

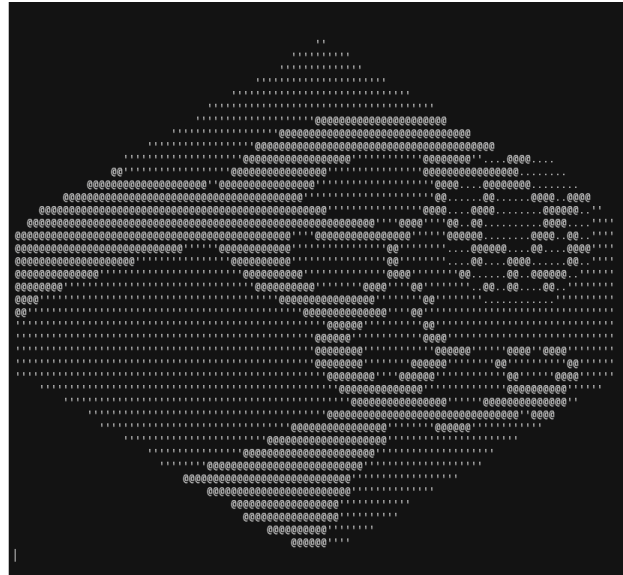
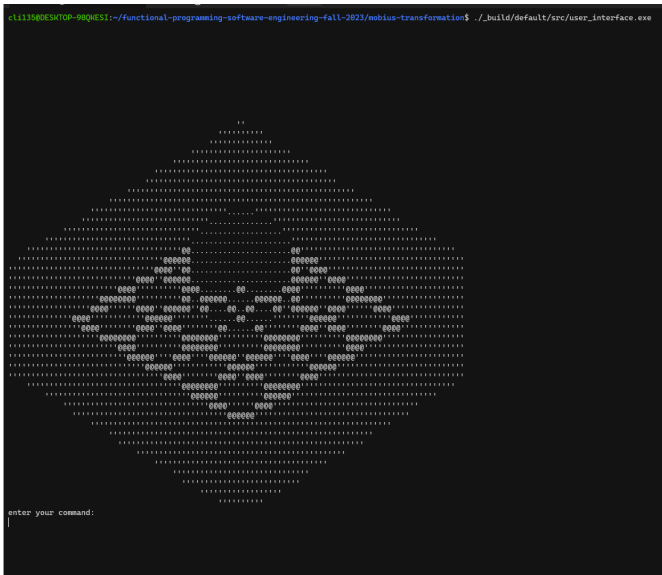
Date: 12-08-2023 (December 8, 2023)

Code Checkpoint README - Moebius Transformation in ASCII Art

Example run of the application upon startup:

```
./_build/default/src/user_interface.exe
```

cool



1. How to run the program / Usage:

- `dune b`
- `dune test`
- `./_build/default/src/user_interface.exe`
 - This command runs the program.

- You will then see an image displayed, with the option to enter commands interactively. Here is a list of commands that are accepted syntax:

How to use the user interface:

```
set [alpha/beta] [angle] : set alpha/beta to the input angle in degree
set alpha 90
set beta 180
```

```
add [alpha/beta] [angle] : increment current alpha/beta by the input angle in degree
add alpha 15
add beta -10
```

```
view [Sphere/Planar/Orthogonal] : change render views
view Sphere
```

```
set center [xfloat] [yfloat] [zfloat] : set the sphere center to a new location,
zfloat must be a positive value.
move center 0. 1. 3.
```

```
set [paramname] [paramvalue] : set all the customizable parameters for the viewport
set img_w 100
set view_size 4
set plane_bd 4
set half_edge_length 2
set line_w 0.25
set grid_size 2
set frame_rate 30
set duration 2.
```

```
cool : this will play a cool animation :)
```

```
reset: reset all parameters
```

```
exit: exit the program
```

2. A list of libraries we are using

- a. Core
- b. OUnit2
- c. [imagelib](#)

i. This library has been tested and is working now, with the functionality displayed below! Namely, we can

- 1. read from PNG files, displaying them as ASCII art, and we can
- 2. write to PNG files, displaying them the rasterizer output as normal PNG images (in `print_ascii.ml`)

ii. The results of writing to PNG files are shown below:



iii. And the results of reading from PNG files and rendering them as ASCII art is also shown below:



3. Codebase in `mobius-transformation/src` as of 12-08-2023:

- `rasterizer.mli`
 - `rasterizer.ml`
 - This performs the rasterization of the images for the Moebius transformation.
 - `math.mli`
 - `math.ml`
 - This is a math library that the `rasterizer.ml` uses.
 - `ascii_printer.mli`
 - `ascii_printer.ml`
 - This contains a function to print out a list of floats as an ASCII image.
 - `user_interface.mli`
 - `user_interface.ml`
 - This interactive executable handles the logic and syntax for the interactive user interface.
 - `print_ascii.mli`
 - `print_ascii.ml`
 - This interactive executable handles the reading from and writing to PNG files.
 - `dune`
-