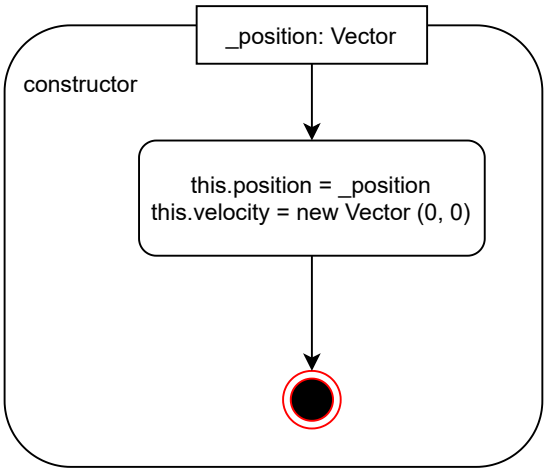
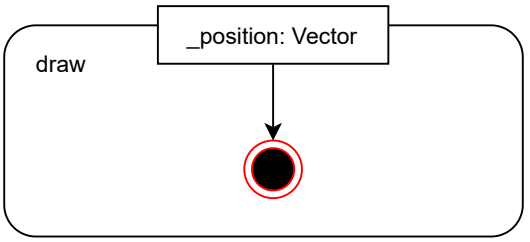
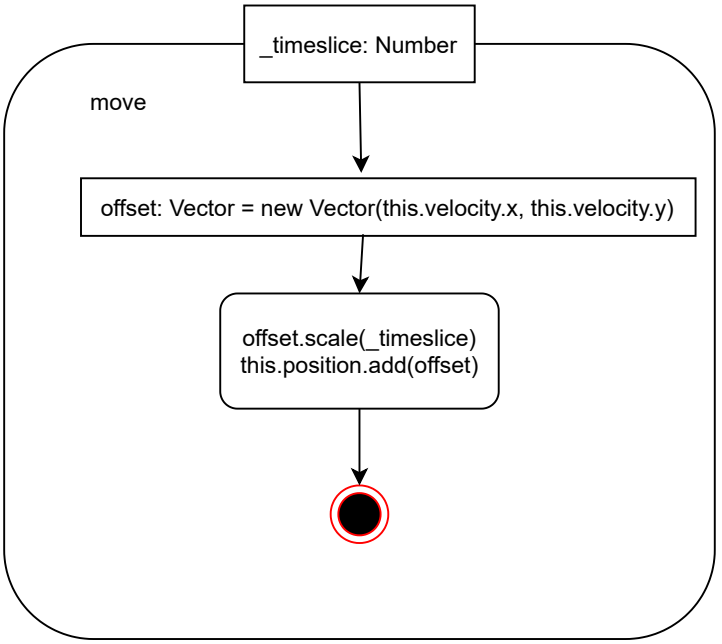
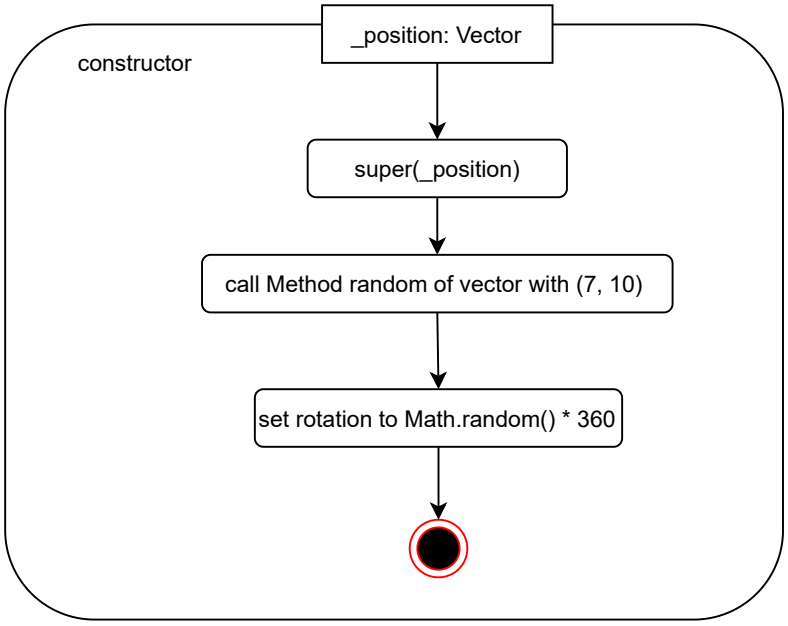
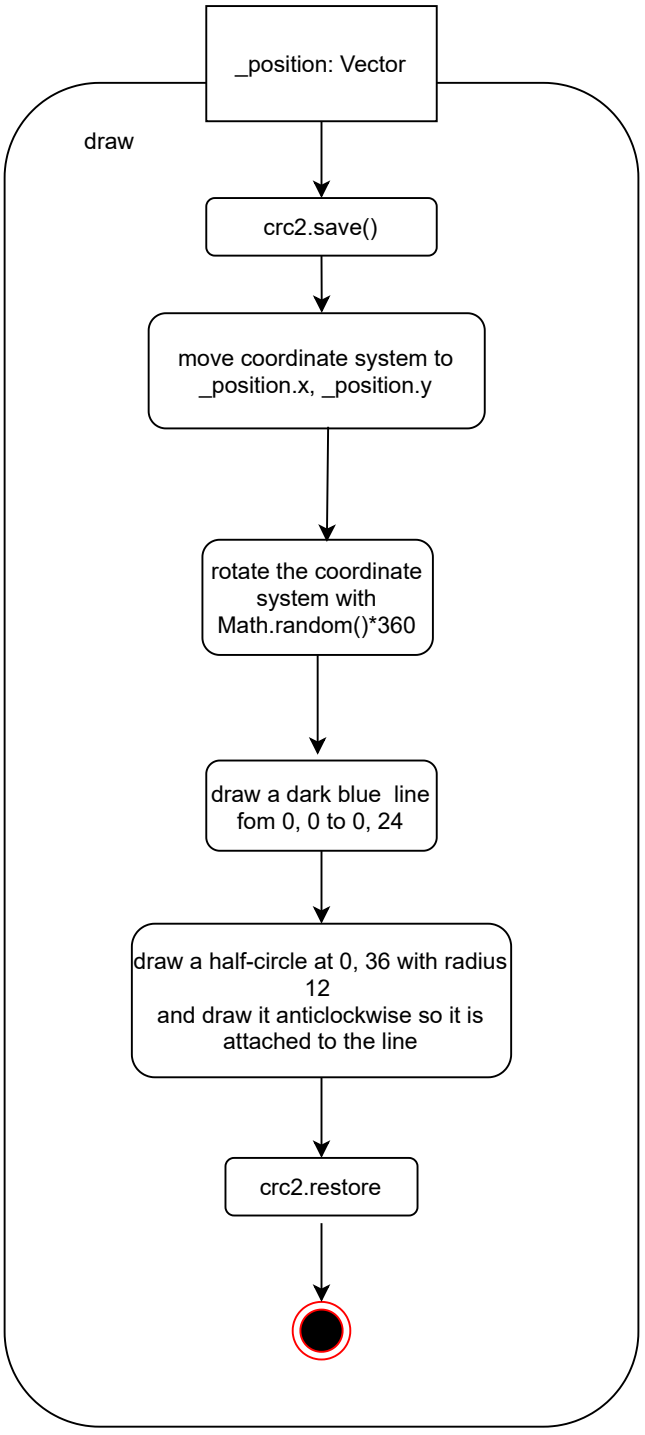
L10_Cell AD

