

## Functionality

Turtle Graphics provides a method of drawing in the terminal. The instructions for the drawing are provided by premade lists of commands, consisting of any combination of the 6 command types. The drawing is achieved by a coordinate system tracking where a virtual pen tip is and printing characters as the pen draws.

There are two additional modes of Turtle Graphics Supplied;

- Simple mode: Ignores all colour commands and prints black foreground and white background
- Debug mode: prints a log of movement and draw coordinates to stderr as they are calculated. It is highly recommended to redirect this output to a file or another terminal.

## Initial State

The pen is initialised at the top left of the terminal window, represented by the (x,y) coordinates of (0,0) with a positive x representing movement to the right and a positive y representing movement towards the bottom of the terminal. The initial angle is 0 degrees, indicating a facing directly right. The initial foreground colour is 7 (white) and the initial background colour is 0 (black). Finally, the initial pattern is '+'.

## Available Commands

Turtle Graphics accepts the following commands;

1. FG [x]: Set the foreground colour of the pen to x (0-15)
2. BG [x]: Set the background colour of the pen to x (0-7)
3. MOVE [y]: Move the pen y units in current heading
4. DRAW [y]: Draws a line of y units the current pattern in current heading
5. ROTATE [y]: Rotates current heading anticlockwise y degrees
6. PATTERN [z]: Sets the pen's pattern to z.

Where x is an integer, y is a real number and z is any non-whitespace printable character.

## Input File

There is no limit to the number of commands an input file can contain, within the limits of your operating system to assign sufficient memory. The commands may be in uppercase, lowercase or any combination thereof. A single error in the file will cause the program to abort. Errors generally fit into 2 categories; missing information or incorrect information.

Missing information includes;

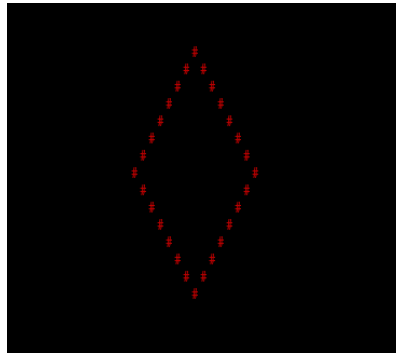
- An empty line
- A missing command name
- A missing parameter for the command

Incorrect information includes;

- A command name not one of the predefined 6
- An incorrect data type in a parameter (Note: x.0 is considered invalid for integer input)
- Any extra characters (including whitespace) after the parameter

### Sample Input and Output: (From Assignment sheet)

```
rotate -45
move 30
FG 1
Pattern #
DRAW 10
Rotate 90
draw 10
ROTATE 90
dRAW 10
ROTATE 90
DRAW 10
```



## Running Turtle Graphics

Turtle graphics has no UI and is controlled entirely through the command line and the input files. To run Turtle Graphics from the directory it is in use the command;

```
./TurtleGraphics x
```

To run simple mode;

```
./TurtleGraphicsSimple x
```

To run debug mode;

```
./TurtleGraphicsDebug x [2>y]
```

Where x is the file name containing the drawing commands. [2>y] is an optional and recommended argument to send the stderr output to a new location, where y is a location.

## Logging

Turtle Graphics keeps a log of all of the coordinates used in draw and move commands. Whenever the program executes, graphics.log is appended with a separator "---" indicating a new execution and then the coordinates in the form

Command type (x<sub>1</sub>,y<sub>1</sub>)-(x<sub>2</sub>,y<sub>2</sub>)

Where <sub>1</sub> signifies starting coordinate and <sub>2</sub> signifies the end coordinate.

### Logging Sample

A log provided by the commands in the sample input.

```
---
MOVE ( 0.000, 0.000)-( 21.213, 21.213)
DRAW ( 21.213, 21.213)-( 28.284, 28.284)
DRAW ( 28.284, 28.284)-( 35.355, 21.213)
DRAW ( 35.355, 21.213)-( 28.284, 14.142)
DRAW ( 28.284, 14.142)-( 21.213, 21.213)
```