

Assignment 3
Turtle Graphics
Due Nov 4 23:59

We are going to write a Clojure program to implement turtle graphics. See the wikipedia article on turtle graphics (http://en.wikipedia.org/wiki/Turtle_graphics). We will support the turtle operations moving, turning, and raising/lowering a pen to draw, undo, step and run. We will not support changing the color of the pen, but you can if you want. These operations are described below.

The turtle language consists of the following operations.

pen up

When the pen is up and the turtle moves nothing is drawn.

pen down

When the pen is down and the turtle moves a black line is drawn

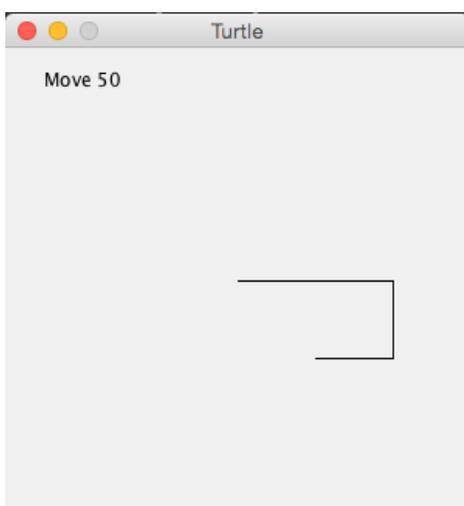
move

The move operation requires an amount to move. This operation moves the turtle given amount in the current direction.

turn

The turn operation requires an amount to turn. This operation turns the direction of the turtle the given amount in degrees. You will need to covert the degrees to radians.

When running a Turtle program the result of the turtle's pen is shown in a window. A turtle program is just a list of turtle commands.



Our Turtle program run in two modes: run and step. In the step mode the user can step the program forward one operation by pressing the forward arrow key. The user can undo the last operation by pressing the back arrow key. The user can undo back to the start of the turtle program. Each time the program executes an operation it prints the operation in the display window. A turtle program starts in the step mode. Pressing the “r” key changes to the run mode. In the run mode the turtle program is executed without the user having to press the arrow keys. When the turtle program comes to the end we switch back to step mode so the user can step the program backwards.

Graphics

We will use the Clojure library Quil to draw in a window. See <https://github.com/quil/quil> and <http://quil.info/>. Quil does not handle menus, button and other standard GUI element. It just handles drawing, keyboard and mouse events. Quil has several middleware functions, but do not use them. In particular do not use fun-mode. The middleware functions maintain state for you. Part of the assignment is to deal with maintaining state in Clojure.

What you need to do

You need to write a Clojure program that reads a turtle program from a file. Your Clojure program executes the turtle program as described above and displays the graphics in a Quil window. Don't use an absolute path when reading the file. Put the file in the directory containing your Clojure project and use relative paths to read the file. The sample program that you turn in should contain at least three of each commands. Each turn command must be followed by at least one move command. A move command must occur between every other move.

There are several issues to address. First is how to represent the turtle program. (Hint maps are your friend.). Second is how to maintain state and keep track of where you are in the turtle program. There are details on how to use Quil, but we will take about that in class.

Grading

Item	Points
Basic commands (move, turn, Pen up, Pen down)	16
run mode	14
step mode	20
undo	20
Maintaining state in your clojure code	20
Graphics in Quil	25
Code Quality	10
Display the move just executed	10
Other	10

What to turn in

You need to create a leiningen project for this assignment. The command “lein new quil assignment3” will create a project that includes Quil and a sample quil program. You could also create a regular project and add Quil to it. In your README.md file indicate the function to

execute to start your programming running and include a sample turtle program. If there are known problems with your code please indicate them in the README file. Make sure that your turtle program is included in this directory. Zip up this project and upload it to the course portal.