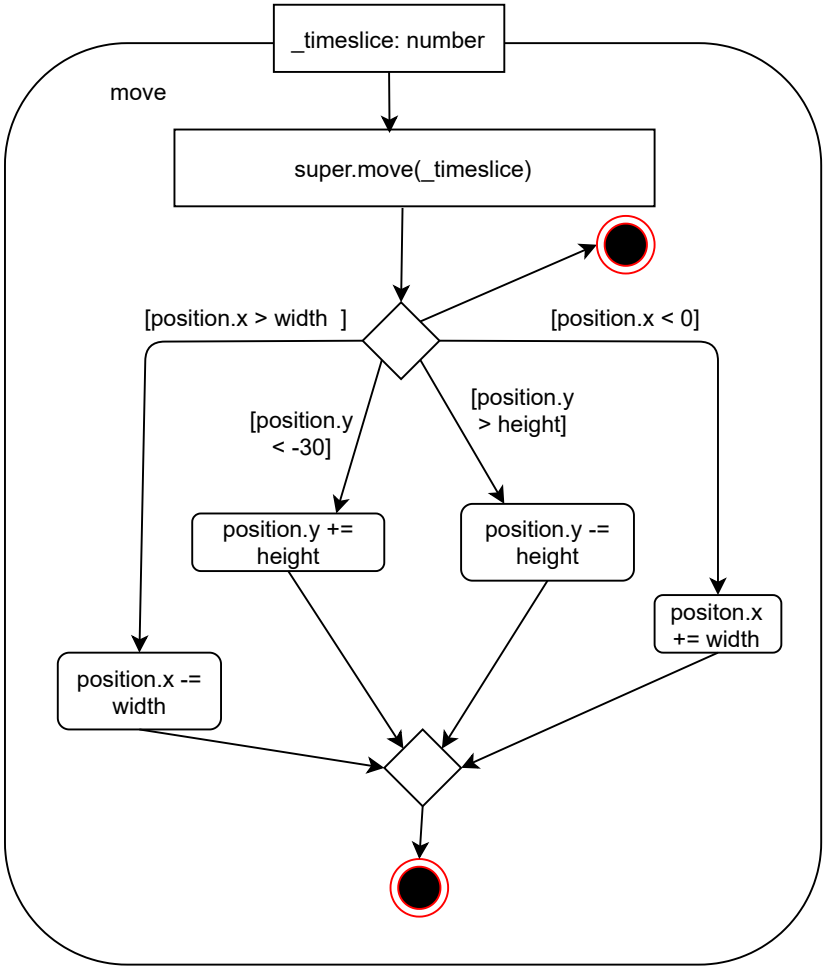

_position: Vector
constructor
this.position = _position
this.velocity = new Vector (0, 0)

