

NetLogo로 따라하는 Agent-based Modeling

by. Hongjoon Park

2026.01.26.

Netlog 화면 구성

세 개의 탭

- Interface
- Info
- Code

인터페이스 탭

- 툴바
- 대시보드
- 커맨드 센터

위젯(widget)

- button
- slide
- input
- monitor
- switch
- chooser
- plot
- output
- note

월드(world)

- settings : $x\text{cor} \in [-16 \sim 16]$ 일때 x축에 퍼치는 총 17개

커맨드 센터

- 에이전트 선택(tap 키, 화살표 위아래 키) → 명령

네 종류의 에이전트(agents)

- observer
- turtles
- patches
- links

에이전트 내부 변수(특성 값)

터틀(Turtle) 관련

- `who` – 터틀의 ID
- `xcor` – 터틀의 x좌표
- `ycor` – 터틀의 y좌표
- `heading` – 터틀의 방향(각도)
- `shape` – 터틀의 모양
- `color` – 터틀의 색상
- `size` – 터틀의 크기

패치(Patch) 관련

- `pcolor` – 패치의 색상
- `pxcor` – 패치의 x좌표
- `pycor` – 패치의 y좌표

에이전트 기본 명령

observer

- `create-turtles n [...]` - n개의 터틀 생성
- `crt n [...]` - `create-turtles`의 약자
- `clear-all` - 모든 것을 초기화

turtles

- `forward n` 또는 `fd n` - n만큼 앞으로 이동
- `right angle` 또는 `rt angle` - 오른쪽으로 angle도 회전
- `left angle` 또는 `lt angle` - 왼쪽으로 angle도 회전
- `setxy x y` - 특정 좌표(x, y)로 이동
- `set color <color>` - 색상 설정
- `set heading <direction>` - 방향 설정(0-359도)
- `set size n` - 크기 설정
- `set shape "shape"` - 모양 설정
- `set label "text"` - 라벨 설정
- `pen-down` 또는 `pd` - 이동 경로 그리기 시작
- `pen-up` 또는 `pu` - 이동 경로 그리기 중단

patches

- `set pcolor <color>` - 패치 색상 설정
예) if pycor > 0 [set pcolor red]

흐름 제어 명령

if

if 조건 [명령어]

- 조건이 참일 경우에만 명령어를 실행
- 예시: `if pcolor = red [set pcolor blue]`

ifelse

ifelse 조건 [참일 때 명령어] [거짓일 때 명령어]

- 조건이 참이면 첫 번째 블록의 명령어를, 거짓이면 두 번째 블록의 명령어를 실행
- 예시: `ifelse pxcor > 0 [set pcolor red] [set pcolor blue]`

repeat

repeat 반복횟수 [명령어]

- 지정된 횟수만큼 명령어를 반복
- 예시: `repeat 5 [fd 1 rt 90]`

wait

wait 시간(초)

- 지정된 시간만큼 실행을 일시 중지
- 예시: `wait 1`

변수 선언과 할당

`let`

`let` 변수명 값

- 지역 변수를 선언하고 초기화
- 예시: `let x 10`

`set`

`set` 변수명 값

- 이미 선언된 변수의 값을 변경
- 예시: `set x x + 1`

프로시저(Procedure) 정의

프로시저는 특정 작업을 수행하는 코드 블록(= 함수)

기본 구문

`to` 프로시저이름

 ; 코드 내용

`end`

매개변수가 있는 프로시저

`to` 프로시저이름 [매개변수1 매개변수2 ...]

 ; 코드 내용

`end`

리포터(Reporter) 정의

- Reporter는 NetLogo에서 정보나 계산된 값을 반환하는 역할
- 리포터는 NetLogo에서 값을 반환하는 함수라고 이해하면 됨

기본 구문

`to-report 리포터이름`

`; 코드 내용`

`report 반환값`

`end`

매개변수가 있는 리포터

`to-report 리포터이름 [매개변수1 매개변수2 ...]`

`; 코드 내용`

`report 반환값`

`end`

내장 리포터(reporter)

