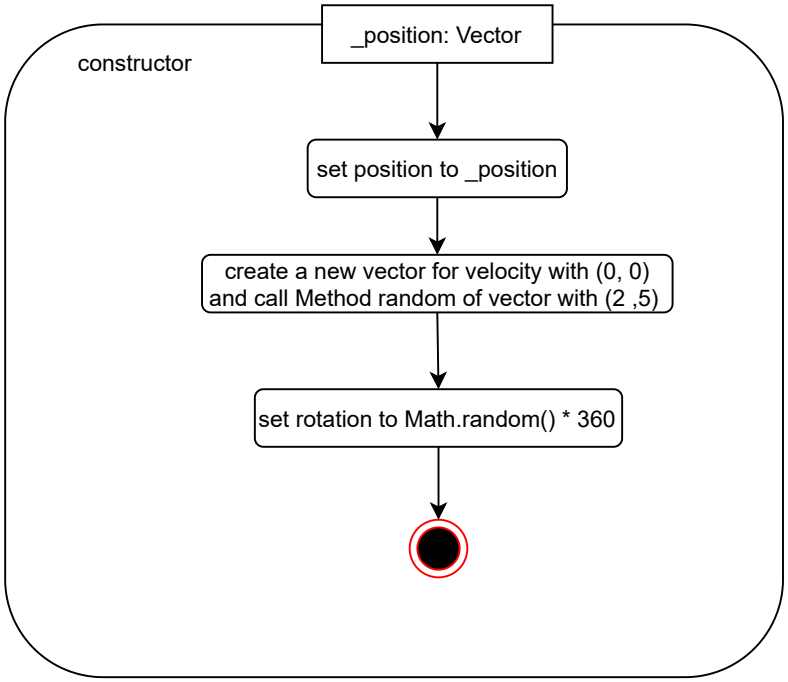
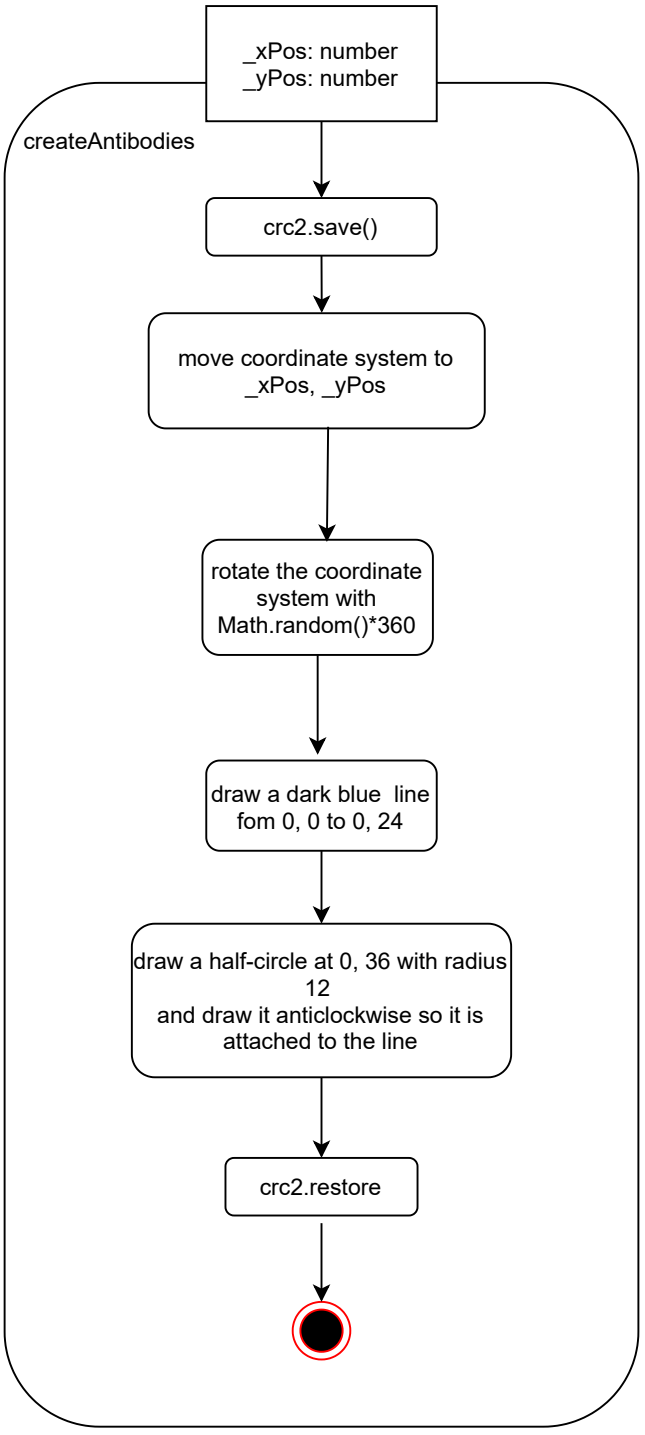
