



# NETLOGO TUTORIAL I

---

Lisa McManus  
NBHS STEM Club

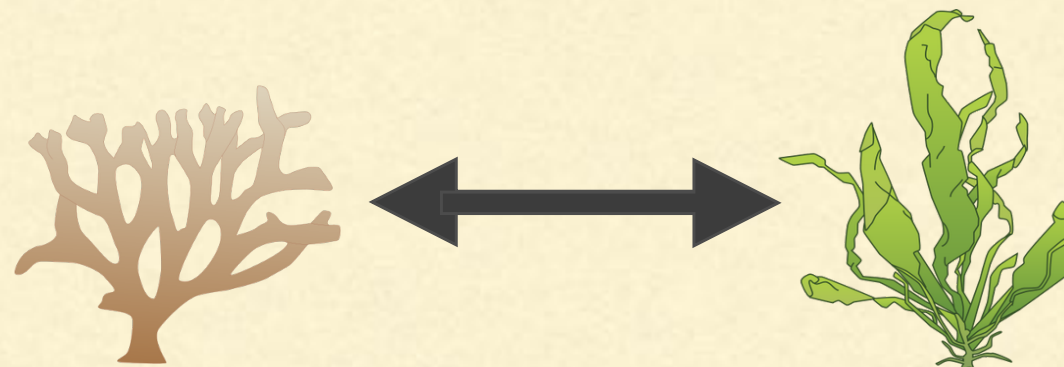
---

---

# AGENT-BASED MODELS (ABMS)

---

- Computational method where a phenomenon is modeled in terms of agents and their interactions
- ABMs represent a bottom-up modeling approach, where we provide rules for the behavior of each individual
- “Simple rules for simple entities”

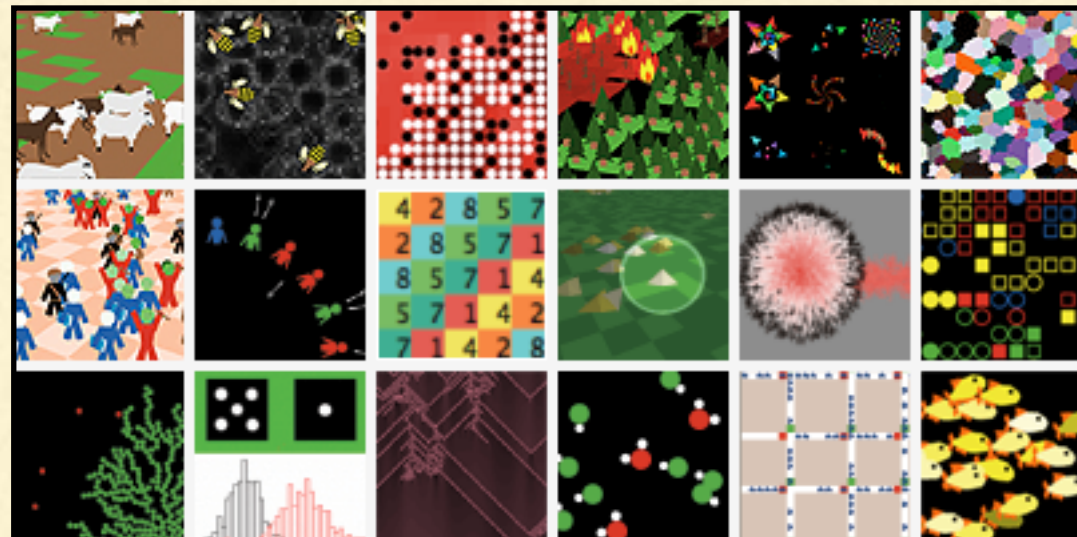


---

# NETLOGO

---

- NetLogo is an agent-based modeling environment
- Created by Uri Wilensky in 1999
- Currently being developed at the Center for Connected Learning and Computer-Based Modeling at Northwestern University





---

# AGENTS

---

- Agents are objects that follow instructions
  - NetLogo has 4 types: **turtles**, **patches**, **links** and the **observer**
  - **Turtles** move around in a 2d world
  - Each **patch** is a square piece of “ground” over which turtles can move
  - **Links** connect 2 **turtles**
  - The **observer** looks out onto the world and gives instructions
-

---

# [HTTPS://NETLOGOWEB.ORG](https://netlogoweb.org)

---

- Go to the website listed above
  - Under the ‘Search the Models Library’ Tab go to ‘Code Examples/Tutorial 3’
-

---

# COMMAND CENTER

---

Type each of the following statements, one at a time, into the command center command line (hit enter after each instruction):