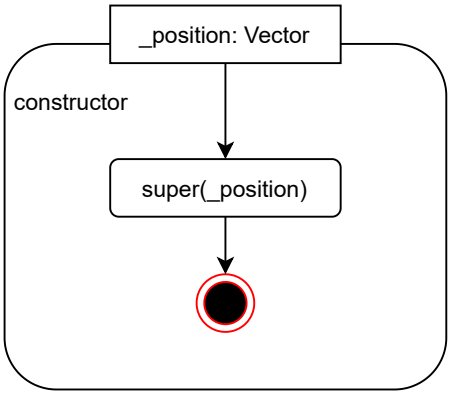

_position: Vector
draw

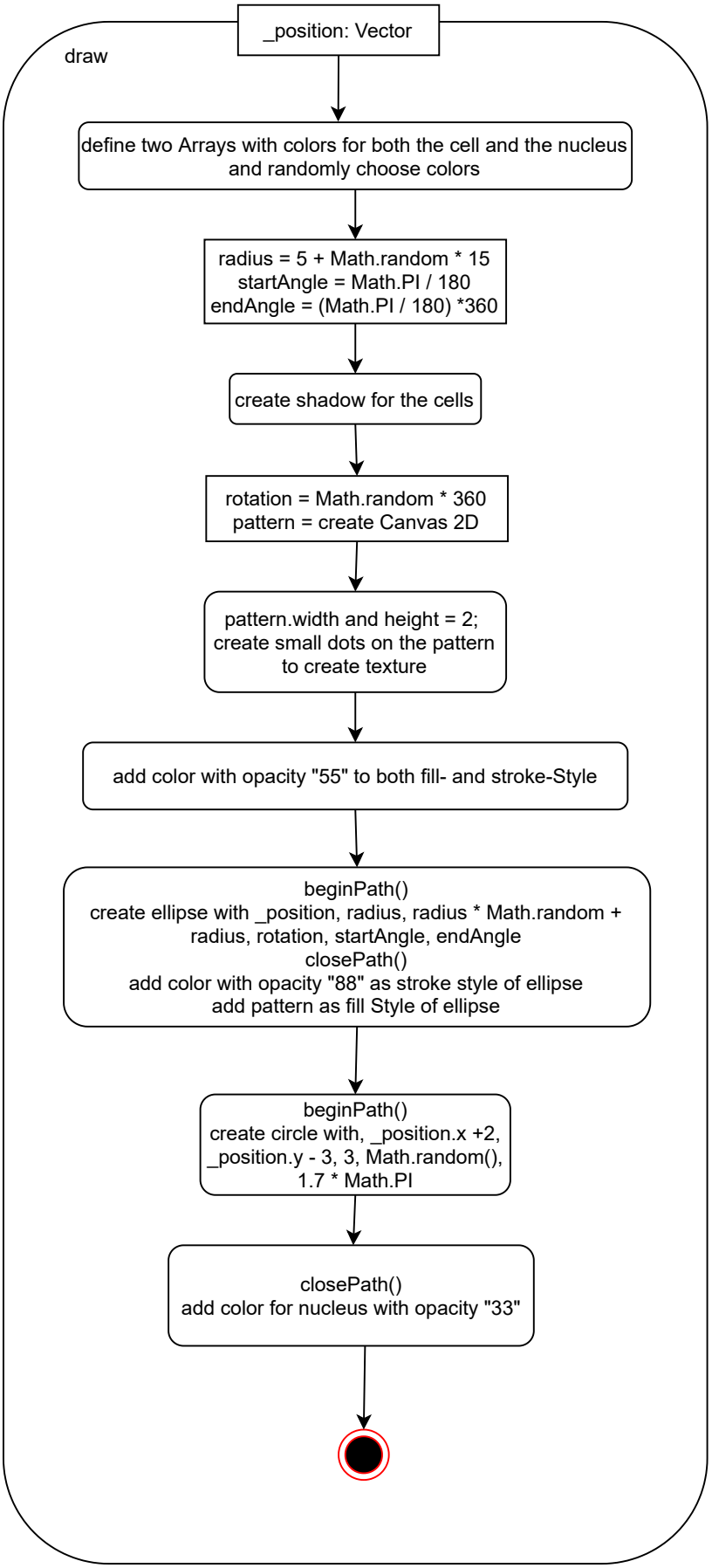

_timeslice: Number
move
offset: Vector = new Vector(this.velocity.x, this.velocity.y)
offset.scale(_timeslice)
this.position.add(offset)


constructor
_position: Vector
super(_position)
call Method random of vector with (7, 10)
set rotation to Math.random() * 360

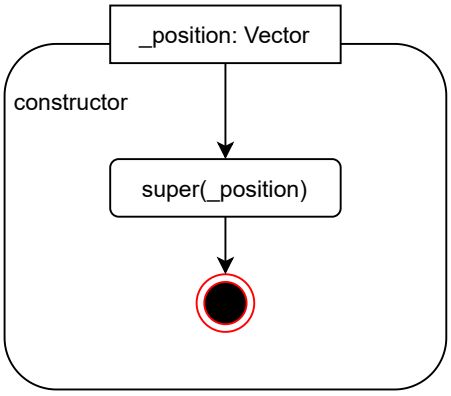

draw
_position: Vector
crc2.save()
move coordinate system to _position.x, _position.y
rotate the coordinate system with Math.random()*360
draw a dark blue line fom 0, 0 to 0, 24
draw a half-circle at 0, 36 with radius 12
and draw it anticlockwise so it is attached to the line
crc2.restore

