breed [fish a-fish]

breed [ships ship]

ships-own [capacity]

fish-own [number]

globals [number-dead ship-full max-catched-fish]

to setup-fish

create-fish initial-number-fish

ask fish [

setxy random-pxcor random-pycor

set size 1.5

set color white

set shape "fish"

set number initial-number-fish

]

end

to setup-ships

create-ships initial-number-ships

ask ships [

setxy random-pxcor random-pycor

set size 5

set color black

set shape "boat"

set capacity initial-capacity

]

end

to setup

clear-all

setup-sea

setup-fish

setup-ships

set max-catched-fish tac

reset-ticks

end

to setup-sea

ask patches [

set pcolor blue ]

end

to go

tick

if not any? fish [ stop ]

ask fish [

move

reproduce-fish

]

ask ships [

move

catch-fish

]

end

to move ; turtle procedure

rt random 50

lt random 50

fd 1

end

to reproduce-fish

if random-float 10 < fish-reproduce [

hatch 1 [rt random-float 360 fd 1]

set number count fish

]

end

to catch-fish

ifelse model-version = "accepting-the-TAC-rules"[

ask fish[

if tac \* number > number-dead

[

let prey one-of fish-here

if prey != nobody [

ask prey [die]

set number-dead number-dead + 1

set number (number - number-dead)

]

set ship-full number-dead / initial-capacity

]

]

]

[

let prey one-of fish-here

if prey != nobody [

ask prey [die]

set number-dead number-dead + 1

]

set ship-full number-dead / initial-capacity

]

end