

Turtle Tango

TT

Turtle Tango (TT) is a line-drawing language that uses audio input to modulate graphical output.

1) Introduction

For a generation of programmers, Logo and turtle graphics provided a simple yet seductive introduction to writing code. By situating ourselves in the position of our friendly triangular Turtle, we envisioned the instructions required in order to induce the "motion" that created on-screen drawings. Given marching instructions and a pen pressed down, Turtle could draw for us whatever our sprouting CS-minds imagined.

TT nostalgically revisits a subset of turtle graphics in order to provide a learning vehicle for language and compiler design and implementation. To add novelty, deepen the project challenge, and inspire inventive graphical representations of music, TT augments Turtle's traditional trot by affording him the capacity to modify his strict response to marching command; the programmer can allow him some frivolity on the way to his destination by modifying his step in accordance to audio input. Turtle can tango to his favorite tunes.

2) Applications

TT solves the problem that a novice or casual programmer may encounter when desiring to unleash passion for his favorite music by creatively interacting with that sound in order to produce alluring visualizations. While C and C++ libraries allow experienced developers to create plug-ins to media players such as Winamp, the skills required to complete such a project remain time-intensive to acquire and develop. Furthermore, these avenues for visualization frequently orient toward temporal and fleeting animations of sound during playback. With its simple roots as a learning language, TT provides a quick and straightforward opportunity for the inspired to experiment, envision, and vivify their favorite music into lasting graphical designs.

3) Environment

TT's state-machine environment consists of four globally defined objects.

- Disco
 - The Disco is the plane representing Turtle's stomping ground. Its size is configurable but static.
- Turtle
 - Turtle cannot move beyond the bounds of his Disco. He is described by a position, an orientation, and a rate of motion. His Position is specified in (x,y) coordinates, his Orientation in positive degrees, and his Rate in pixels/second.

- Pen
 - When the Pen is Down, Turtle draws; when it is Up, he relocates without otherwise impacting the drawing. The Pen has additional configurable properties such as Color, Style, and Width.
- Tune
 - A Tune represents the .wav audio track currently playing in the Disco. A Tune has varying Amplitude across time, and a configurable Volume, which determines the bounds of Turtle's exuberance along his dance.

4) Commands

In addition to a subset of Logo's intuitive commands that enable Turtle to draw, TT employs several simple commands to alter the environment.

- Disco Commands
 - `setDiscoSize(x,y)`
 - `setDiscoColor(color)`
- Turtle Commands
 - `fd(p)`, `bk(p)`. Turtle moves forward or back positive *p* pixels while drawing if Pen is down.
 - `lt(d)`, `rt(d)`. Turtle turns *d* degrees relative to his current Orientation.
 - `setPosition(x,y)`. Turtle jumps to the position (*x,y*) without drawing.
 - `setOrientation(d)`. Turtle turns to the Orientation *d*.
 - `setRate(r)`. Sets Turtle's rate of movement.
- Pen Commands
 - `pu()`, `pd()`. Sets the Pen Up, Down.
 - `setColor(c)`. Sets the Pen's Color to *c*.
 - `setLineStyle(s)`. Sets the Pen's LineStyle to *s*.
 - `setLineWidth(w)`. Sets the Pen's Width to *w*.
- Tune Commands
 - `cue(filePath)`. Prepares the .wav file at *filePath* for playback.
 - `pause()`. Pauses the Tune in the Disco.
 - `play()`. Resumes play of the Tune in the Disco.
 - `stop()`. Closes and releases the .wav file.
 - `setVolume(v)`. Sets the volume of the Tune to *v*.

5) Language Constructions

TT allows the following, predominantly common, modern language constructions.

- Types
 - TT supports integer types; any other numeric values encountered are rounded. A small collection of colors and styles are represented by integer values. TT also recognizes a `FilePath`, or string representing the location on disc of a .wav file.

- Variables
 - TT supports user-defined variables of type integer (int) and assignment using '='.
- Routines
 - TT supports user-defined routines accepting 0 or more arguments. Every routine returns an integer.
- Control
 - Following the Logo tradition, TT supports: "repeat(n) {...}" which indicates that the block contained between the braces will be executed n times. TT also handles the more conventional structures "if expression {...} else {...}" and "while expression {...}" that behave according to established norms.
- Standards
 - TT supports comments with "/*...*/", as well as the standard arithmetic operators (+, -, *, /) and comparators (<, <=, ==, >=, >, !=).