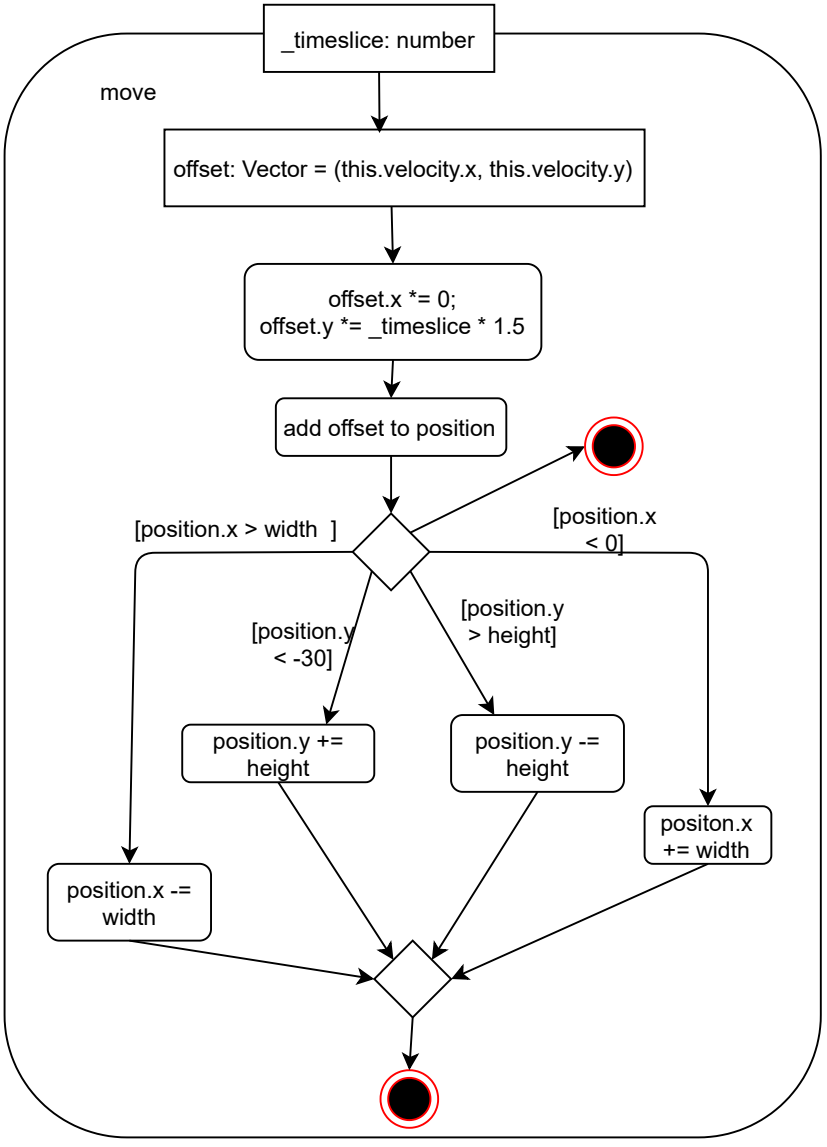
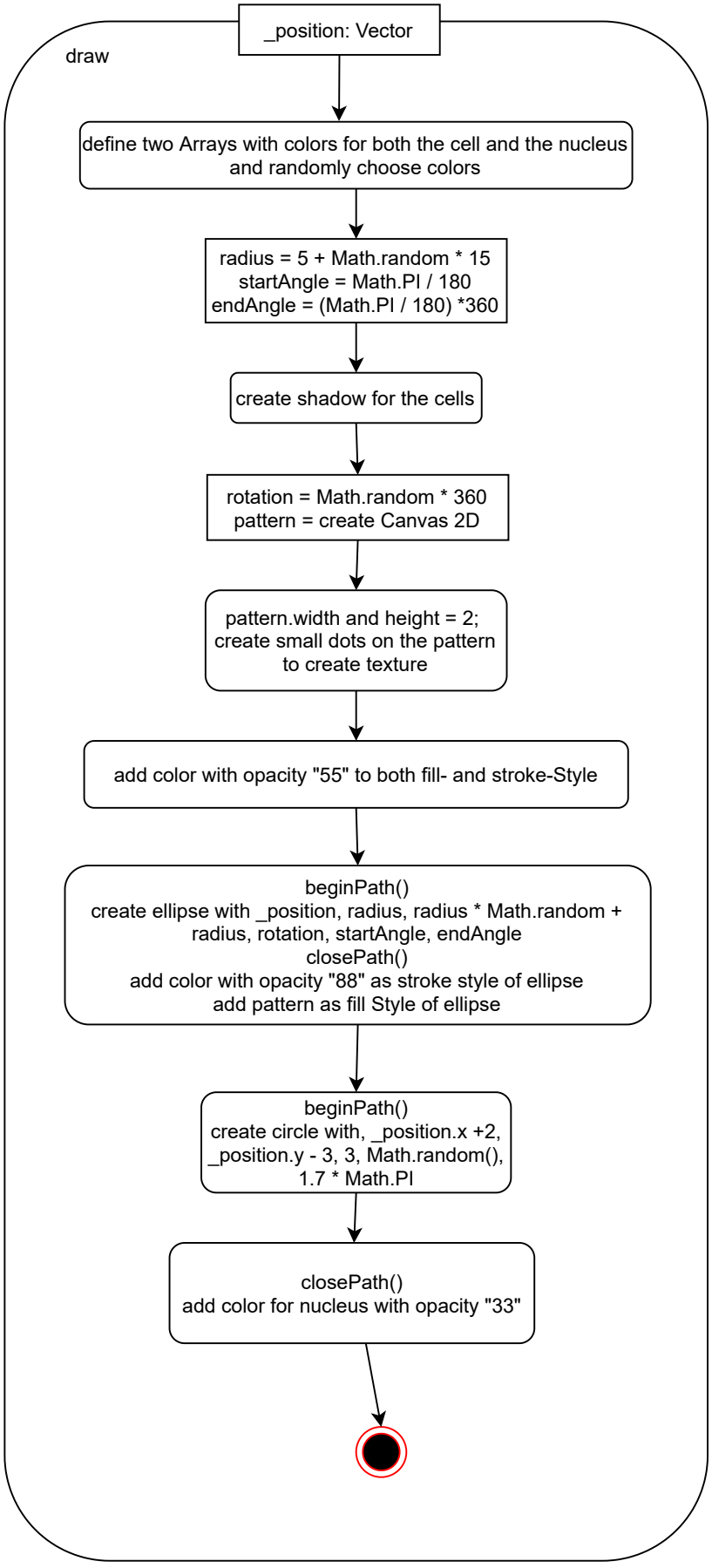

