```
globals
[
  water-color              ;;Background color
  male-fish-color          ;;color of male fish
  female-fish-color        ;;color of female fish
  feminized-fish-color     ;;color of feminized male fish
  bird-color               ;;color of birds
  algae-color              ;;color of algae

  fish-stride              ;;distance moved by fish
  bird-stride              ;;distance moved by bird

  bird-size                ;;size of bird sprite
  fish-size                ;;size of fish sprite

  bird-max-age             ;;max age of the birds
  fish-max-age             ;;max age of the fish

  bird-reproduction-age    ;;min age at which birds can reproduce
  fish-reproduction-age    ;;min age at which fish can reproduce

  max-fish-offspring       ;;max number of children fish can have at once
  max-bird-offspring       ;;max number of children bird can have at once

  fish-survival            ;;probability fish makes it to adult-hood
  fish-find-egg            ;;used to determine how likely a male fish impregnates egg

  bird-survival            ;;probability bird makes it to adult-hood
  bird-egg                 ;;used to determine how likely bird lays fertile egg

  algae-growth-delay       ;;delay time before algae grows back
  algae-growth-rate        ;;amount of energy that algae grows back per tick
  algae-max-energy         ;;max amount of food an algae can have

  estrogen-concentration   ;;total concentration of estrogen in lake
  estrogen-resistance      ;;ability of male fish to resist feminization
  estrogen-accumulated-max      ;;max amount of estrogen a male fish can store in body
]

breed [fishes fish]
breed [birds bird]

turtles-own [current-age max-age feminized? male? energy estrogen-accumulated]
patches-own [algae-energy countdown]

to setup
  clear-all

  set-patch-size   13

  set estrogen-concentration 0
  set estrogen-resistance 200.0
  set estrogen-accumulated-max 0.01

  set algae-max-energy    100
  set algae-growth-rate   10
  set algae-growth-delay  15

  set fish-survival       0.30
  set fish-find-egg       10

  set bird-survival       0.15
  set bird-egg            1

  set fish-size     0.8
  set bird-size     1.2

  set fish-stride   0.8
  set bird-stride   0.5

  set max-fish-offspring 2
  set max-bird-offspring 1

  set fish-max-age  100
  set bird-max-age  100

  set fish-reproduction-age  20
  set bird-reproduction-age  20

  set male-fish-color (cyan)
  set female-fish-color (pink)
  set feminized-fish-color (yellow)
  set water-color (blue)
  set algae-color (green)
  set bird-color(orange)

  set-default-shape fishes "fish"
  set-default-shape birds "bird"

  add-water
  add-fish
  add-birds
  reset-ticks
end

to add-water ;;Initializes algae clusters
  ask patches [
    set algae-energy random (algae-max-energy / 3.3)
    color-algae
  ]
end

to add-fish ;;initializes the fish
  create-fishes 400 [
    set color male-fish-color
    set size fish-size
    set max-age fish-max-age
    set current-age 0 + random fish-max-age
    set feminized? false
    set male? true
    set energy 51 + random 50
    set estrogen-accumulated 0
    setxy random world-width random world-height
  ]
  create-fishes 400 [
    set color female-fish-color
    set size fish-size
    set max-age fish-max-age
    set current-age 0 + random fish-max-age
    set feminized? false
    set male? false
    set energy 51 + random 50
    set estrogen-accumulated 0
    setxy random world-width random world-height
  ]
end

to add-birds ;;initializes the birds
  create-birds 75 [
    set color bird-color
    set size bird-size
    set max-age bird-max-age
    set current-age 0 + random bird-max-age
    set feminized? false
    set male? true
    set energy 51 + random 50
    set estrogen-accumulated 0
    setxy random world-width random world-height
  ]
  create-birds 75 [
    set color bird-color
    set size bird-size
    set max-age bird-max-age
    set current-age 0 + random bird-max-age
    set feminized? false
    set male? false
    set energy 51 + random 50
    set estrogen-accumulated 0
    setxy random world-width random world-height
  ]
end

to move-bird ;;procedure to rotate and move birds randomly
  rt random 50 - random 50
  fd bird-stride
end
```

```
to move-fish ;;procedure to rotate and move fish randomly
  rt random 50 - random 50
  fd fish-stride
end

to reproduce-fish ;;procedure to hatch new fish at age 0
  let males (count fishes with [male? = true])
  let fertile-males (count fishes with [male? = true and feminized? = false])
  ;;Reproduction chance for female is based on number of males and base survival rate for eggs
  let reproduction-chance (fish-survival * (fertile-males / (males + fish-find-egg)))
  let reproduction-threshold random-float 1.0
  if (reproduction-chance > reproduction-threshold)
    [
      ;;Spawns a random number of offspring
      let number-male-offspring (random (max-fish-offspring + 1))
      let number-female-offspring (max-fish-offspring - number-male-offspring)
      hatch number-male-offspring
      [
        set color male-fish-color