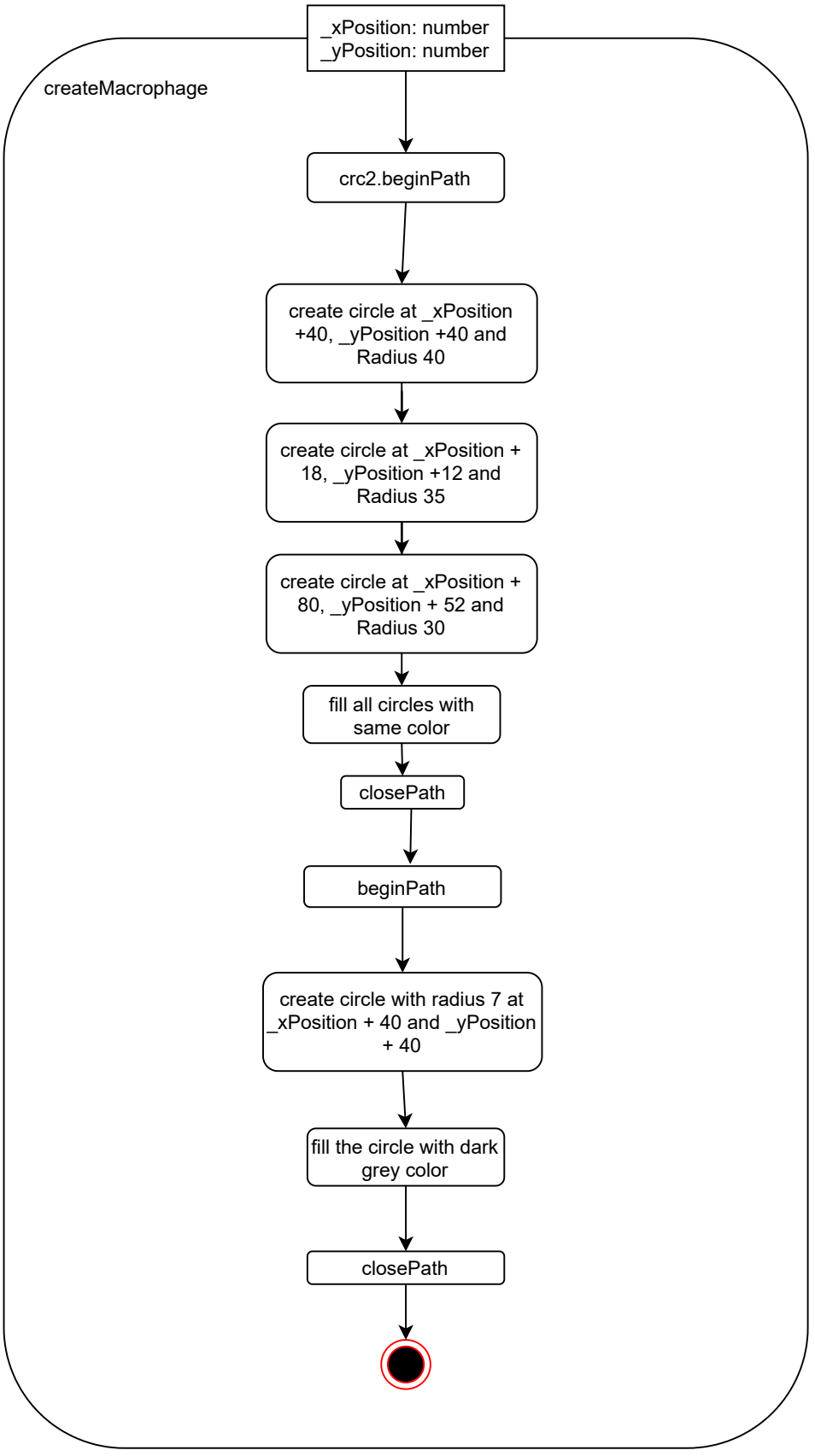

move
_timeslice: number
super.move(_timeslice)
[position.x > width]
[position.x < 0]
[position.y < -30]
[position.y > height]
position.y += height
position.y -= height
positon.x += width
position.x -= width


_position: Vector
constructor
super(_position)

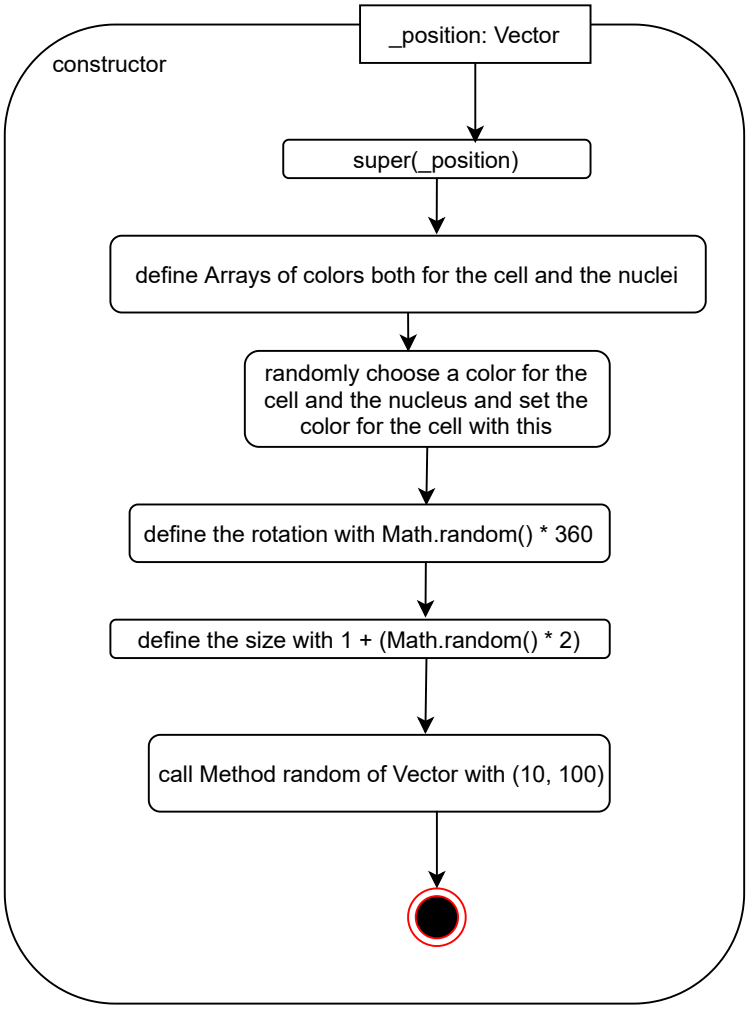

_position: Vector
draw
define two Arrays with colors for both the cell and the nucleus and randomly choose colors
radius = 5 + Math.random * 15
startAngle = Math.PI / 180
endAngle = (Math.PI / 180) *360
create shadow for the cells
rotation = Math.random * 360
pattern = create Canvas 2D
pattern.width and height = 2;
create small dots on the pattern to create texture
add color with opacity "55" to both fill- and stroke-Style
beginPath()
create ellipse with _position, radius, radius * Math.random + radius, rotation, startAngle, endAngle
closePath()
add color with opacity "88" as stroke style of ellipse
add pattern as fill Style of ellipse
beginPath()
create circle with, _position.x +2, _position.y - 3, 3, Math.random(), 1.7 * Math.PI
closePath()
add color for nucleus with opacity "33"