constructor
_position: Vector
set position to _position
create a new vector for velocity with (0, 0)
and call Method random of vector with (2 ,5)
set rotation to Math.random() * 360

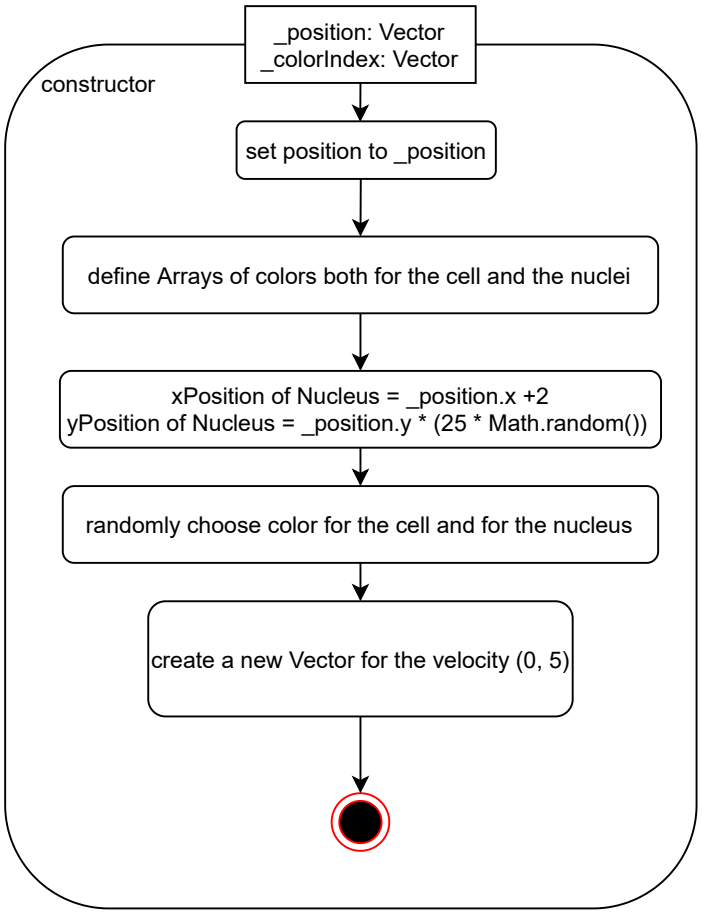

createAntibodies
_xPos: number
_yPos: number
crc2.save()
move coordinate system to
_xPos, _yPos
rotate the coordinate
system with
Math.random()*360
draw a dark blue line
fom 0, 0 to 0, 24
draw a half-circle at 0, 36 with radius
12
and draw it anticlockwise so it is
attached to the line
crc2.restore

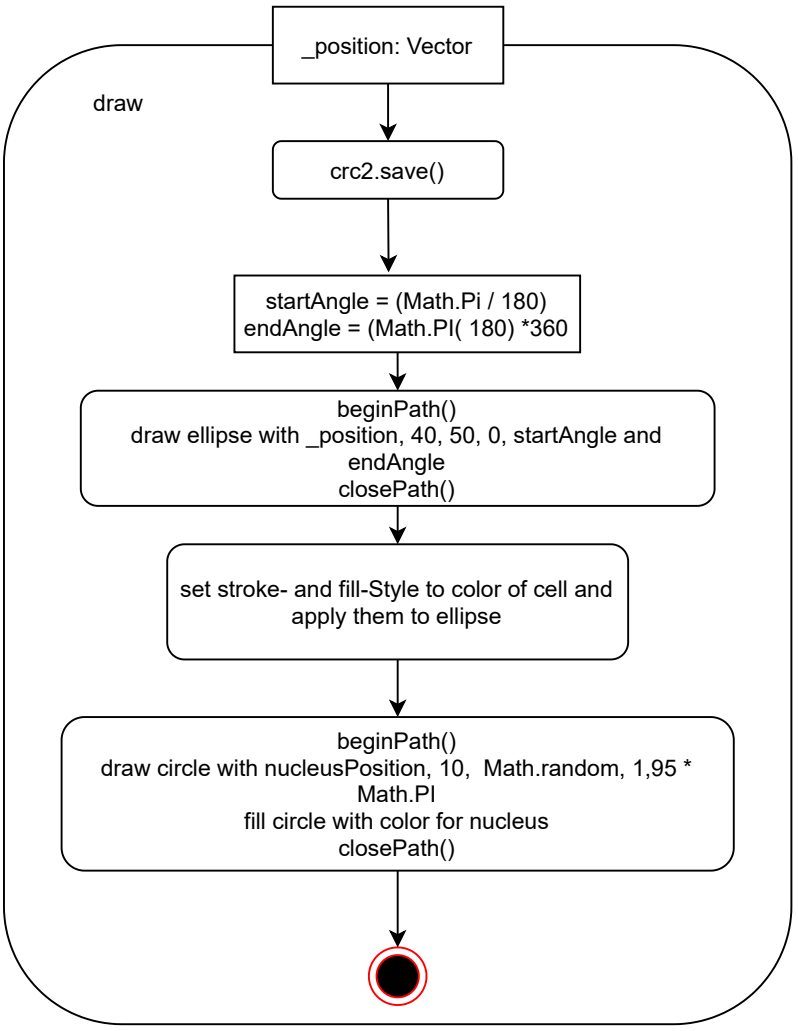

move
_timeslice: number
offset: Vector = (this.velocity.x, this.velocity.y)
offset.x *= 0;
offset.y *= _timeslice * 1.5
add offset to position
[position.x > width]
[position.x
< 0]
[position.y
< -30]
[position.y
> height]
position.y +=
height
position.y -=
height
positon.x
+= width
position.x -=
width