        set size fish-size
        set max-age fish-max-age
        set current-age 0
        set feminized? false
        set male? true
        set energy 100
        set estrogen-accumulated 0
        rt random 360
        fd random 10
      ]
      hatch number-female-offspring
      [
        set color female-fish-color
        set size fish-size
        set max-age fish-max-age
        set current-age 0
        set feminized? false
        set male? false
        set energy 100
        set estrogen-accumulated 0
        rt random 360
        fd random 10
      ]
    ]
end

to reproduce-bird  ;;procedure to hatch new birds
  let males (count birds with [male? = true])
  ;;Reproduction chance for female is based on number of males and base survival rate for eggs
  let reproduction-chance (bird-survival * (males / (males + bird-egg)))
  let reproduction-threshold random-float 1.0
  if ((reproduction-chance > reproduction-threshold) and (count fishes > count birds))
    [
      ;;Spawns a random number of offspring
      let number-male-offspring (random (max-bird-offspring + 1))
      let number-female-offspring (max-bird-offspring - number-male-offspring)
      hatch number-male-offspring
      [
        set color bird-color
        set size bird-size
        set max-age bird-max-age
        set current-age 0
        set feminized? false
        set male? true
        set energy 51 + random 50
        set estrogen-accumulated 0
        rt random 360
        fd random 10
      ]
      hatch number-female-offspring
      [
        set color bird-color
        set size bird-size
        set max-age bird-max-age
        set current-age 0
        set feminized? false
        set male? false
        set energy 100
        set energy 51 + random 50
        set estrogen-accumulated 0
        rt random 360
        fd random 10
      ]
    ]
end

to eat-algae ;;allows the fish to eat algae
  ifelse algae-energy >= 3.5 [
    set algae-energy (algae-energy - 3.5)
  ]
  [set energy 0]
end

to eat-fish ;;allows the bird to eat a fish
  if (any? fishes-here) [
    ask one-of fishes-here [die]
    ifelse (energy <= 100)
    [set energy energy + 15]
    [set energy 100]
  ]
end

to grow-algae ;;After a random countdown equals 0 grow some algae
```

```
  set countdown (countdown - 1)
  if (countdown <= 0)
  [
    set algae-energy (algae-energy + algae-growth-rate)
    if (algae-energy > algae-max-energy) [set algae-energy algae-max-energy]
  ]
  if (algae-energy < 0) [set algae-energy 0]
  color-algae
end

to color-algae ;;Colors the algae a deeper green based on their energy
  ifelse (algae-energy > 0)
    [set pcolor (scale-color green algae-energy (algae-max-energy * 2) -50)]
    [set pcolor water-color]
end

to age-fish ;;fish life functions
  set current-age (current-age + 1)
  set energy (energy - 1)
  fish-death
end

to age-bird ;;bird life functions
  set current-age (current-age + 1)
  set energy (energy - 10)
  bird-death
end

to fish-death ;;kills organism if they are to old or have not eaten in a while
  if (current-age > max-age) [die]
  if (energy < 0) [die]
end

to bird-death ;;kills organism if they are to old or have not eaten in a while
  if (current-age > max-age) [die]
  if (energy < 0) [die]
end

to store-estrogen ;;Male fish store estrogen in body
  if (estrogen-accumulated > 0) [set estrogen-accumulated estrogen-accumulated - 0.0005]
  if (estrogen-accumulated < 0) [set estrogen-accumulated 0]
  if (estrogen-accumulated < estrogen-accumulated-max) [
    ifelse (current-age < fish-reproduction-age) ;;Juvenile males are more affected by estrogen
    [set estrogen-accumulated (estrogen-accumulated + estrogen-concentration * 0.001)
     if (estrogen-accumulated > 0) [set estrogen-accumulated estrogen-accumulated - 0.0005]
     if (estrogen-accumulated < 0) [set estrogen-accumulated 0]
    ]
    [set estrogen-accumulated (estrogen-accumulated + estrogen-concentration * 0.0005)]
  ]
end

to feminize-fish ;;Each tick the male fish call this function. Simulates temporal effects of estrogen
```

```
  store-estrogen
  let feminize-chance ((estrogen-concentration / estrogen-resistance) + estrogen-accumulated)
  let feminize random-float 1.0
  if (feminize-chance > feminize)
  [
    set feminized? true
    set color feminized-fish-color
    set estrogen-concentration (estrogen-concentration - 0.0015)
  ]
end

to add-estrogen ;;The user can add estrogen to the lake. It increases chance male fish get feminized
  set estrogen-concentration (estrogen-concentration + estrogen-ppt)
  if (estrogen-ppt > 0) [
    ask patches [
      set pcolor (scale-color violet estrogen-ppt 200 0)
    ]
  ]
end

to go ;;Every tick call basic life functions of fish, birds, and algae
  if ticks >= 1000 and constant-simulation-length? [stop]
  ask fishes [
    move-fish
    ifelse male? = false
    [reproduce-fish]
    [feminize-fish]
    eat-algae
    age-fish
  ]
  ask birds [
    move-bird
    if male? = false
    [reproduce-bird]
    eat-fish
    age-bird
  ]
  ask patches [
    set countdown random algae-growth-delay
    grow-algae
  ]
  tick
end
```