Class 19: Turtle Graphics

Held: Wednesday, February 24, 2010

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.
- Due: Assignment 4: Exploring Colors.

Notes:

- Reading for Friday: Iteration.
- It’s Pop Tarts day in CSC 151.
- EC for today’s panel on Open Information Culture (4:15 pm Burling).
- EC for Wednesday’s gates lecture or Thursday’s convo.
- EC for Thursday’s CS Extra: Tony Pan on his internship at Microsoft (4:15 3821).
- EC for Friday’s CS Table: No Silver Bullet (noon, JRC PDR).
- Assignment 4 is due. Assignment 5 will be distributed on Friday.

Overview:

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

Modeling the Drawing Process: Turtle Graphics

- We’ve now seen two (more?) models of images:
  - We can use GIMP-style graphics to select and then do something with the selection.
  - We can use drawings-as-values-style graphics to build composite images.
- These models permit us to create a variety of interesting drawings.
- However, they do 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 MediaScheme

- 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)*
  - *(turtle-show! turtle)*
- You can also set the turtle’s brush and color.