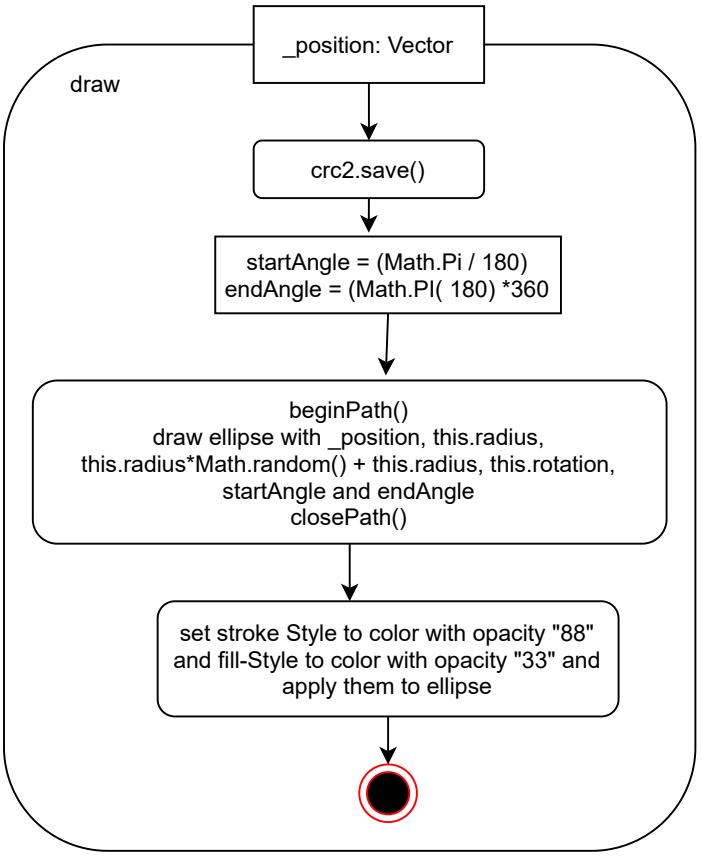
L10 Macrophage AD

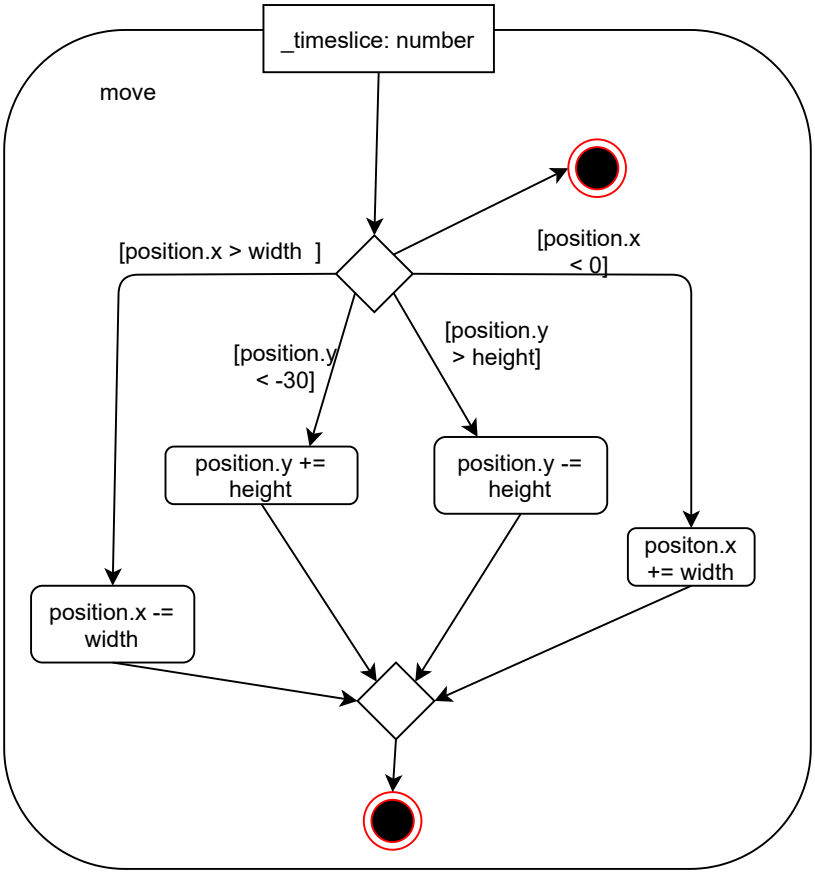




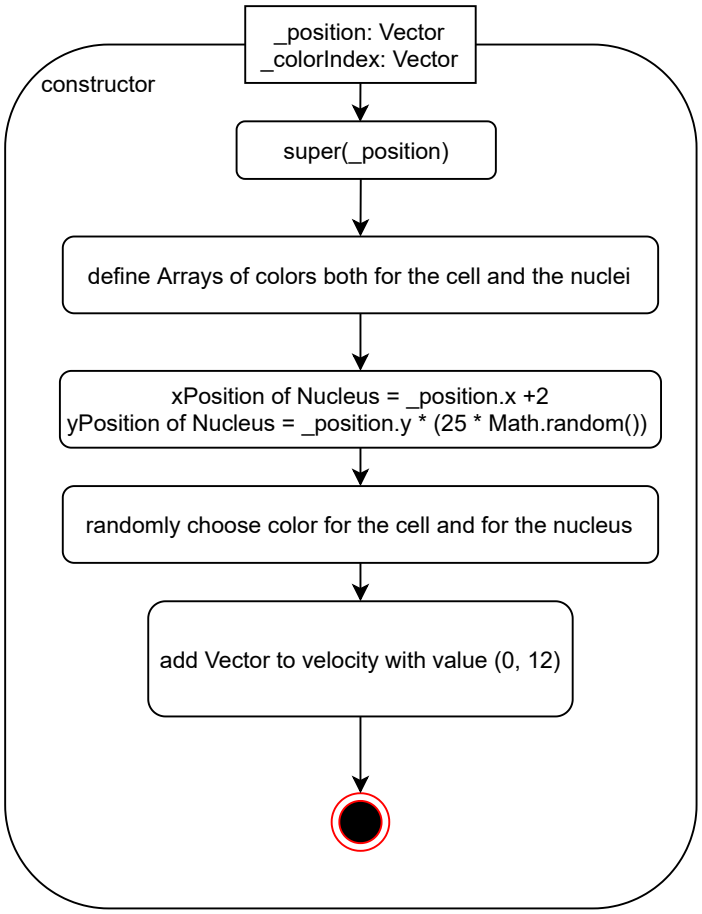
L10 Particle AD

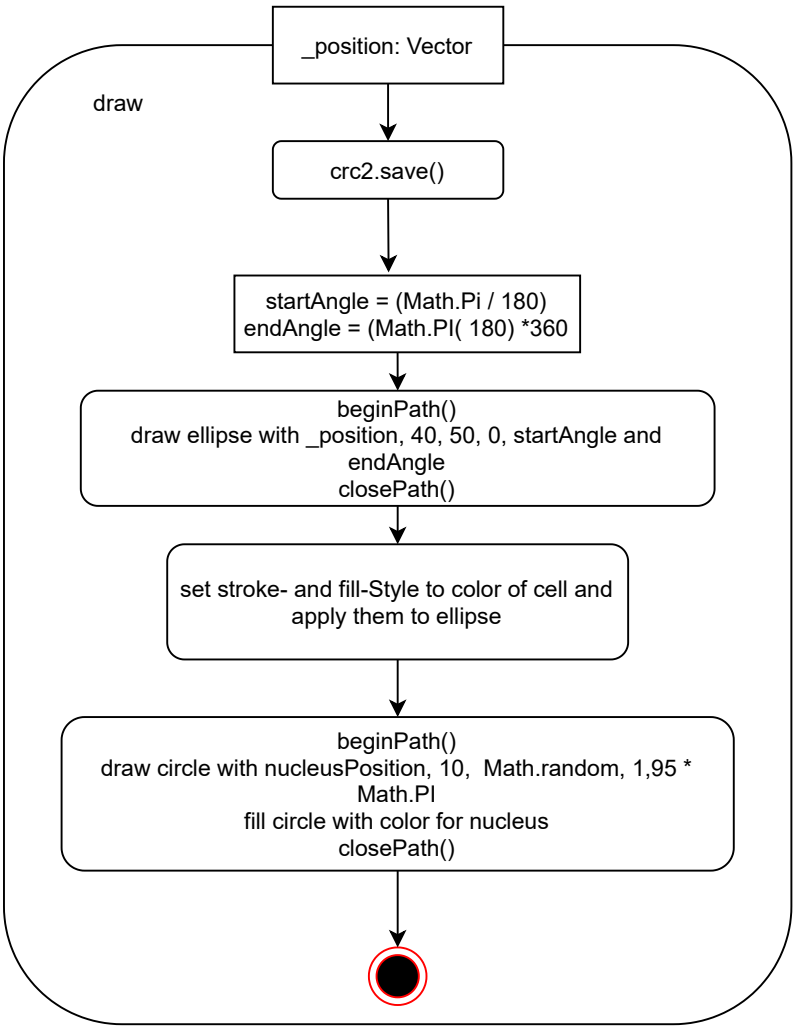


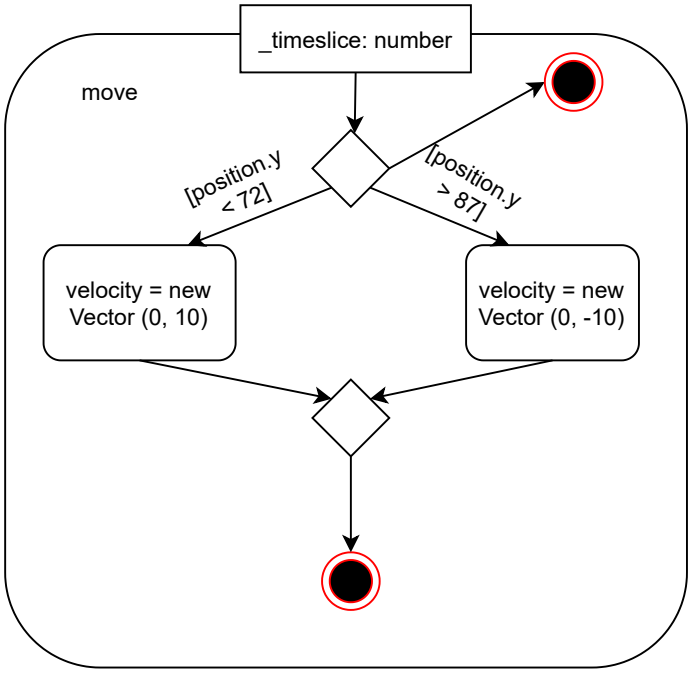


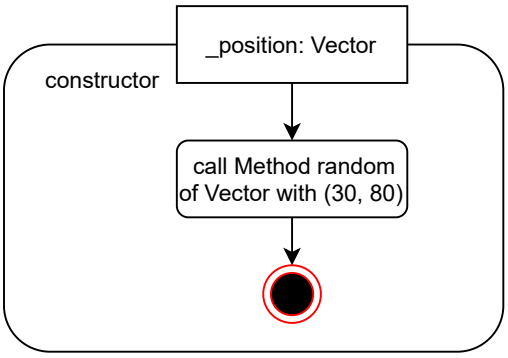


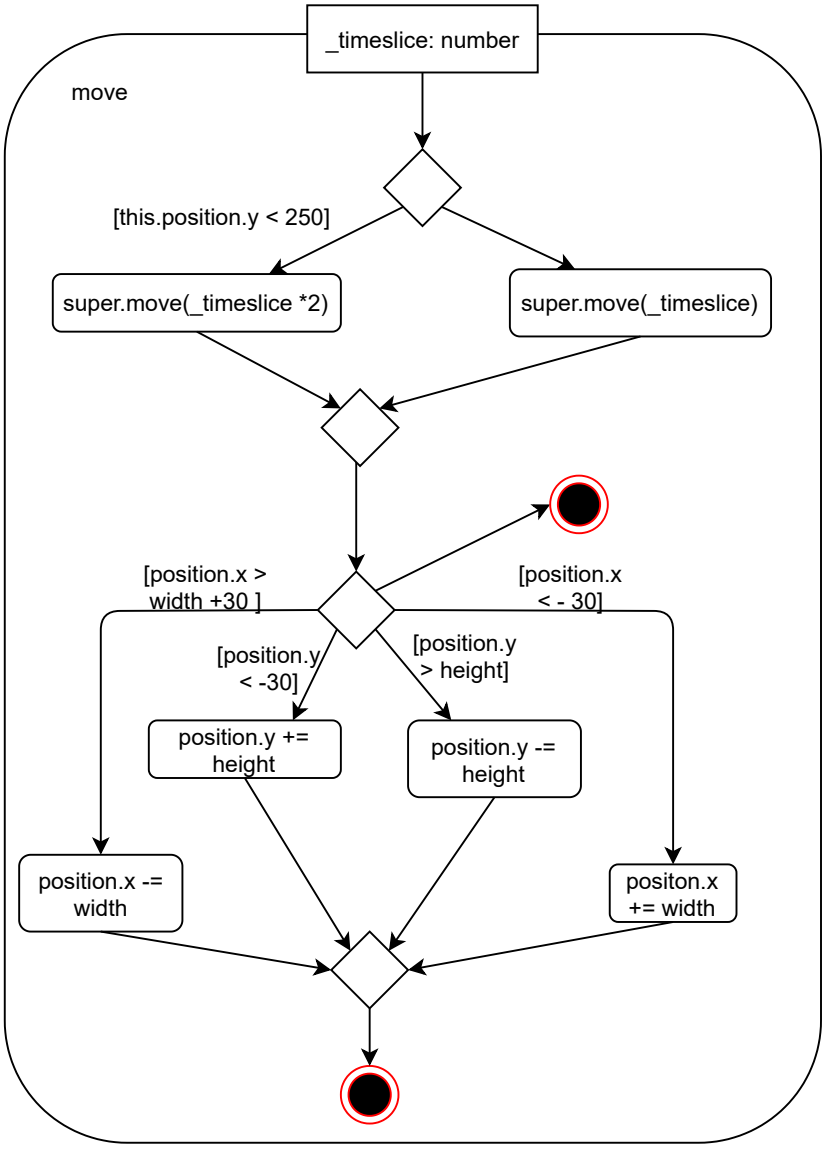
L10 BodyCell AD

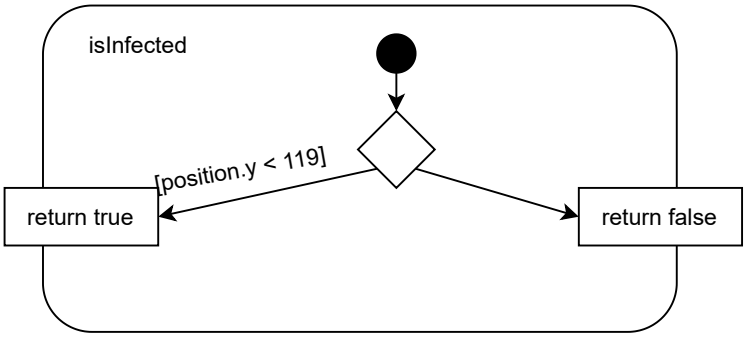


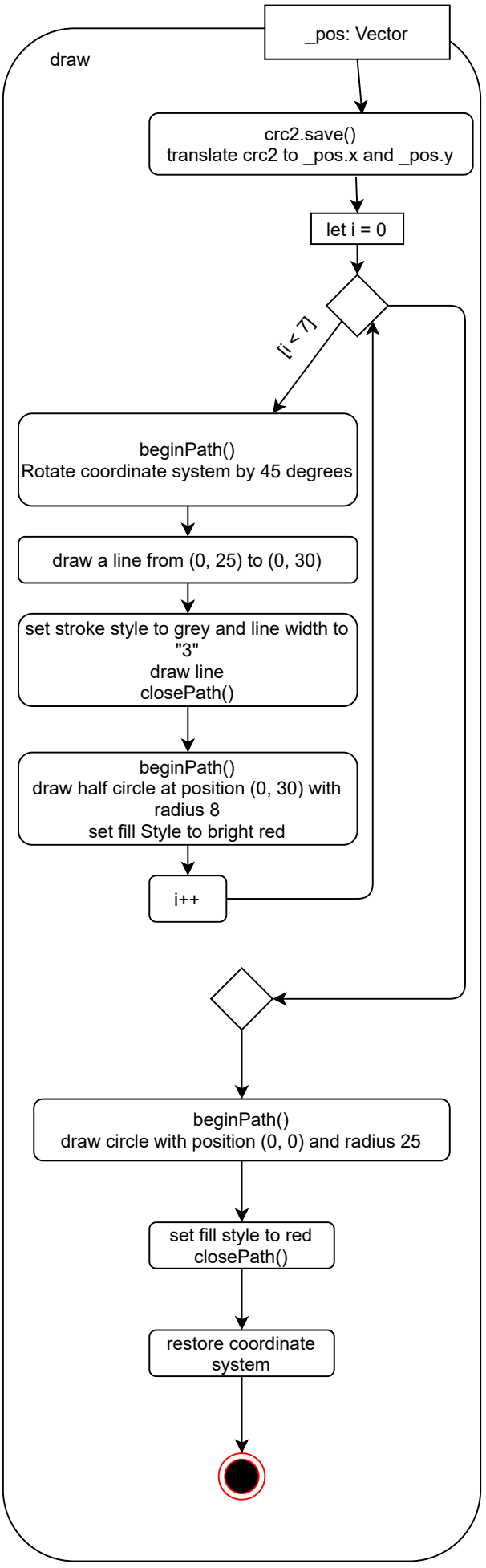


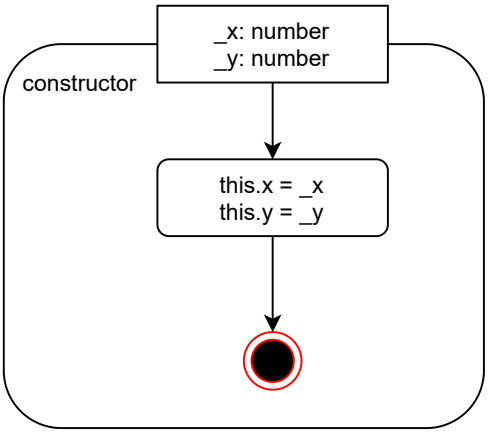


