



Reference Documentation

Script Structure

```
function setup(p) {  
  // Called once  
  this.var = 23;  
}  
  
function tick(p) {  
  // Called 60 times per second  
  // p, a Perspective object  
}
```

Perspective

Agent is always at (0, 0)

p.drop(shape, [pos], [size], [color], [alpha], [rotation], [border])	Drop/draw a particle. shape = 'line' 'circle' 'rect' 'triangle' start = new Vector(...) [x, y] size = int new Vector(...) [x, y] color = new Color(...) alpha = 0..1 rotation = 0..360 border = new Color(...)
p.t	Frame number
p.v	Current speed vector
p.remember_pos()	Remember current position
p.last_pos	Last remembered position vector
p.other_agents	A list of Vectors to other agents, sorted by distance. v.ident gives the name of the other agent, in case you wanna know.
p.closest_agent	Closest agent vector
p.closest_particle	Closest other agent particle vector
p.left(d), p.right(d), p.turn(d)	Turn d degrees
p.turnTo(v, f)	Turn a fraction f (0..1) towards a given Vector

p.setSpeed(x)	Set length of speed vector
p.scaleSpeed(f)	Scale length of speed vector
p.adjustSpeed(s)	Add or subtract from speed vector
p.setV(v)	Replace speed vector
p.signals	Signals from L.I.S.A. Signals available: p.signals.dist (5..255) p.signals.sound (0..100) p.signals.btn (0, 100) p.signals.light (0..8, colors)
p.every(f, [name])*	True every f frames
p.hz(h, [name])*	True h times per second
p.periodic(f)	Periodically true and false for f frames

(*) If you don't provide a name for hz(), and every(), it'll use an internal counter, which will be incorrect in complicated if's. In that case, provide a name.

Vector

new Vector(x, y)	Create Carthesian vector
v.x, v.y	Vector components
Vector.fromPolar(r, theta)	Create polar vector (theta in radians)
v.plus(w), v.minus(w)	Add/subtract vector
v.len()	Length
v.resize(len)	Resize vector (keep direction)
v.rotate(a)	Rotate by radians
v.times(f)	Scale by factor f

Color

<code>new Color(r, g, b)</code>	New color object (r, g, b between 0..255)
<code>c.r, c.g, c.b</code>	Color components
<code>Color.grey(a)</code>	Greyscale color (a in 0..255)
<code>palette[i]</code>	Get a predefined color (i in 0..8)
<code>c1.mix(c2, f)</code>	Color mix between c1 and c2 (f in 0..1)
<code>c1.add(c2, f)</code>	Add c2 to c1
<code>c1.sub(c2, f)</code>	Subtract c2 from c1
<code>c.inv()</code>	Invert color
<code>c.scale(f)</code>	Scale r, g and b by factor f

Global Functions

<code>maybe(p)</code>	Return true or false based on a probability p (0..1)
<code>randInt(a, b)</code>	Return random integer from [a..b)
<code>randNr(a, b)</code>	Return random float from [a..b)
<code>pick(arr)</code>	Pick a random element from the array
<code>deg(r)</code>	Radians to degrees
<code>rad(d)</code>	Degrees to radians
<code>clip(x, a, b)</code>	Return x, limited to [a..b]

Custom Signals

You can add custom signals if you want. GET or POST values to to:

`http://<my-ip>:3000/x/signals`

Yes, you can overwrite other people's signals.
Please don't.