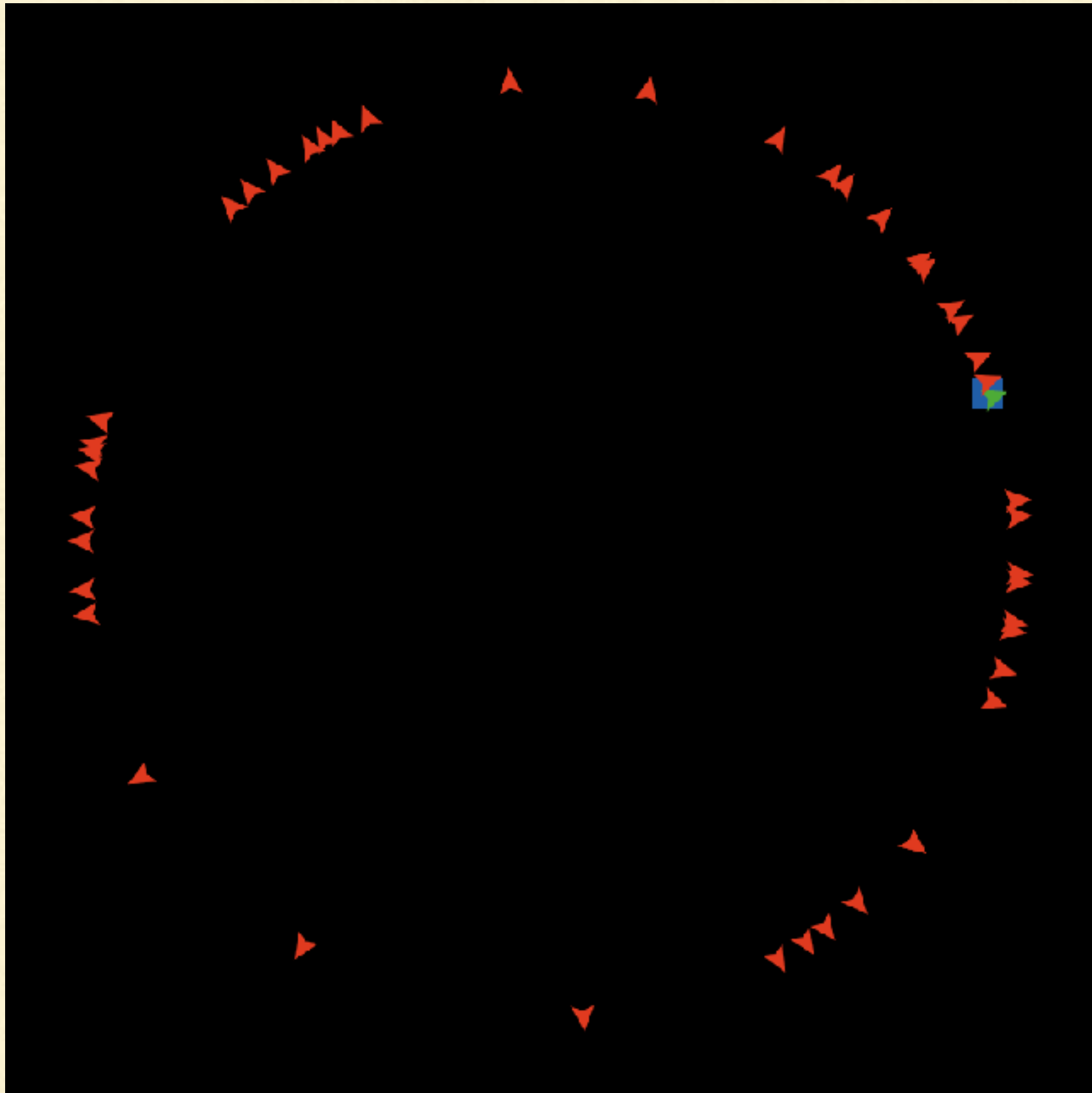
*clear-all*

*create-turtles 50 [set color red]*

*ask turtles [forward 15]*

*ask turtle 0 [set color green set pcolor blue]*

---





---

# NOTES ON TURTLES

---

- Turtles must be created and do not exist by default
  - Turtles move across the patches in the view
  - We can command the observer to **ask** every turtle to move
  - Blocks of related code are inside brackets: [forward 10]
  - A turtle is identified by its id or 'who' value
-



---

# COMMAND CENTER

---

You change who you issue commands to. This is called changing the context:

- **observer** context:
    - used to create new turtle and link agents
    - issue commands to the interface window (like clear-all)
    - can be used to instruct pass commands to patches or turtles
  - **patches** context: give instructions directly to **patches**
  - **turtles** context: give instructions directly to **turtles**
  - **links** context: give instructions directly to **links**
-