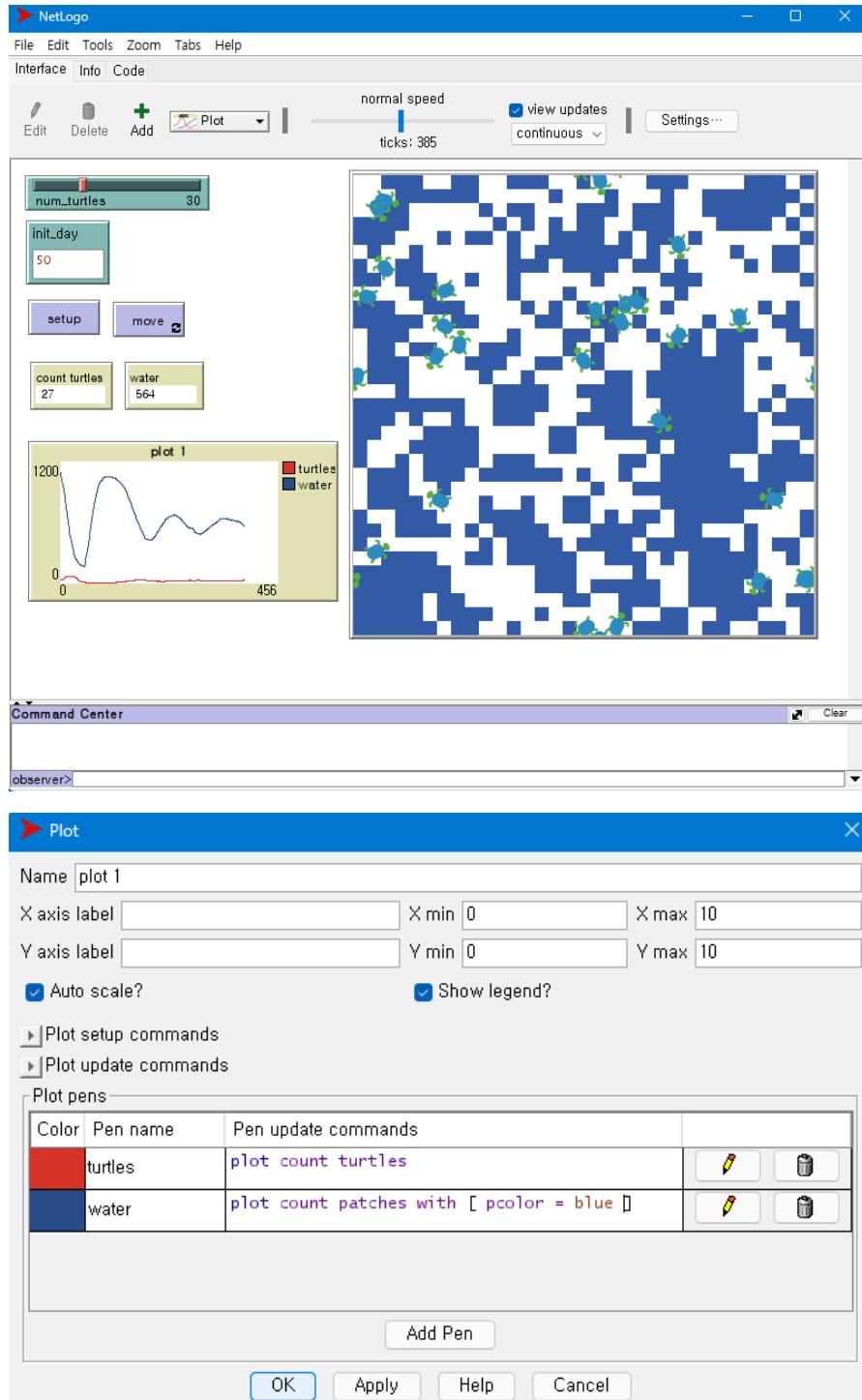
에이전트집합 관련

- `count` 에이전트집합 – 집합 내 에이전트 수를 반환
- `any?` 에이전트집합 – 집합에 에이전트가 있는지 여부를 반환(불리언)
- `all?` 에이전트집합 [조건] – 모든 에이전트가 조건을 만족하는지 여부를 반환

랜덤 관련

- `random n` – 0부터 n-1 사이의 무작위 정수를 반환

기본 예제_어항(화면)