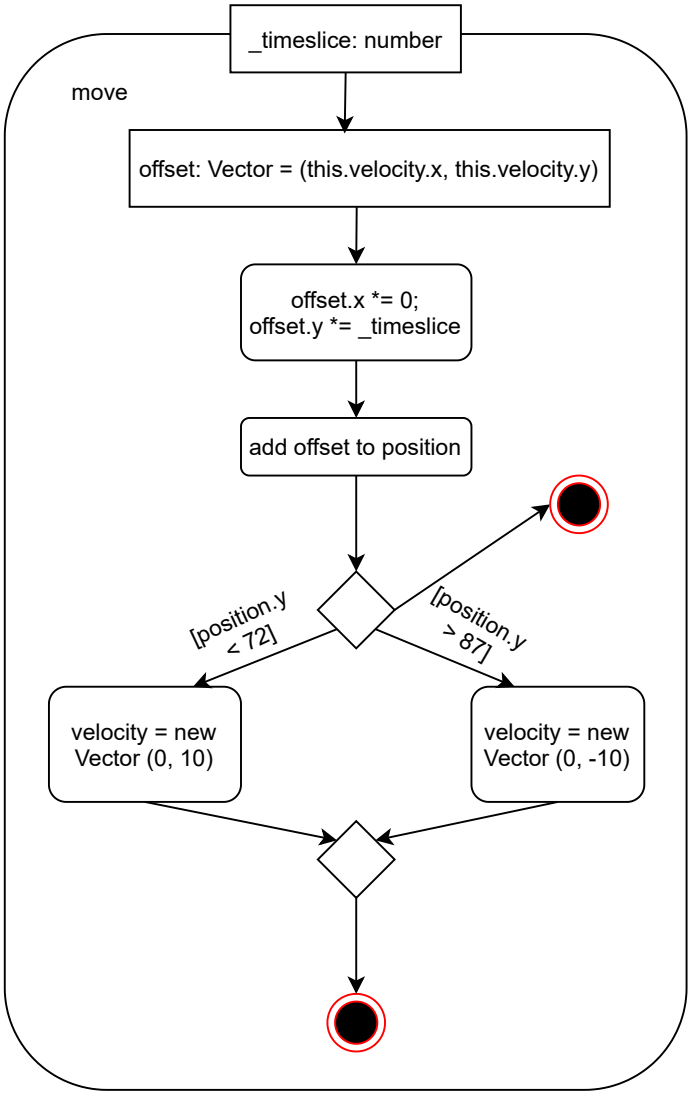
L09 Background-Cells AD



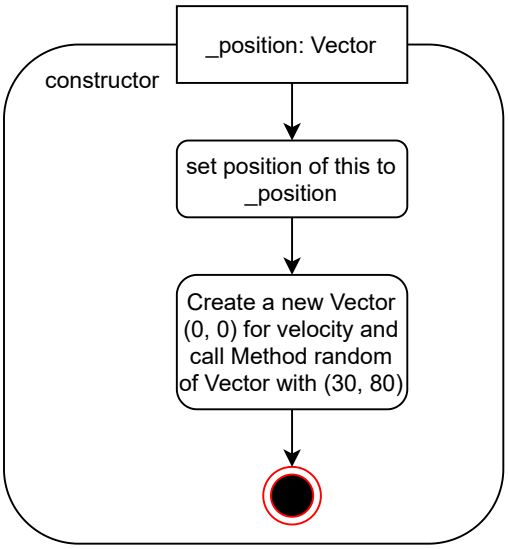
L09 BodyCell AD

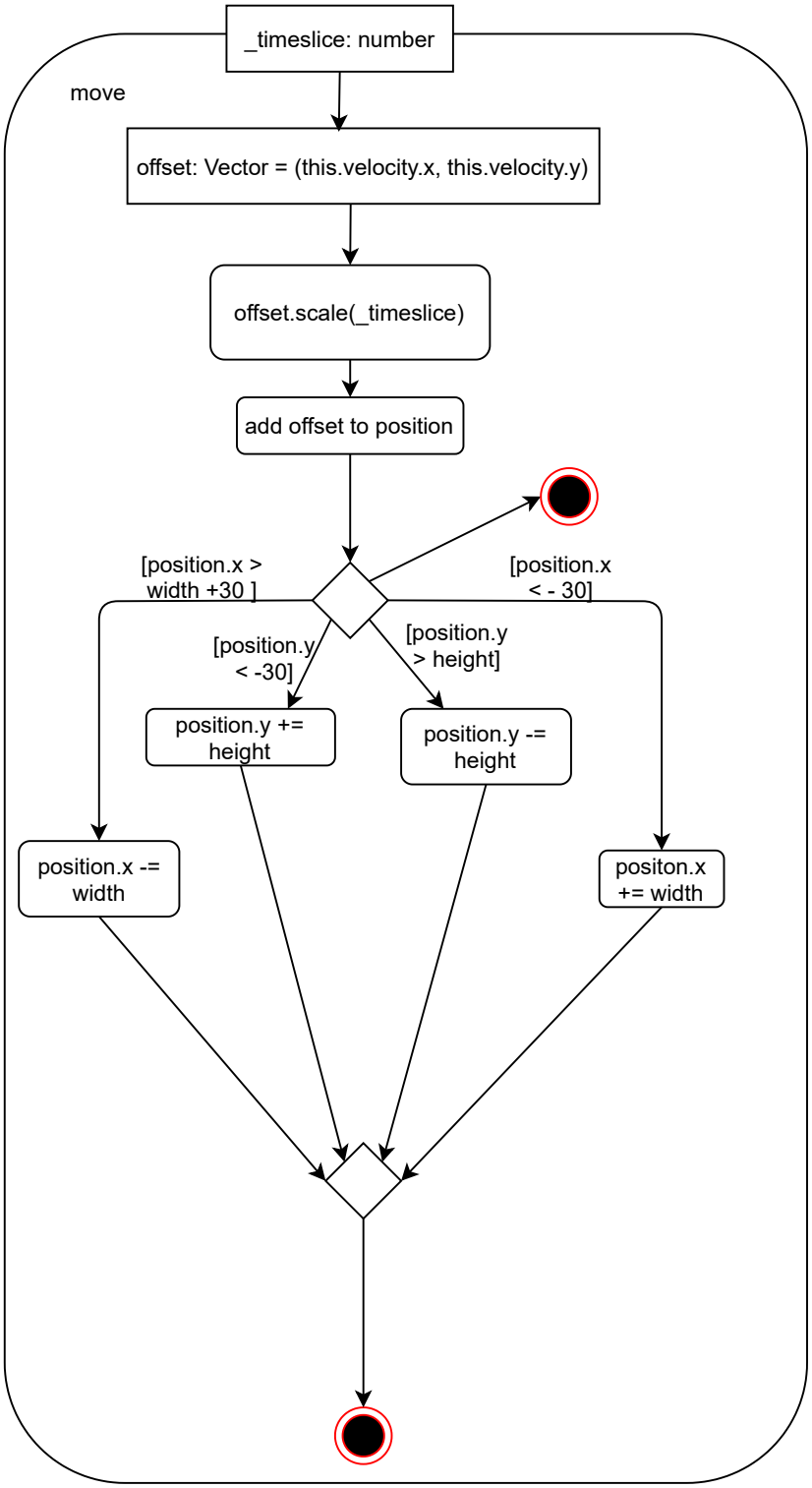


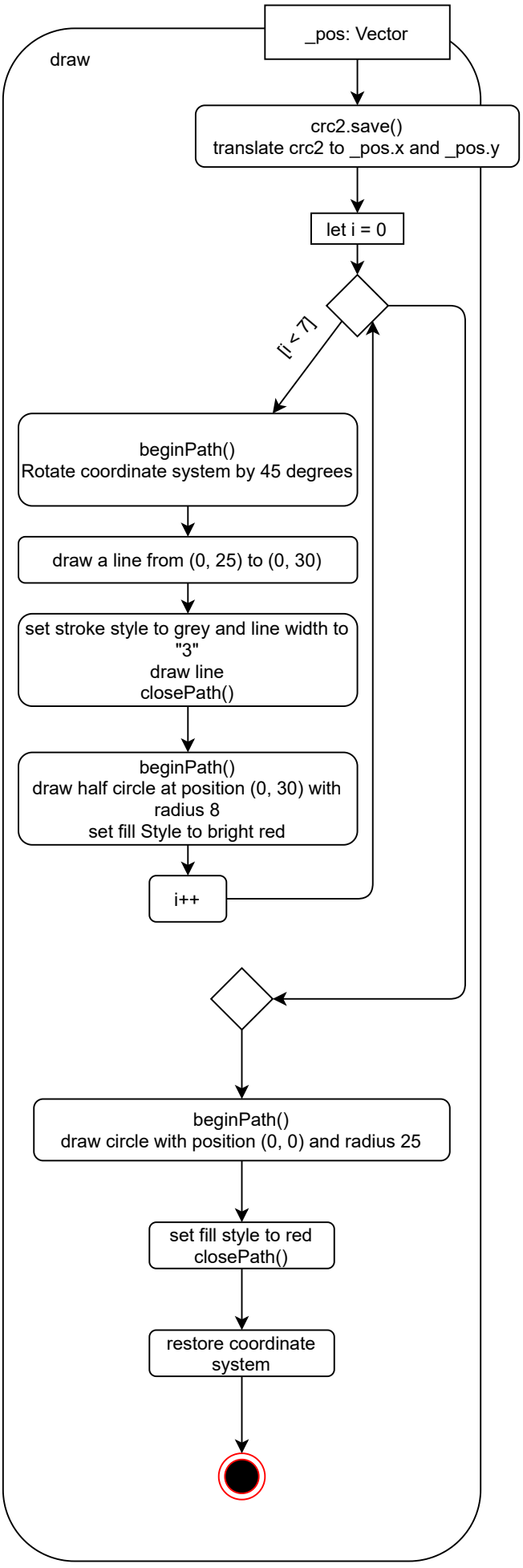


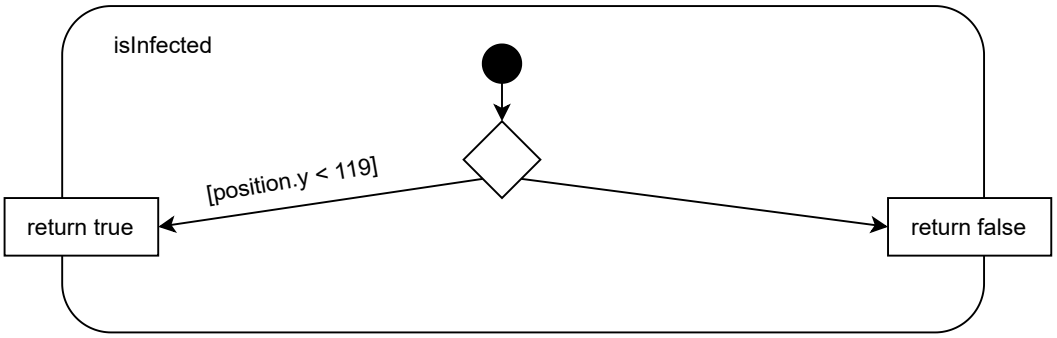


L09 Corona AD

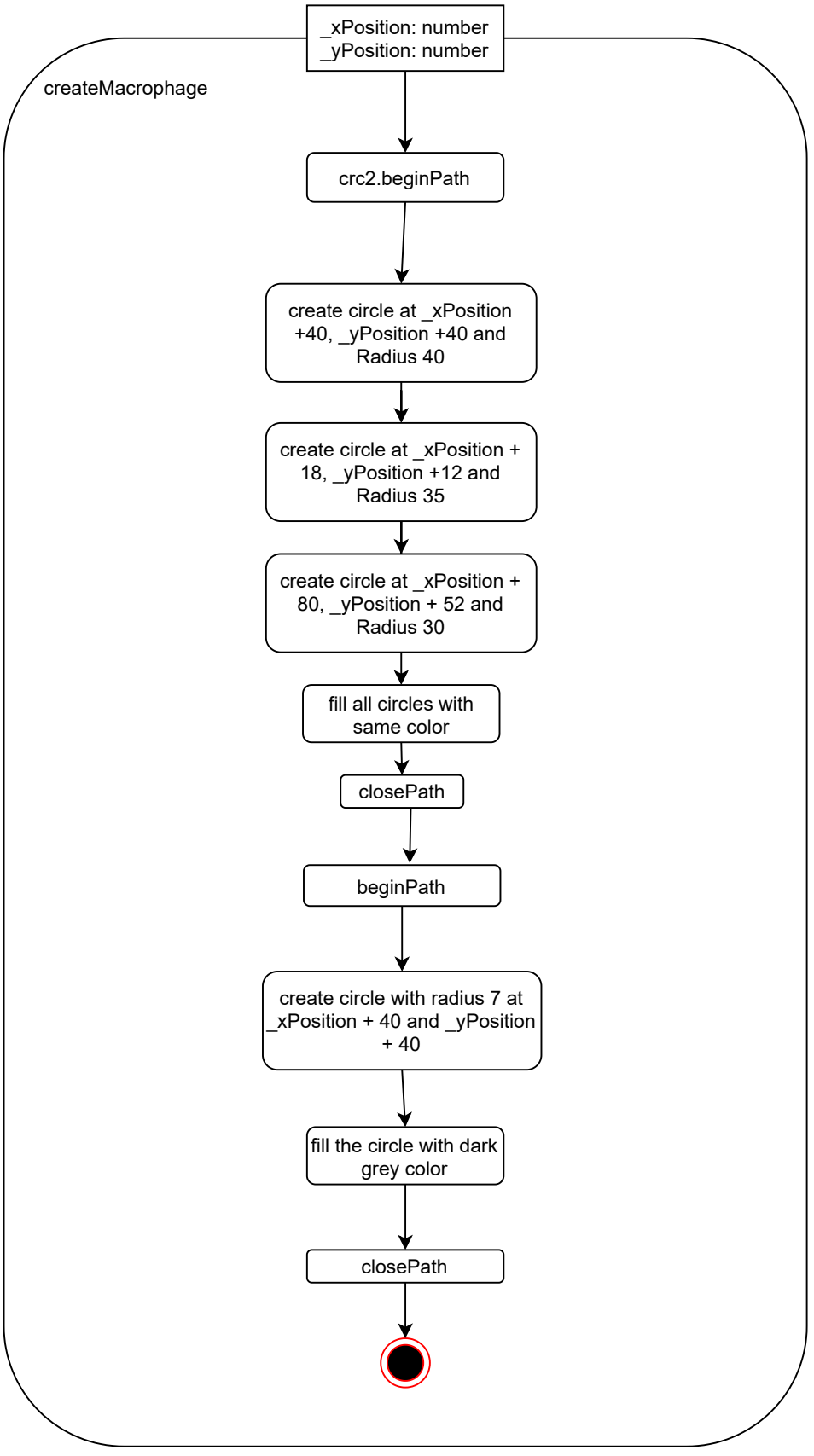


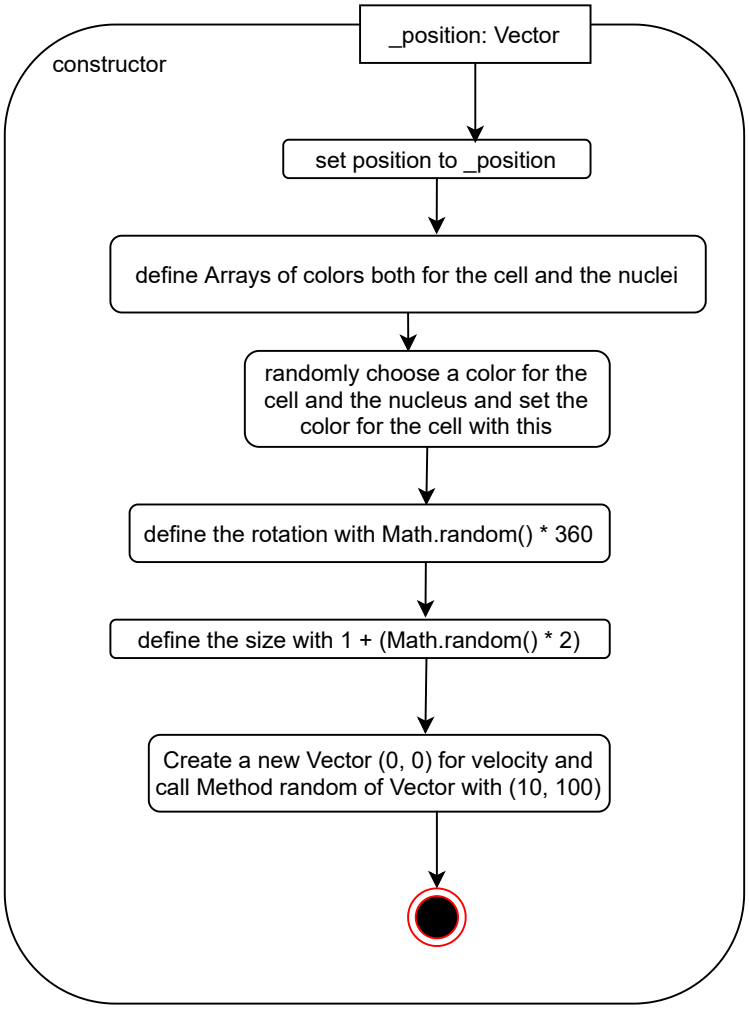


