Turtle Graphics Mode

Go Play Space supports the Turtle graphics mode to help visualize algorithms and make learning experience more fun.

Whenever Go code is executed and produces some console output, this output is parsed, and found control commands are interpreted on a drawing board. You can start with something basic like one `fmt.println` statement that prints all the commands sequentially, and then build your own functional API and algorithms — in Go — that would produce the desired sequence of control commands.

Try These Examples

1. Star — an example on how to draw a star
2. House — an example on drawing a house with multiple colors
3. Tree — an example on using recursion to draw a tree
4. Colored squares — an example of using a simple “API” wrapper functions and *for* loops
5. Colored squares (Russian) — the same “Colored squares” example above, but with function/variable names in Russian. This example demonstrates how you can offer your kids a drawing API in their spoken language of choice.

Control Commands Reference

1. `draw mode` — triggers the draw mode; any commands before this line are ignored; gopher starts at the center of the board
2. `left` — equivalent to `left 90`: turn 90 degrees counter-clockwise
3. `left *N*` — N degrees counter-clockwise [0..360]; fractional values allowed
4. `right` — equivalent to `right 90`: turn 90 degrees clockwise
5. `right *N*` — N degrees clockwise [0..360]; fractional values allowed
6. `forward` — make one grid step
7. `forward *N*` — make grid steps; fractional values allowed
8. `color off` — turn color off (leave no trace; this is the default state)
9. `color *Color*` — set stroke color; any web color would work, e.g. “red”, “black”, “#f5c0e2”, “rgba(0,0,0,0.3)”
10. `width *N*` — set brush width to pixels; fractional values allowed; default is 2
11. `say *Message*` — shows a temporary ’balloon’ message

How Commands Are Interpreted

Once your Go code produces the full console output, this output is split into lines, each line is trimmed and then checked against the available commands. Any output before `draw mode` line is ignored. There must be only one command per line. Lines that don't match any of the commands above are ignored, so you can safely print debug message lines along with the actual commands.

Feedback

Feel free to provide your feedback, suggestions or bug reports in the official bug tracker, or message @afan in the Gophers Slack channel.