

KNOX

Logo

Knox Game Design

August 2022

Levi D. Smith

DESIGN

Overview

- Created in 1967
- Language family: LISP
- Turtle graphics

KNOX

GAME

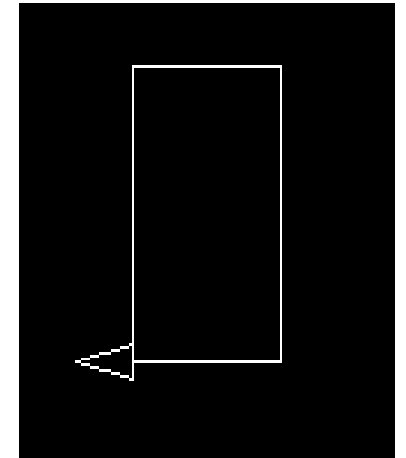
DESIGN

KNOX
GAME
DESIGN

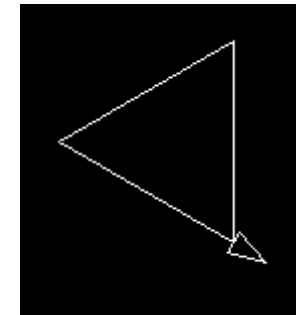
Moving the turtle

- Moving
 - forward N / fd N
 - back N / bk N
- Turning (degrees)
 - right N / rt N
 - left N / lt N
- Draw on/off
 - penup
 - pendown
- Set direction (degrees)
 - seth N

```
? forward 100
? right 90
? forward 50
? right 90
? forward 100
? right 90
? forward 50
```



```
? forward 100
? left 120
? forward 100
? left 120
? forward 100
```



Screen coordinates

- Default
 - center is 0, 0
 - positive X to left to right
 - Positive Y bottom to top

KNOX
GAME
DESIGN

KNOX
GAME
DESIGN

Cleaning up

- clearscreen
- hideturtle

KNOX

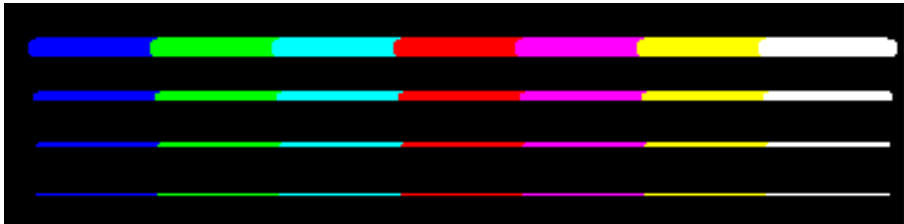
GAME

DESIGN

KNOX
GAME
DESIGN

Change drawing attributes

- setpencolor *N*
- setpencolor [r g b]
- setpensize *N*



```
? setpencolor 1
? setpensize 1
? repeat 7 [ forward 50 setpencolor pencolor + 1]
```

```
? setpencolor [100 0 50] If using r g b values, range is 0 to 100
```

RGB	int	color
000	0	black
001	1	blue
010	2	green
011	3	cyan
100	4	red
101	5	magenta
110	6	yellow
111	7	white
	8	brown
	9	tan
	10	forest
	11	aqua
	12	salmon
	13	purple
	14	orange
	15	grey

KNOX
GAME
DESIGN

Variables

- make "*var_name value*"

```
? make "secretnumber 42  
? print :secretnumber  
42
```

```
? make "secretnumber random 100 + 1  
? print :secretnumber  
58
```

```
? make "n 123  
? print :n  
123  
? make "m rw  
456  
? print :m  
456
```

Input

- readword / rw

```
? print readword  
hello  
hello
```

```
? make "numguess readword  
50  
? print :numguess  
50
```


Control / Conditions

- if *<condition>* [*<true_action>*]
- ifelse *<condition>* [*<true_action>*] [*<false_action>*]
- while [*<condition>*] [*<true_action>*]

```
? while [:i < 10][print :i make "i :i + 1]
1
2
3
4
5
6
7
8
9
```

```
? if :numguess = 50 [ print "true ]
true
? if :numguess = 25 [ print "true ]
? ifelse :numguess = 50 [ print "true ] [ print "false ]
true
? ifelse :numguess = 42 [ print "true ] [ print "false ]
false
```