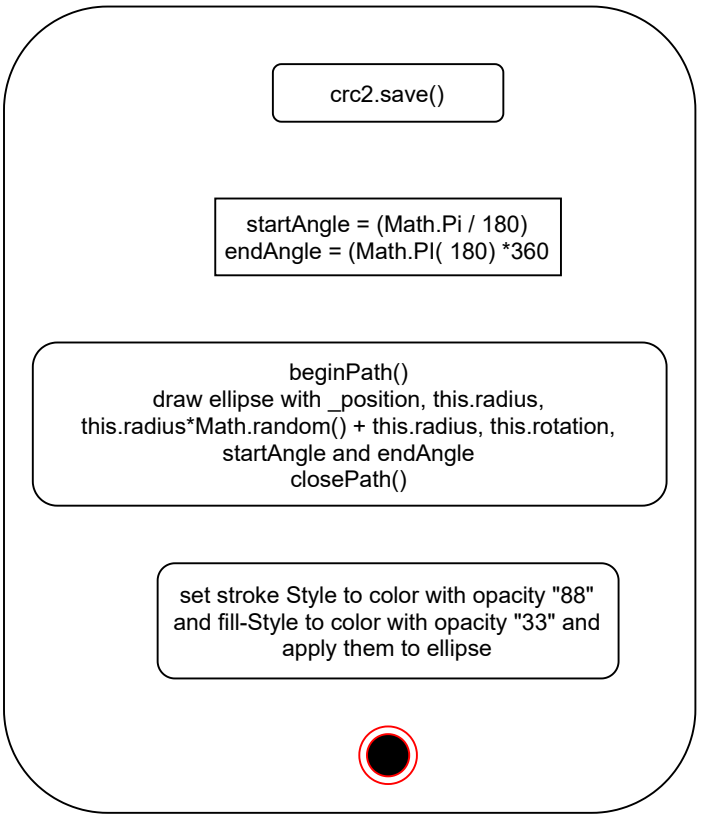


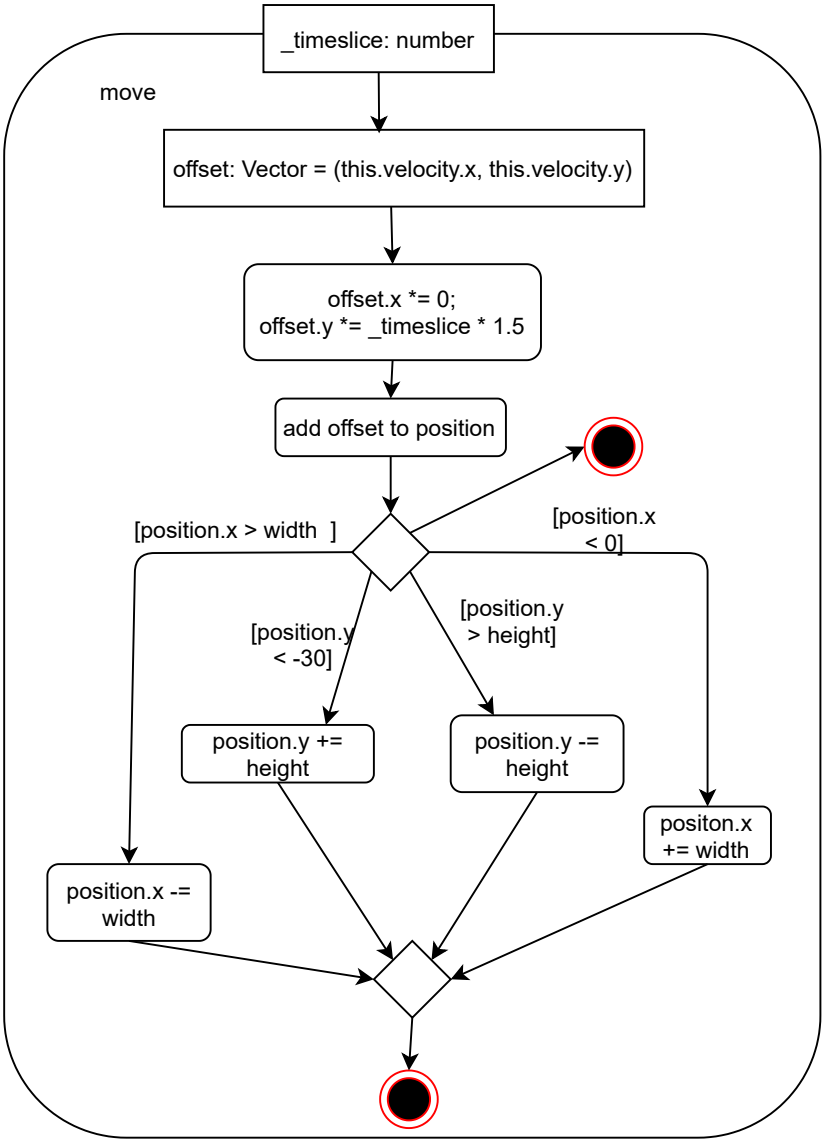

isInfected
[position.y < 119]
return true
return false

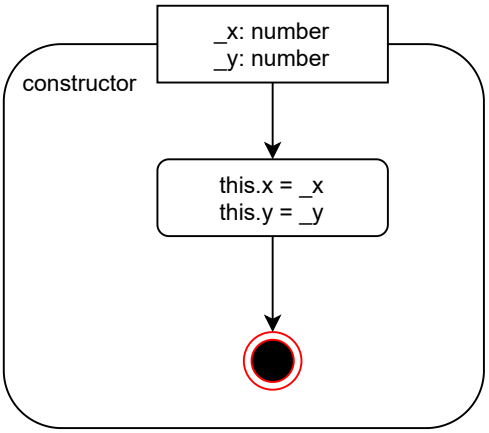
L09 Macrophage AD

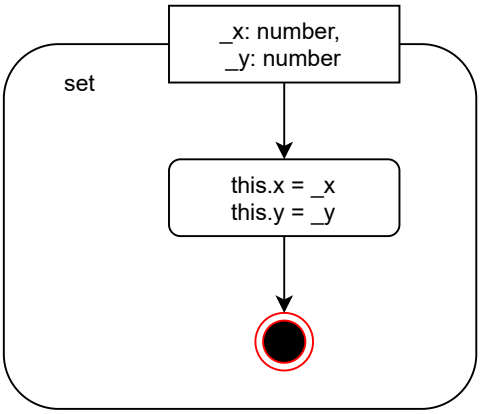


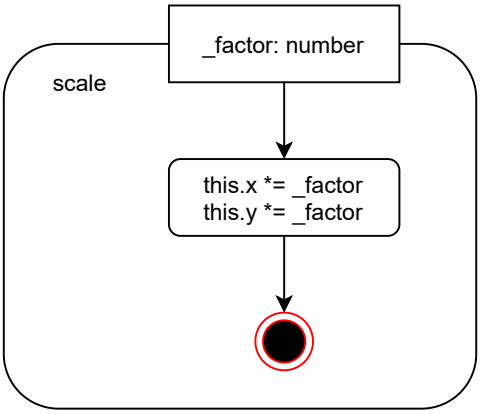

_position: Vector
constructor
set position to _position