기본 예제_어항 (코드)

```

turtles-own [energy]
patches-own [day]

to setup
  clear-all
  crt num_turtles [
    set energy 10
  ]
  ask patches [
    set pcolor blue
    set day 0
  ]
  ask turtles [
    setxy random-xcor random-ycor
    set shape "turtle"
    set size 2
  ]
  reset-ticks
end

to move
  ask turtles[
    rt random 90
    lt random 90
    fd 1
    set energy energy - 1
    if energy <= 0 [ die ]
    if pcolor = blue [
      set pcolor white
      set energy energy + 2
    ]
    if energy > 15 [
      hatch 1 [set energy 10]
      set energy energy - 10
    ]
  ]
  ask patches [
    if pcolor = white [
      if day = init_day [
        set pcolor blue
        set day 0
      ]
    ]
  ]

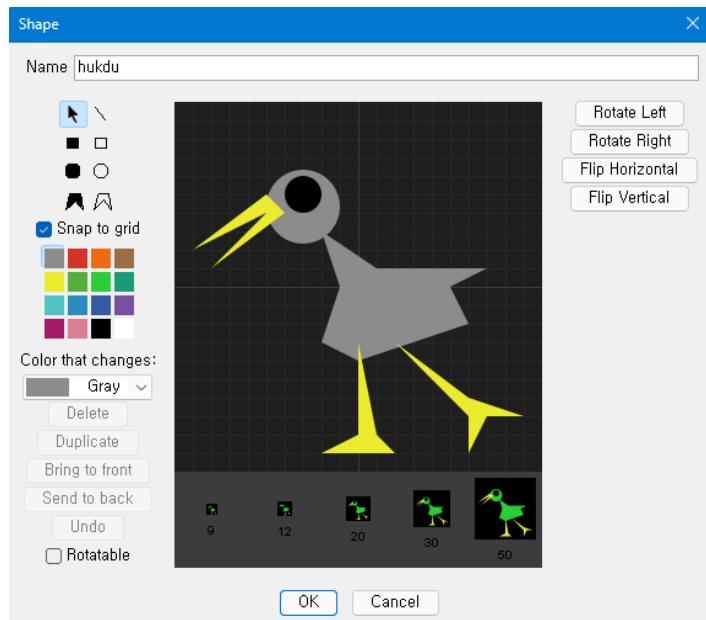
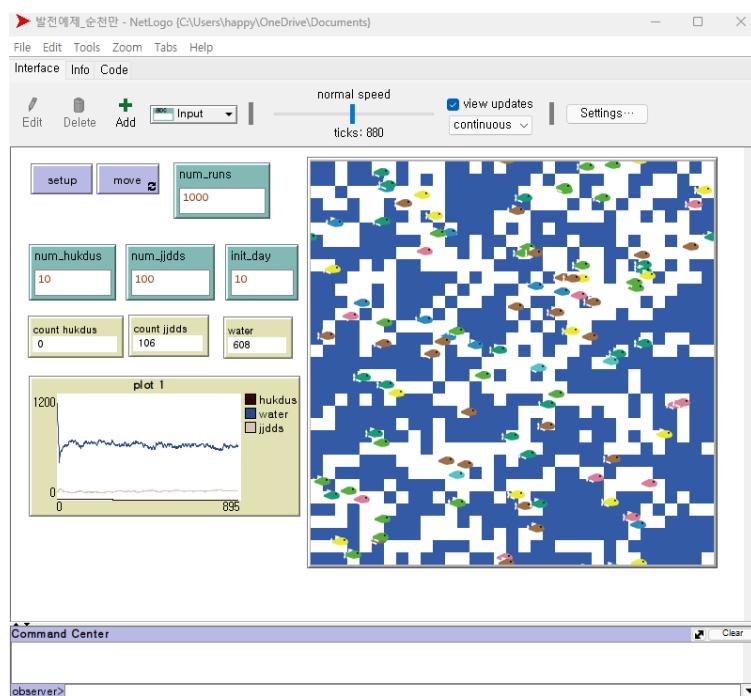
```

```

    set day day + 1
]
]
if count turtles = 0 [stop]
wait 0.05
tick
end

```

발전 예제_순천만(화면)



발전 예제_순천만(코드)

```

breed [hukdus hukdu]
breed [jjdds jjdd]

hukdus-own [energy]
jjdds-own [energy]
patches-own [day]

to setup
  clear-all
  create-hukdus num_hukdus [ set energy 100 ]
  create-jjdds num_jjdds [ set energy 10 ]

  ask patches [
    set pcolor blue
    set day 0
  ]
  ask hukdus [
    setxy random-xcor random-ycor
    set shape "hukdu"
    set size 5
  ]
  ask jjdds [
    setxy random-xcor random-ycor
    set shape "fish"
    set size 1.5
  ]
  reset-ticks
end

to move
  if ticks = num_runs [stop]
  if count hukdus = 0 [stop]

  ask jjdds[
    rt random 90
    lt random 90
    fd 1
    set energy energy - 1
    if energy <= 0 [ die ]
    if pcolor = blue [
      set pcolor white
      set energy energy + 2
  
```

```
]
if energy > 15 [
    hatch 1 [set energy 10]
    set energy energy - 10
]
]
ask hukdus[
    rt random 120
    lt random 120
    fd 1
    set energy energy - 1
    let bab one-of jjdds-here
    if bab != nobody[
        set energy energy + 10
        ask bab [ die ]
    ]
    if energy <= 0 [ die ]
    if energy > 150 [
        set energy energy - 100
        hatch 1 [ set energy energy + 100 ]
    ]
]
ask patches [
    if pcolor = white [
        if day = init_day [
            set pcolor blue
            set day 0
        ]
        set day day + 1
    ]
]
if count turtles = 0 [stop]
wait 0.02
tick
end
```

Experiment

Experiment name

Vary variables as follows (note brackets and quotation marks):

```
["num_jjdds" 100 50]
["init_day" 30 20 10]
["num_hukdus" 10 5]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions
run each combination this many times

Run combinations in sequential order
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

```
count hukdus
count jjdds
```

one reporter per line; you may not split a reporter across multiple lines

Measure runs at every step
If unchecked, runs are measured only when they are over

Setup commands:	Go commands:
<input type="text" value="setup"/>	<input type="text" value="move"/>

Stop condition:

the run stops if this reporter becomes true

Final commands:

run at the end of each run

Time limit
stop after this many steps (0 = no limit)