CS151.01 2009F *Functional Problem Solving*

Class 17: Turtle Graphics

Held: Friday, 25 September 2009

Summary: We explore another imperative model of images, turtle graphics. In this model, we give drawing expressions to robotic turtles.

Related Pages:

- EBoard.
- Lab: Turtle Graphics.
- Reading: Turtle Graphics.

Notes:

- In case you missed the announcement, Assignment 4 is now ready. I'll take questions now and early next week.
- Please have groups chosen for this assignment by Monday.
- Surprise! There is no lab writeup for this week.
- Because we'll be going over the exam on Monday, there are no readings for Monday.
- We're using the normal Friday strategy: Take the quiz and then start the lab. I won't lecture. Please leave your quiz at your desk, and I'll pick it up.

Overview:

- Modeling images through process: Turtle graphics.
- Some historical notes.
- Turtle graphics in DrFu.

Modeling the Drawing Process: Turtle Graphics

- Our explorations of GIMP-style graphics have emphasized two main operations: selecting and then doing something with the selection.
- This model permits us to create a variety of interesting drawings.
- However, it does not model how we normally draw, which involves taking pen (or brush) to paper (or canvas).
- The turtle graphics approach to describing images provides a simple model for how we might describe drawings.
- At any point, the person following the instructions has a pen in hand. You need to give the person information on the direction in which to move the pen and the amount to move it. (That's right, no curves here; just lots and lots of straight lines.)
- We separate the two basic operations: You can tell the person drawing to move forward or to turn in a particular direction.
- It's so simple, even a turtle can do it.

- What if you don't want continuous lines? You can tell the turtle to lift or drop the pen.
- Turtle graphics has been used to control robots that draw.
- Note that turtle graphics, much like GIMP graphics, is an imperative model: You give a series of commands to the thing doing the drawing.

Some Historical Notes

Disclaimer: Although I knew much of this information, I did crib some ideas from Wikipedia and the Web or these notes.

- Turtle graphics were invented by Seymour Papert (at MIT) in part of his development of the LOGO programming language. (1960's and beyond)
- LOGO was designed as a computer language intended to help children think better (or at least more algorithmically).
- The original implementation of LOGO did, in fact, have a kind of robot (commonly referred to as a turtle robot) hooked up to a computer. Hence, it made sense for the language to have some basic operations for the robot.
- As computers became more commonplace, it made sense to simulate the turtle on the screen (since not everyone who had a computer would have a turtle robot).
- And it makes sense to show the turtle's path.
- After awhile, drawing on the screen became as interesting as (or more interesting than?) controlling the physical robot.
- The turtle graphics model has persisted, in various forms, over the years.
- Turtle graphics and LOGO are often used in constructionist approaches to teaching. The goal is that students explore freely, starting with a few basic tools and strategies, they come up with their own problems and develop solutions to those problems.

Turtle Graphics in DrFu

- Create a new turtle that draws on a particular image with `(turtle-new image)`.
- Move it forward with `(turtle-forward! turtle amt)`
- Turn it with `(turtle-turn! turtle angle)`
- Lift the pen with `(turtle-up! turtle)`
- Put the pen on paper with `(turtle-down! turtle)`
- Additional operations for people who can't keep track of position and orientation
  - `(turtle-teleport! turtle col row)`
  - `(turtle-face! turtle angle)`