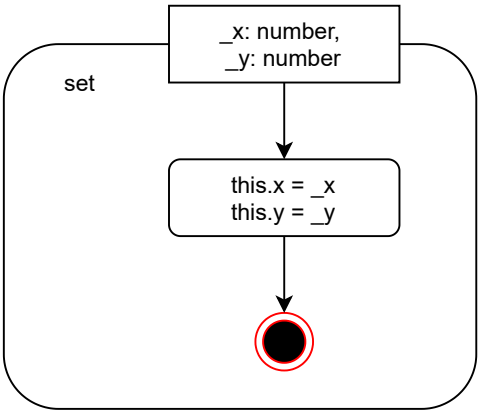

constructor
_position: Vector
call Method random
of Vector with (30, 80)

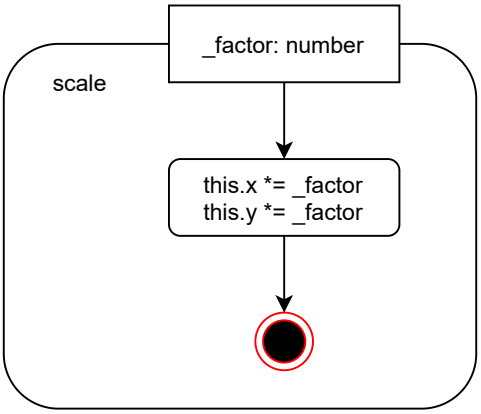

move
_timeslice: number
[this.position.y < 250]
super.move(_timeslice *2)
super.move(_timeslice)
[position.x >
width +30]
[position.x
< - 30]
[position.y
< -30]
[position.y
> height]
position.y +=
height
position.y -=
height
position.x -=
width
positon.x
+= width

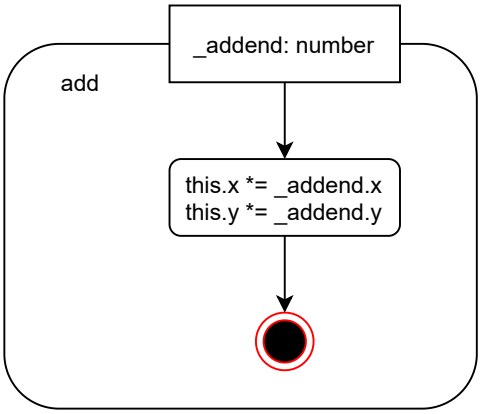

isInfected
[position.y < 119]
return true