define Arrays of colors both for the cell and the nuclei
randomly choose a color for the cell and the nucleus and set the color for the cell with this
define the rotation with Math.random() * 360
define the size with 1 + (Math.random() * 2)
Create a new Vector (0, 0) for velocity and call Method random of Vector with (10, 100)

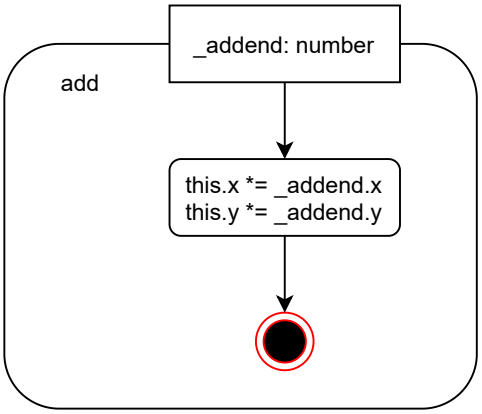

crc2.save()
startAngle = (Math.Pi / 180)
endAngle = (Math.PI(180) *360
beginPath()
draw ellipse with _position, this.radius,
this.radius*Math.random() + this.radius, this.rotation,
startAngle and endAngle
closePath()
set stroke Style to color with opacity "88" and fill-Style to color with opacity "33" and apply them to ellipse

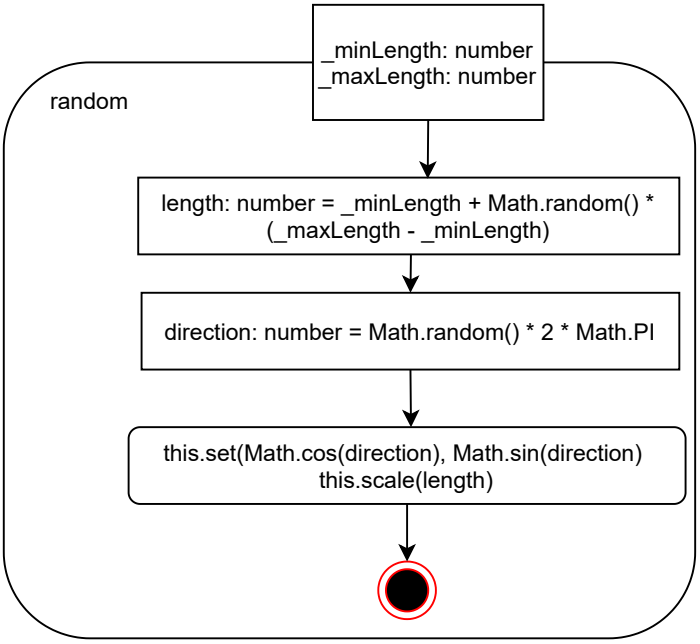

_timeslice: number
move
offset: Vector = (this.velocity.x, this.velocity.y)
offset.x *= 0;
offset.y *= _timeslice * 1.5
add offset to position
[position.x > width]
[position.x < 0]
[position.y < -30]
[position.y > height]
position.y += height
position.y -= height
positon.x += width
position.x -= width


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)