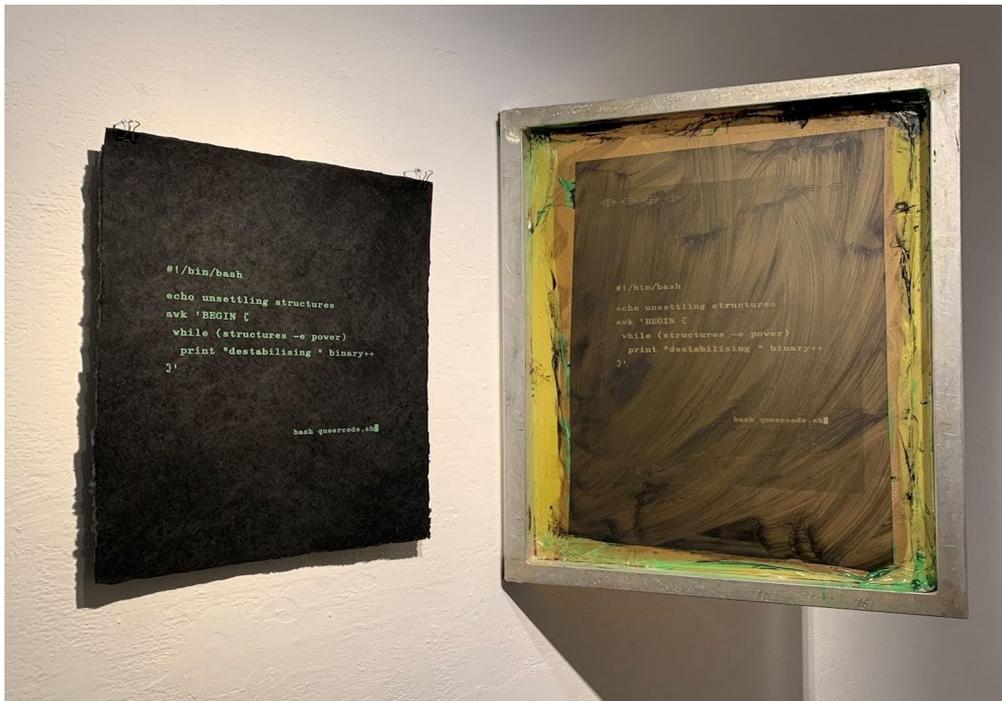

Image: Queering Code, in silkscreen print and silkscreen frame. 2022