return false

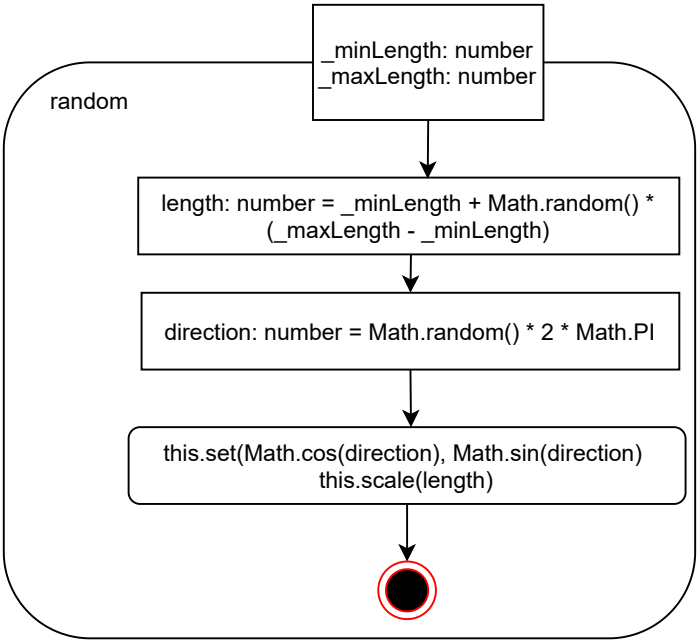

draw
_pos: Vector
crc2.save()
translate crc2 to _pos.x and _pos.y
let i = 0
[i < 7]
beginPath()
Rotate coordinate system by 45 degrees
draw a line from (0, 25) to (0, 30)
set stroke style to grey and line width to
"3"
draw line
closePath()
beginPath()
draw half circle at position (0, 30) with
radius 8
set fill Style to bright red
i++
beginPath()
draw circle with position (0, 0) and radius 25
set fill style to red
closePath()
restore coordinate
system


constructor
_x: number
_y: number
this.x = _x
this.y = _y


set
_x: number,
_y: number
this.x = _x
this.y = _y


scale
_factor: number
this.x *= _factor
this.y *= _factor


add
_addend: number
this.x *= _addend.x
this.y *= _addend.y


random
_minLength: number
_maxLength: number
length: number = _minLength + Math.random() *
(_maxLength - _minLength)
direction: number = Math.random() * 2 * Math.PI
this.set(Math.cos(direction), Math.sin(direction)
this.scale(length)