6) TT Programming Model

A TT program consists of variable definitions and assignments, commands, and routine calls. Control initiates in the main(filePath) routine, which accepts as an argument the FilePath to audio input.

Example TT Program:

```

/* SquareDance.tt */

routine square(x, y)
{
  int entryOrientation = Orientation;
  setOrientation(0);
  setPosition(x, y);
  repeat 4
  {
    fd(50);
    rt(90);
  }
  setOrientation(entryOrientation);
}

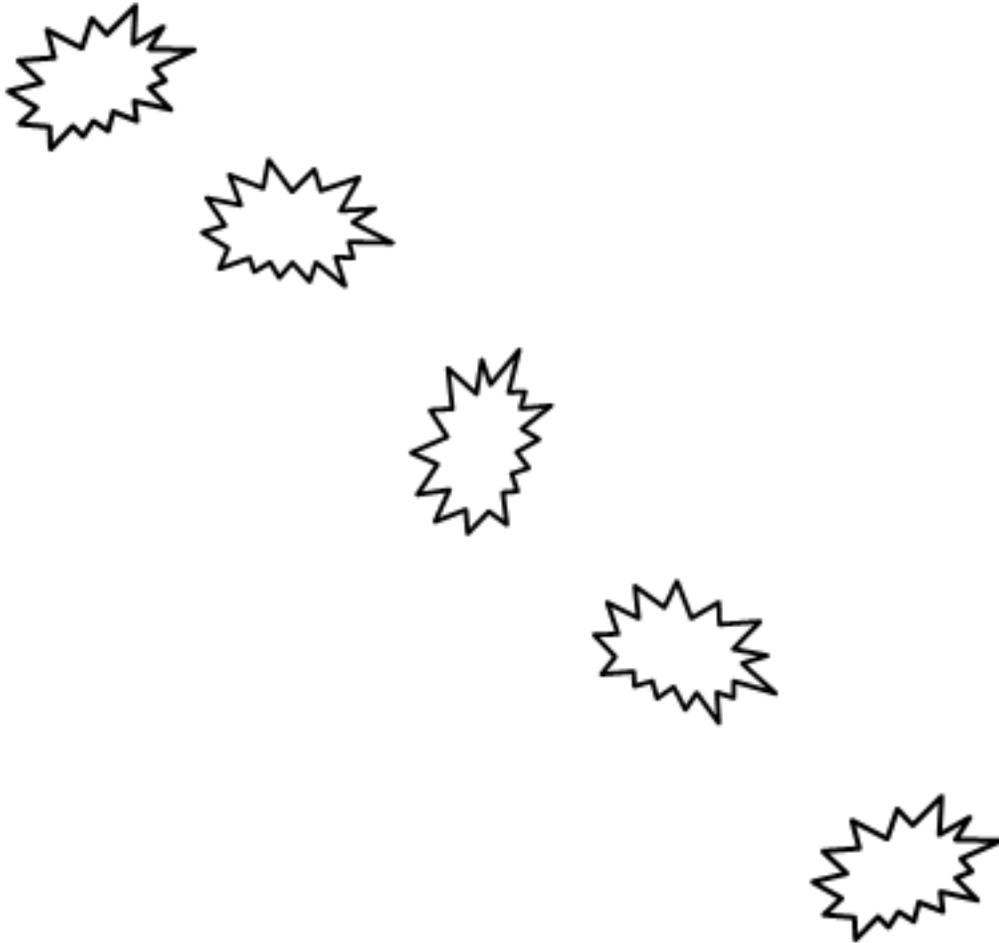
main(filePath)
{
  setDiscoSize(400, 400)
  setRate(10);
  cue(filePath);
  play();
  pd();

  int i = 1;
  while (i <= 5)
  {
    square(i*80, i*80);
    i = i + 1;
  }
}

```

```
}  
  
  stop();  
}
```

This example draws five squares positioned along the diagonal from top left to bottom right. Each square is modulated by the amplitude of the .wav file input specified by filePath. The image below represents a loose depiction of a possible output, driven by a specific audio input:



7) Extension Ideas

With sufficient time and motivation, one could extend TT to accept user input. For example, mouse-, keyboard-, or touch-driven interaction could alter the direction and speed of Turtle in real-time during audible playback of the selected Tune. Alternatively, one could construct a Lindenmayer system that allows Turtle to groove while creating fractal graphics. Ultimately, allowing multiple turtle instances to draw simultaneously would enable Turtle to greet his tango partner, Turtlet.