---

# TURTLES CONTEXT

---

- Switch to the turtle context by clicking on the front of the command line and choosing turtles.
- Enter the following commands. Don't include the part with ;; - These are comments and will be very important when we code our project

*fd 5 ;; move forward 5 units*

*rt 180 ;; turn right 180 degrees*

*pd fd 5 pu ;; pen-down, move forward 5, pen-up*

*set hidden? true ;; turn invisible*

*set pcolor white ;; set patch color to white*

---

---

# PRIMITIVES

---

- Primitives are **variables** (attributes) and **instructions** (commands) that are built into the language
  - We didn't have to create these
  - The instructions like *forward* and *clear-all*, and variables like *color* and *pcolor* are all referred to as **primitives**
-



---

# WORKING WITH TURTLES

---

<u>Name</u>	<u>Description/Example</u>
<b>create-turtles (crt)</b>	crt 5 ;; create 5 new turtles
<b>cro</b>	<i>As above but turtle headings range from 0 to 360, evenly spaced.</i>
<b>clear-turtles (ct)</b>	ct ;; deletes all turtles
<b>forward (fd) back (bk)</b>	ask turtles [fd 5] ;; go forward 5 steps
<b>setxy</b>	ask turtle 5 [ setxy 5 5 ] ;; move to 5,5
<b>right (rt) left (lt)</b>	ask turtles [ rt 5 ] ;; turn right 5 degrees

<u>Name</u>	<u>Description/Example</u>
<b>jump</b>	ask turtles [jump 5] ;; go forward 5 steps
<b>hatch</b>	ask turtle 0 [hatch 1] ;; ask turtle 0 to create 1 new turtle
<b>die</b>	ask turtle 0 [die] ;; ask turtle 0 to delete itself.
<b>pen-down pen-up</b>	ask turtle 0 [pen-down] ;; ask turtle 0 to start drawing
<b>ht st</b>	ask turtles [ht] ;; hide all turtles
<b>random-xcor random-ycor</b>	ask turtles [ setxy random-xcor 0] ;; move turtles to a random xcor