KNOX
GAME
DESIGN

Random numbers

- random N
 - Zero to N (exclusive)

```
? print (random 6) + 1
3
? print (random 6) + 1
4
? print (random 6) + 1
6
? print (random 6) + 1
2
```

dice roll (1 to 6)

```
? print random 100
22
? print random 100
75
? print random 100
24
```

random int 0 to 99

KNOX
GAME
DESIGN

Number guessing game

```
? make "num_guess -1
? make "num_secret (random 100) + 1
? make "num_guess_count 0
? while [:num_guess <> :num_secret][
~  make "num_guess readword
~
~  print :num_guess
~
~  make "num_guess_count :num_guess_count + 1
~
~  if [:num_guess > :num_secret] [ print "Lower]
~
~  if [:num_guess < :num_secret] [ print "Higher]
~
~  if [:num_guess = :num_secret] [
~
~      print (sentence "Correct :num_guess_count "guesses)
~
~  ]
~ ]
```

```
50
50
Lower
25
25
Higher
37
37
Higher
43
43
Higher
46
46
Correct 5 guesses
? |
```

BOX

ME

SIGN

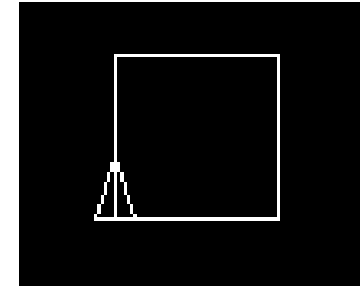
Defining procedures

- to *proc_name*

<definition>

end

```
? to square
> repeat 4 [forward 50 right 90]
> end
square defined
? square
```



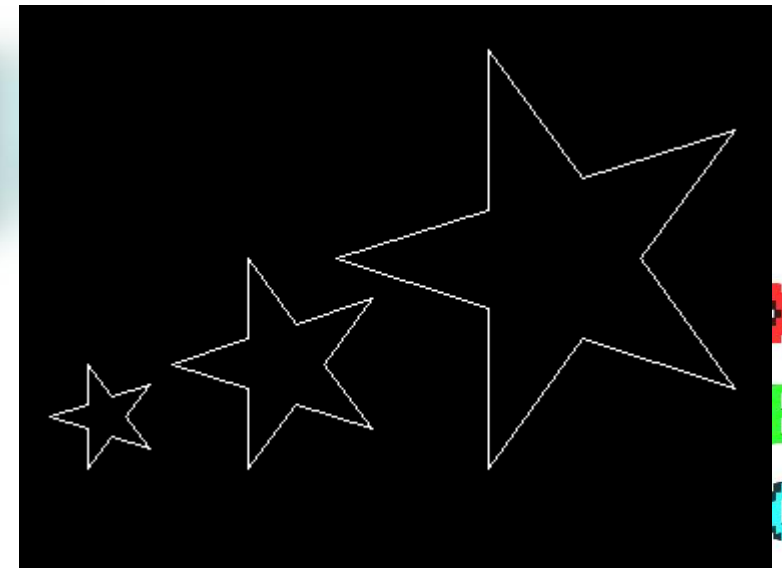
- to *proc_name :param1 :param2*

<definition>

end

```
? to star :size
> repeat 5 [fd :size lt 72 fd :size rt 144]
> end
star defined
```

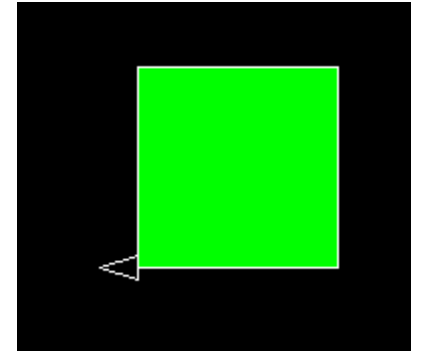
```
? clearscreen setx 0 pendown
? star 20 penup setx 80 pendown
? star 40 penup setx 200 pendown
? star 80 penup ht
```



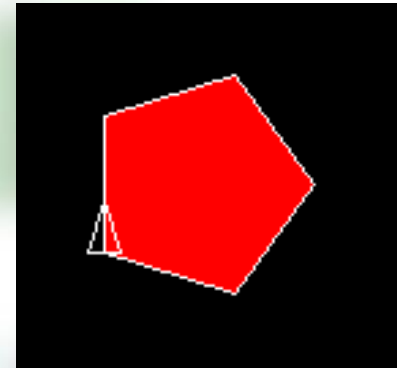
X
E
GN

Filling Areas

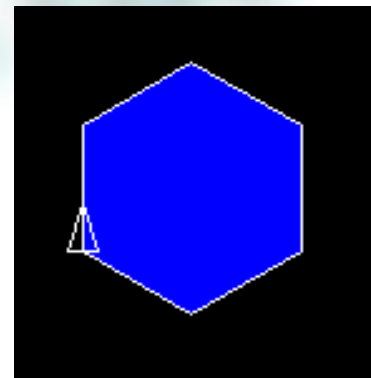
```
? filled 2 [ fd 100 rt 90 fd 100 rt 90 fd 100 rt 90 fd 100 ]  
2 |
```



```
? filled 4 [ repeat 5 [fd 50 rt 72] ]  
2 |
```



```
? filled 1 [ repeat 6 [fd 50 rt 60] ]  
2 |
```



