Turtle Graphics Quick Reference

Guido Gonzato, PhD

April 16, 2019

1 Turtle Graphics

The SDL_bgi sources archive includes a `test/` directory that contains test programs and also a turtle graphics implementation (files `turtle.h` and `turtle.c`).

Turtle graphics is based on polar coordinates: lines are drawn specifying the distance in pixels and an angle in degrees, called *heading* (0-359). Heading is the same as the bearing of a compass: it starts from 0 ("North") and increases clockwise. For more information: https://en.wikipedia.org/wiki/Turtle_graphics

Some functions have aliases with the same name as their Python counterparts; https://docs.python.org/2/library/turtle.html

---

void back (int *px*)

Moves the turtle backwards by *px* pixels.
Aliases: `backward()`, `bk()`

---

void forwd (int *px*)

Moves the turtle backwards by *px* pixels.
Aliases: `forward()`, `fd()`

---

void turnleft (int *deg*)

Turns the turtle left by *deg* degrees.
Aliases: `left()`, `lt()`

---

void turnright (int *deg*)

Turns the turtle right by *deg* degrees.
Aliases: `right()`, `rt()`

`void setposition (int x int y)`

Moves the turtle to new coordinates x, y.
Aliases: `go_to()`, `setpos()`

`void setx (int x)`

Sets the turtle's x coordinate.

`void sety (int y)`

Sets the turtle's y coordinate.

`void setheading (int deg)`

Sets the turtle heading to deg degrees.
Alias: `seth()`

`void home (void)`

Moves the turtle to the screen centre and sets the heading to 0.

`int * position (void)`

Returns a pointer to a 2-element integer array containing the turtle coordinates.

`int xcor (void)`

Returns the turtle's x coordinate.

`int ycor (void)`

Returns the turtle's y coordinate.

`int heading (void)`

Returns the turtle's heading in degrees.

`void savestate (void)`

Saves the turtle state, i.e. its coordinates and heading.
TODO: make this a stack?

`void restorestate (void)`

Restore previously saved state. If no state has been saved, this function is equivalent to `home()`.

`void pendown (void)`

Activates drawing (puts the pen down).
Aliases: `pd()`, `down()`

`void penup (void)`

Suspends drawing (lifts the pen down).
Aliases: `pu()`, `up()`

`int isdown (void)`

Returns 1 if the pen is down.

`void hideturtle (void)`

Hides the turtle.

`void showturtle (void)`

Draws the turtle.

`int isvisible (void)`

Returns 1 if the turtle is visible.

`int turtleshape (int` *shape*`)`

Sets the turtle shape; available shapes are `T_CIRCLE` or `T_TRIANGLE`.

`int turtlesize (int` *px*`)`

Sets the turtle size (radius or side) to *px* pixels.

`void wrap (void)`

Wraps around the screen.

`void wrap (void)`

Does not wrap around the screen.