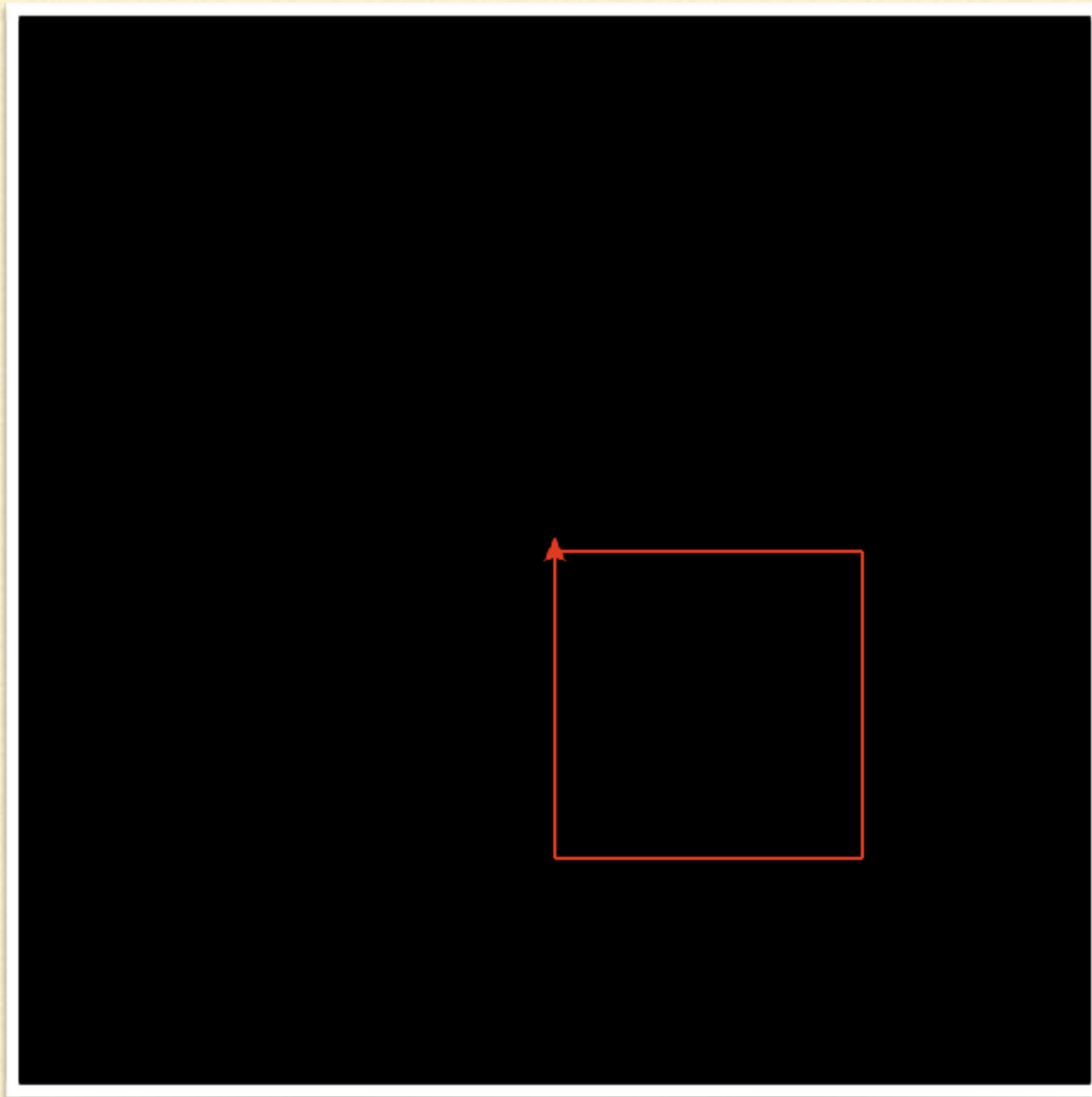
---

---

# EXERCISE

---

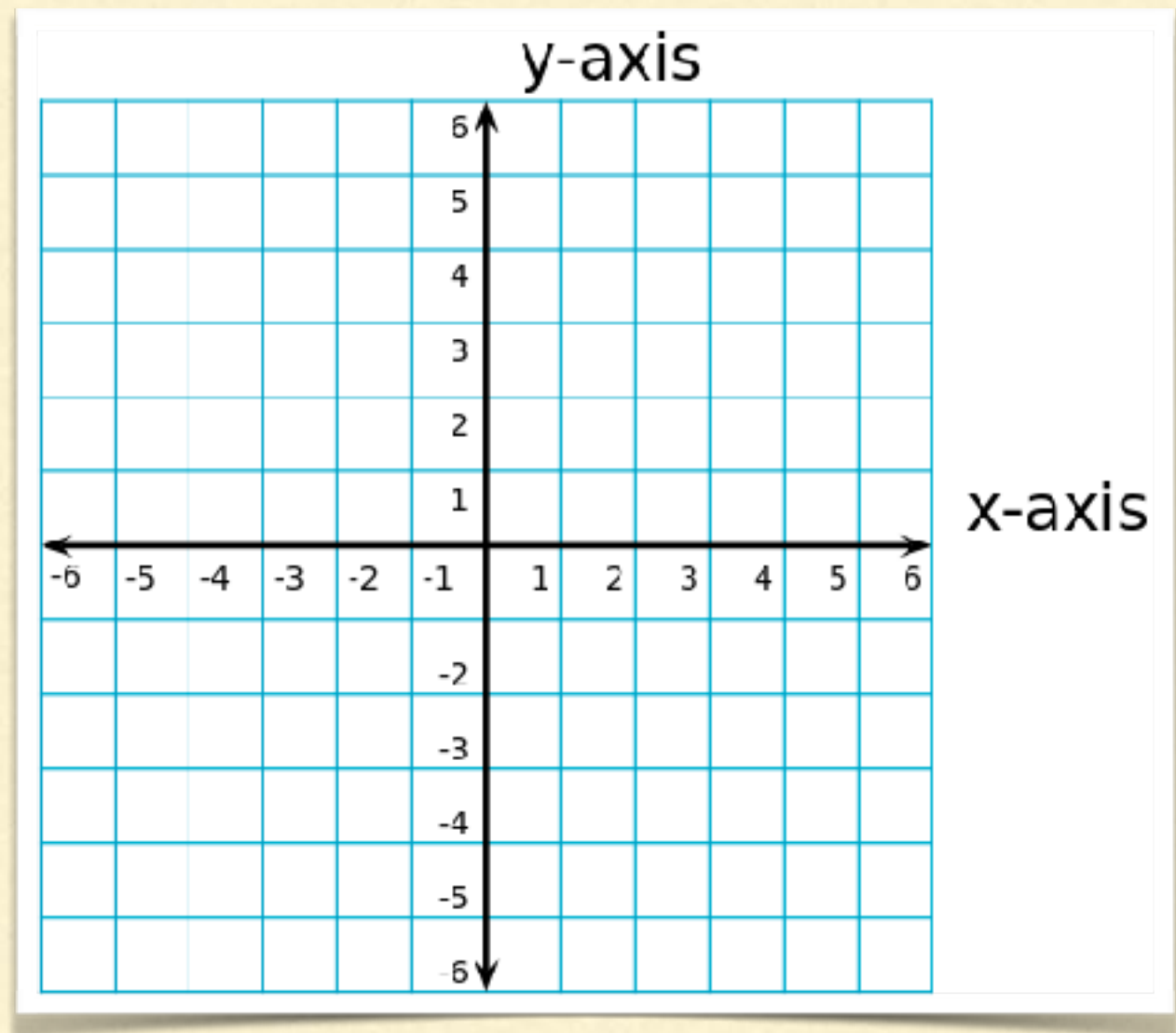
- Draw a square using a turtle
  - The following commands might be useful:
    - set heading x* ;; x is an angle between 0 and 360
    - pen-down*
    - clear-all*
    - forward* or *fd*
    - back* or *bk*
-