KNOX
GAME
DESIGN

Example

```
clearscreen
setpencolor 4
filled 4 [ fd 144 rt 90 fd 222 rt 90 fd 144 rt 90 fd 222 ]
rt 180 fd 222
setpencolor 7
filled 7 [ fd 3 lt 90 fd 144 lt 90 fd 3 lt 90 fd 144 ]
lt 90 fd 3
setpencolor 1
filled 1 [ fd 15 lt 90 fd 144 lt 90 fd 15 lt 90 fd 144 ]
penup
setx 111 + 39
sety 72
pendown
setpencolor 7
filled 7 [ repeat 90 [ fd (39 * 3.14 * 2 / 90) rt (360 / 90) ] ]
setx 111 + 36
setpencolor 1
filled 1 [ repeat 90 [ fd (36 * 3.14 * 2 / 90) rt (360 / 90) ] ]

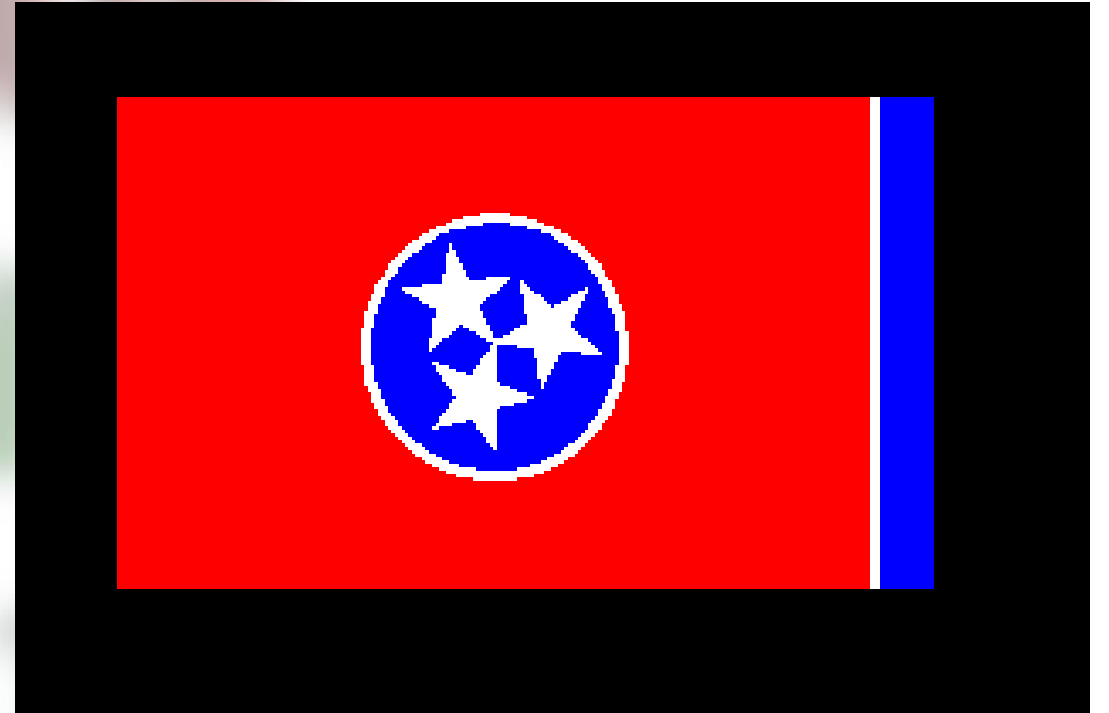
setx 111 sety 72
setpencolor 7

seth 60
filled 7 [ repeat 5 [ fd 12 lt 72 fd 12 rt 144 ] ]

seth 180
filled 7 [ repeat 5 [ fd 12 lt 72 fd 12 rt 144 ] ]

seth 300
filled 7 [ repeat 5 [ fd 12 lt 72 fd 12 rt 144 ] ]

ht
```



KNOX
GAME
DESIGN

Snap!

- Scratch style graphical code blocks
- Development environment runs in web browser
- Mouse / keyboard input