# PATCH CONTEXT

- NetLogo works in a 2d world with a coordinate system
- Change the command context from **turtles** to **observer**
- Enter the following commands:



---

*clear-all*

*ask patches [set pcolor yellow]*

*ask patches with [pxcor < -5] [set pcolor black]*

*ask patches with [pycor > 5] [set pcolor green]*

*ask patch 0 0 [set pcolor red]*

*watch patch 0 0*

*reset-perspective*

*ask patch 0 0 [ask neighbors[set pcolor [pcolor] of myself] ]*

---

---

*clear-patches*

*ask patches [set color random 1 40]*

*ca ;; short for clear-all*

*ask patches [set pcolor white]*

---



---

# NOTES ON PATCHES

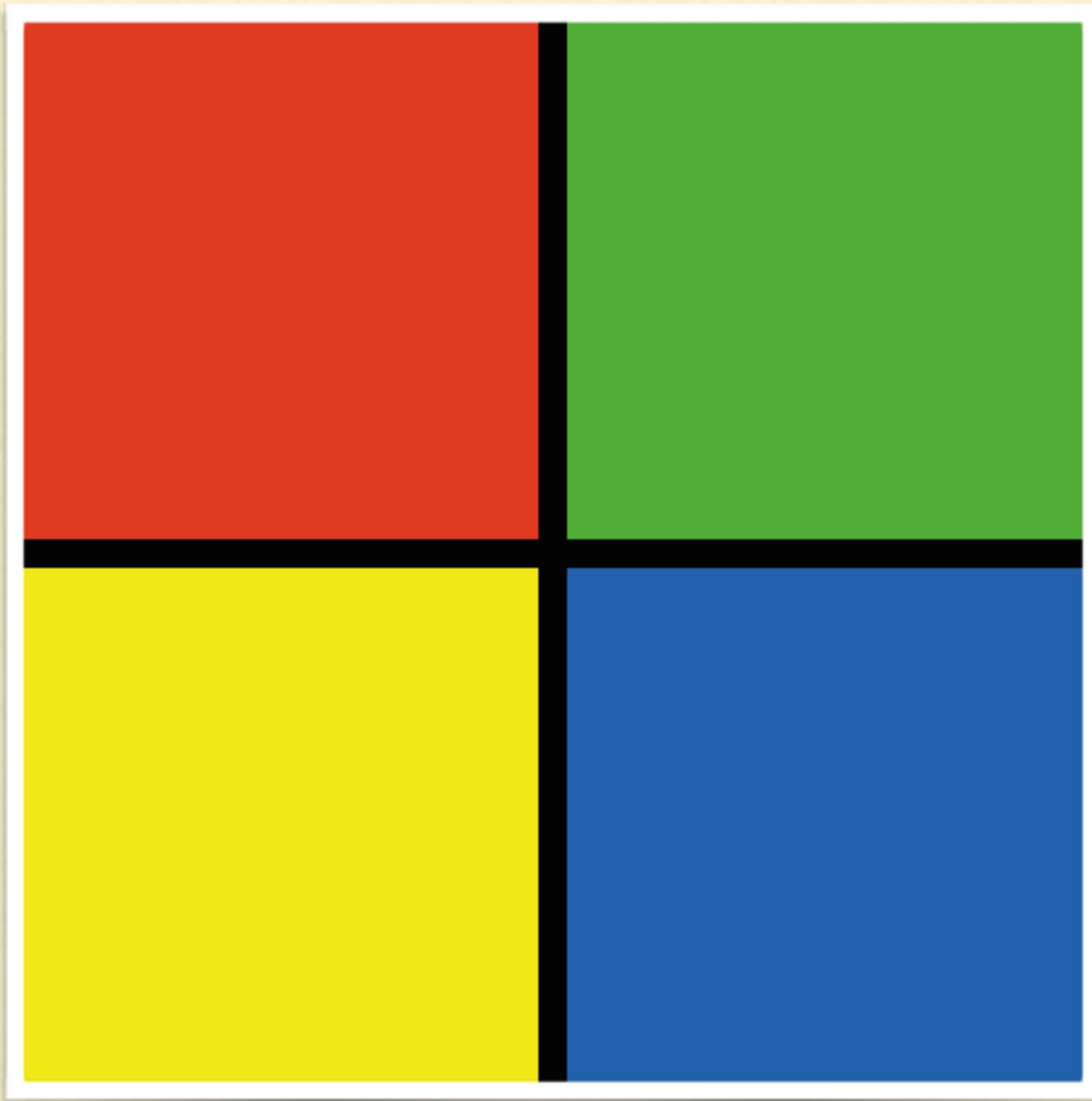
---

- Patches exist by default - no need to create them
  - Can execute commands
  - Do not move
  - Often used to represent the environment or resources
  - Can **sprout** turtles and affect other patches
  - Uniquely identified by coordinates (pxcor, pycor)
-

---

# HOW DO YOU MAKE THIS?

---



Hint: you can have more than one condition or “with” per command

---

---

# PROCEDURES

---

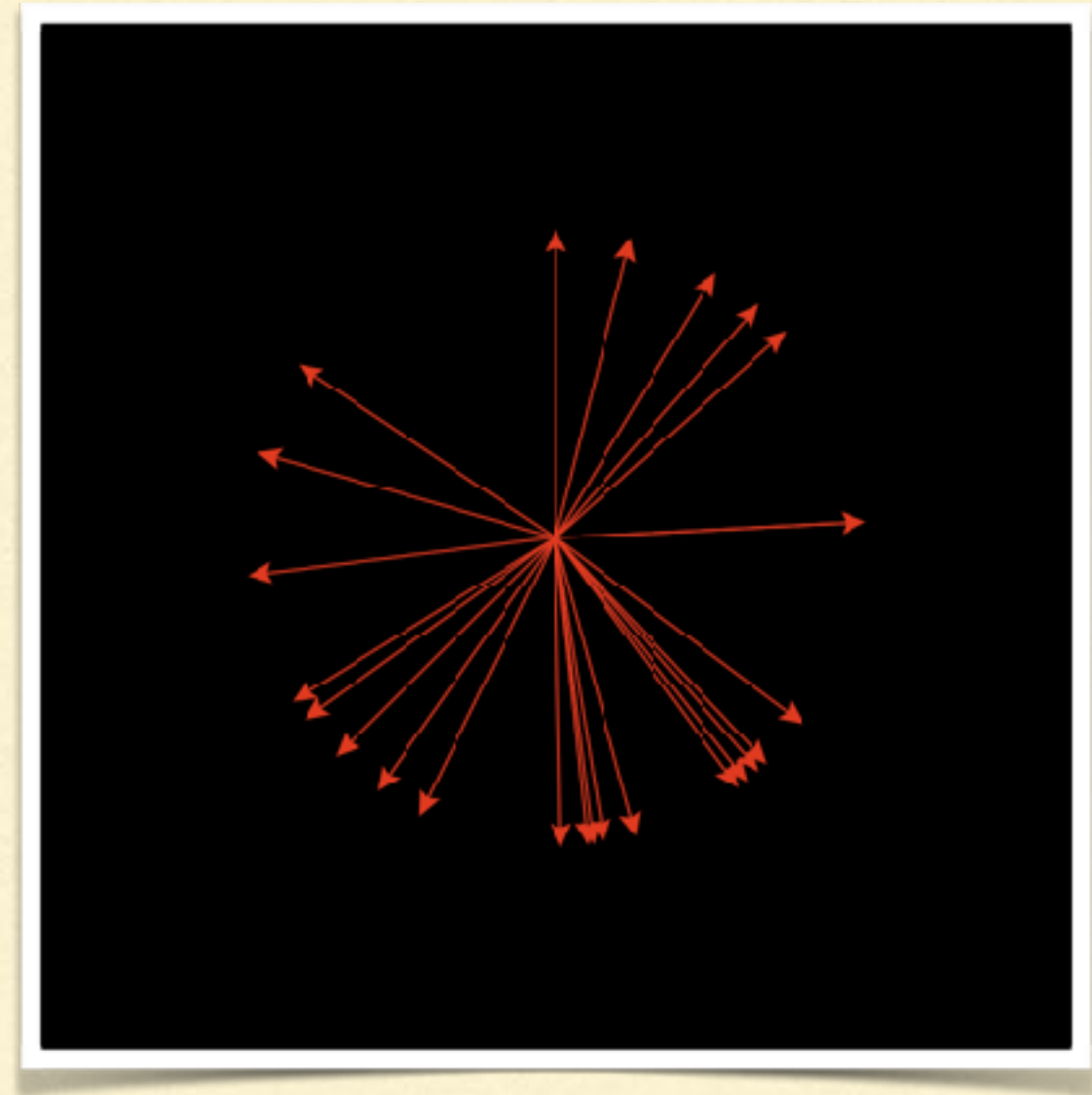
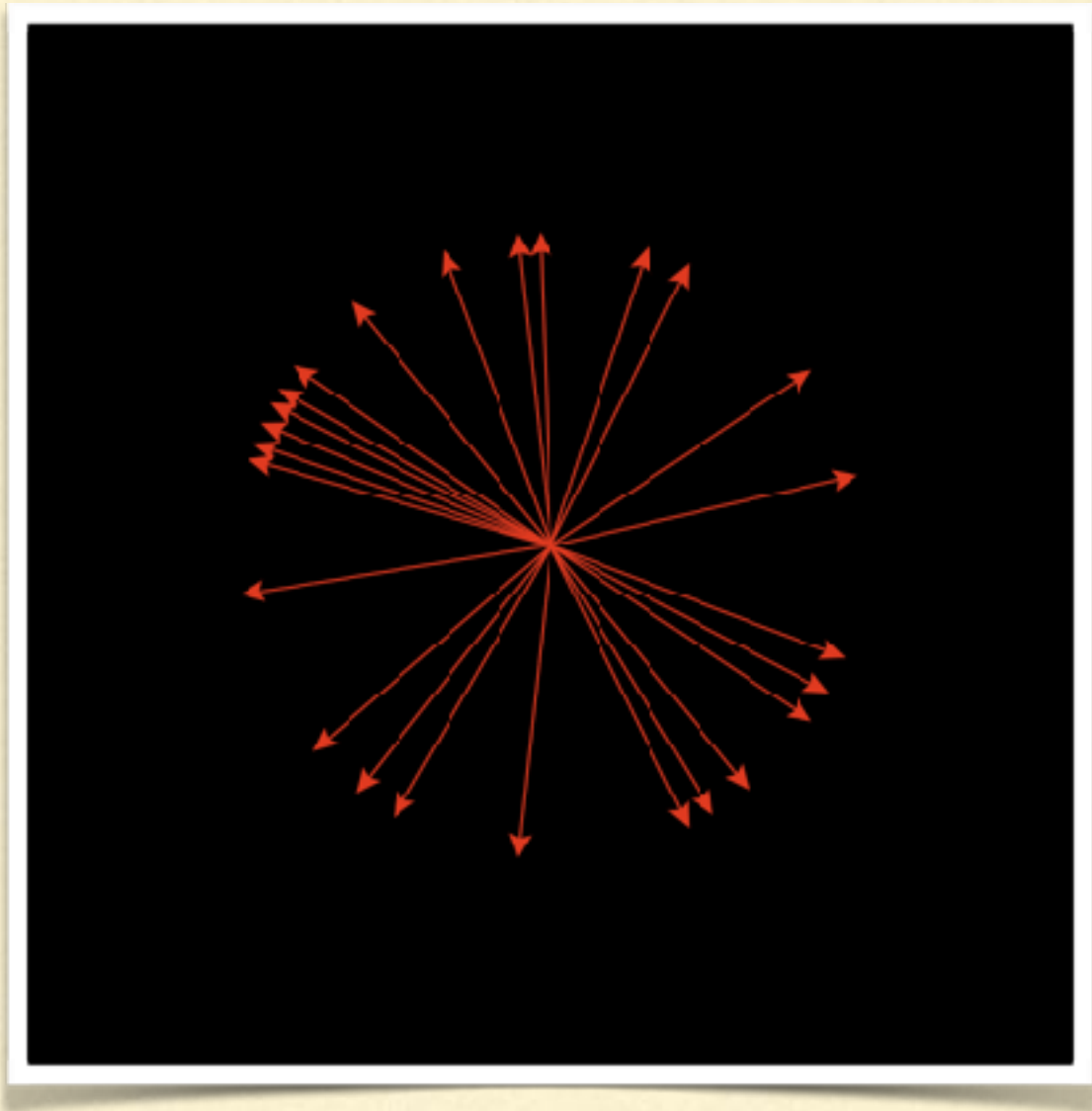
- A set of instructions is known as a procedure or a function
- Click on the 'NetLogo Code' tab: select all the text and delete (Ctrl-A and Delete)
- Type the following and then press 'Recompile Code'

```
to goBoom  
  clear-all  
  create-turtles 25 [set color red]  
  ask turtles  
    pen-down  
    forward  
  ]  
end
```

---



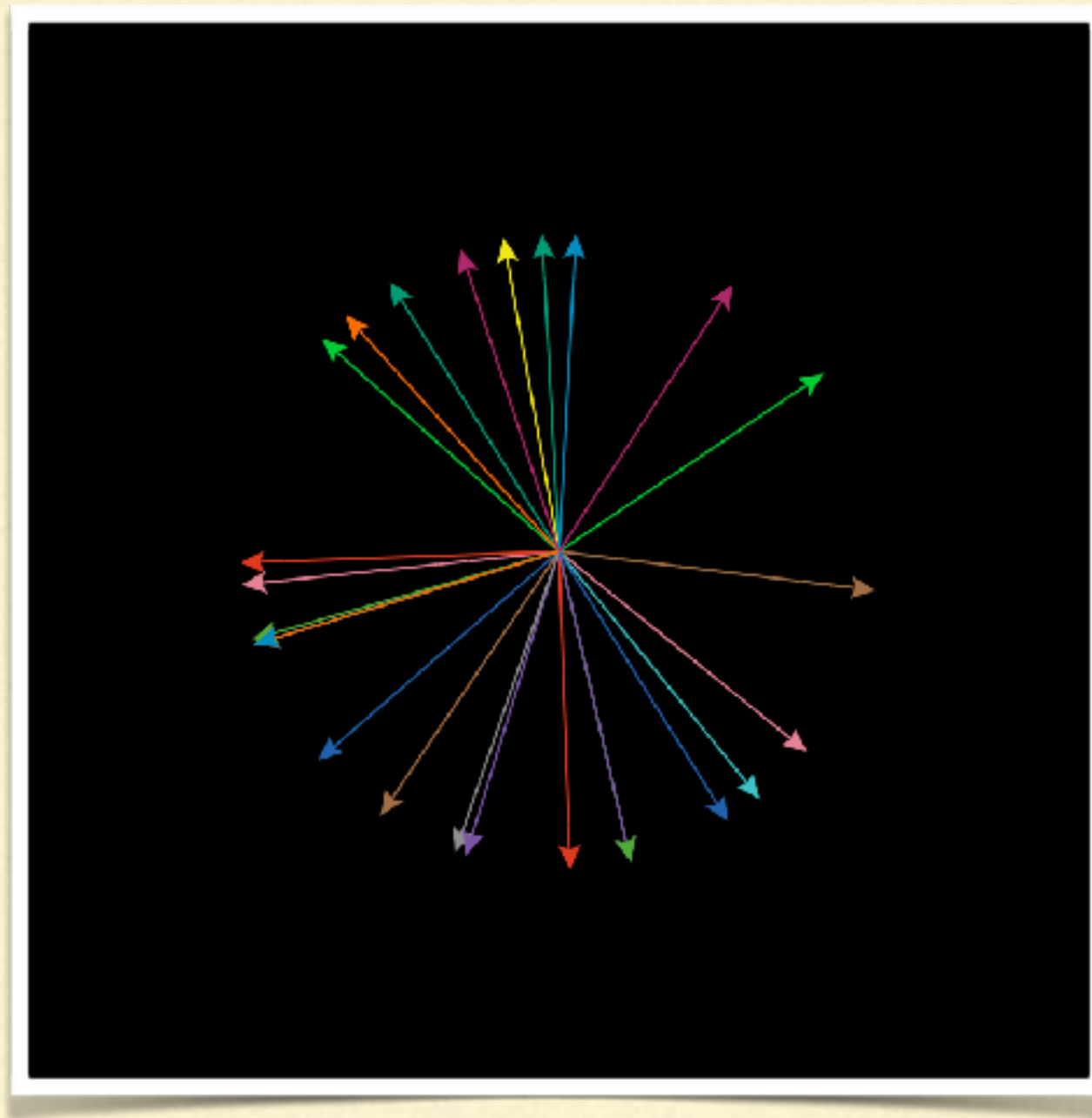
- 
- Change to the observer context in the Command Center
  - Type 'goBoom' and hit enter, try this a few times
  - Why does this procedure look slightly different each time?
-



---

# HOW DO YOU MAKE THIS?

---





---

# NETLOGO SHAPES

---

Row 1: default, airplane, arrow, box, bug, butterfly, car

Row 2: circle, circle 2, cow, cylinder, dot, face happy, face neutral

Row 3: face sad, fish, flag, flower, house, leaf, line

Row 4: line half, pentagon, person, plant, sheep, square, square 2

Row 5: star, target, tree, triangle, triangle 2, truck, turtle

Row 6: wheel, x

