

Springbots

API Documentation

November 7, 2008

Contents

Contents	1
1 Package springbots	2
1.1 Modules	2
2 Module springbots.evolveSpringbot	3
2.1 Functions	3
2.2 Variables	3
2.3 Class EvolveSpringbot	3
2.3.1 Methods	3
2.3.2 Properties	4
3 Module springbots.fitness	5
3.1 Functions	5
3.2 Variables	5
4 Module springbots.gear	6
4.1 Variables	6
4.2 Class Node	7
4.2.1 Methods	7
4.2.2 Properties	7
4.3 Class Spring	8
4.3.1 Methods	8
4.3.2 Properties	8
5 Module springbots.latimname	10
5.1 Functions	10
5.2 Variables	10
6 Module springbots.networkEvolveSpringbot	11
6.1 Class NetworkEvolveSpringbot	11
6.1.1 Methods	11
6.1.2 Properties	12
7 Module springbots.networkSpringbot	13
7.1 Class NetworkSpringbot	13
7.1.1 Methods	13
7.1.2 Properties	13

8	Module <code>springbots.springbot</code>	15
8.1	Functions	15
8.2	Variables	15
8.3	Class <code>Springbot</code>	15
	8.3.1 Methods	16
	8.3.2 Properties	18
9	Module <code>springbots.vector</code>	19
9.1	Class <code>Vector</code>	19
	9.1.1 Methods	19
	9.1.2 Properties	21

1 Package springbots

This package contains modules to describe the Springbot creature. Composed with springs and nodes. It implements the 2D physics behind the simulations steps of springbot's structure, the Springbot class itself and functions to load and dump xml files from springbots objects. It also contains modules which extends the Springbot class in order to provide evolving features and networking distribution, also it has a module which implements a genetic algorithm both local and network distributed to evolve Springbot for specific fitness functions implemented in the 'fitness' module like 'walk', 'swim' and 'jump'

1.1 Modules

- **evolvespringbot**: Springbot evolution extension.
(Section 2, p. 3)
- **fitness**: Tests springbot's fitness for specific tasks:
(Section 3, p. 5)
- **gear**: This module contains the core part of the springbots, which is their nodes and springs.
(Section 4, p. 6)
- **latimname**: This module has a function to generate a latim like name, but not latim at all
(Section 5, p. 10)
- **networkevolvespringbot**: This module contains the NetworkEvolveSpringbot class, it inherits both NetworkSpringbot and EvolveSpringbot to create a springbot which is evolvable and network aware
(Section 6, p. 11)
- **networkspringbot**: This module extends the Springbot class to allow its objects to be sent via xmlrpc or other methods over network by translating(two ways) the object itself to a simpler structure based on dictionary and tuples.
(Section 7, p. 13)
- **springbot**: This module implements the springbot, a creature builded with spring and nodes.
(Section 8, p. 15)
- **vector**: This module implements the Vector object.
(Section 9, p. 19)

2 Module `springbots.evolveSpringbot`

Springbot evolution extension. Implements genetic operations like mutation and crossover

2.1 Functions

random.springbot (<i>nodes_num=10, springs_num=30, noderadius=100</i>)

Creates a new springbot totally random
--

2.2 Variables

Name	Description
BLOODLINE_SEP	Separator string between bloodline ids in Springbot's bloodline field Value: ':'

2.3 Class `EvolveSpringbot`



Known Subclasses: `springbots.networkEvolveSpringbot.NetworkEvolveSpringbot`

Extends `springbot` to evolution methods

2.3.1 Methods

__init__ (<i>self, parent=None, name='Unnamed', startangle=0, random=False</i>)
--

Creates a brand new creature or a descendent from a parent

Overrides: `object.__init__` extit(inherited documentation)

generations (<i>self</i>)

Return generations number

addBloodline (<i>self, parent</i>)

Adds a bloodline id

crossover (<i>self</i> , <i>other</i>)

Crosses this springbot with another one, returning a new one
--

mutate (<i>self</i> , <i>newnodedist</i> =100, <i>nodevariation</i> =10)
--

Mutates a random structure of the springbot, which may be adding or removing a node, adding or removing a spring, changing any spring's parameter or changing a node position.
--

Inherited from springbots.springbot.Springbot(Section 8.3)

`__getitem__()`, `__iter__()`, `__len__()`, `__repr__()`, `__setitem__()`, `add()`, `boundingBox()`, `centerGround()`, `colide()`, `colideWall()`, `draw()`, `getNode()`, `loadXML()`, `massCenter()`, `refresh()`, `remove()`, `removeUnconnected()`, `storeXML()`, `unconnected()`

Inherited from object

`__delattr__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`

2.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3 Module *springbots.fitness*

Tests springbot's fitness for specific tasks:

walk: optimizes maximum horizontal difference of mass center before and after simulation

jump: optimizes maximum mass center height achieved compared with initial height equilibrium:

optimizes maximum mass center height percentage over body's height

height: optimizes maximum aspect ratio of body favoring height

swim: optimizes maximum difference of mass center before and after in a liquid environment

3.1 Functions

walk (<i>springbot</i> , <i>width</i> , <i>height</i> , <i>enable_graphics=False</i> , <i>simulation_time=1000</i>)
--

Rewards maximum horizontal difference of mass center position before and after simulation

jump (<i>springbot</i> , <i>width</i> , <i>height</i> , <i>enable_graphics=False</i> , <i>simulation_time=500</i>)

Rewards maximum difference of mass center height achieved

equilibrium (<i>springbot</i> , <i>width</i> , <i>height</i> , <i>enable_graphics=False</i> , <i>simulation_time=600</i>)
--

Rewards maximum average mass center height percentage over body's height
--

height (<i>springbot</i> , <i>width</i> , <i>height</i> , <i>enable_graphics=False</i> , <i>simulation_time=400</i>)

Rewards maximum average aspect ratio of body's height over width
--

swim (<i>springbot</i> , <i>width</i> , <i>height</i> , <i>enable_graphics=False</i> , <i>simulation_time=1500</i>)
--

Rewards maximum difference between mass centers positions before and after the simulation in a liquid environment without gravity.
--

3.2 Variables

Name	Description
HAS_PYGAME	Value: True

4 Module `springbots.gear`

This module contains the core part of the sprinbots, which is their nodes and springs. Here are implemented all the physics behaviour and interations among those parts.

4.1 Variables

Name	Description
ANGLE_STEP	The rate which the angle changes, this parameter affects the frequency of motion of springs Value: 0.025
GRAVITY	Acceleration vector aplied every step of simulation on nodes Value: (0, 0.3)
AIR_RESISTANCE	% of node's velocity lost every step of simulation Value: 0.01
ELASTICITY	Parameter which determines the force of springs Value: 0.6
VISCOSITY	Causes buoyancy force on springs motion Value: 0.05
NODE_WEIGHT	This parameter is not used on physics Value: 1.26
RADIUS	Node's radius Value: 8
MAX_NORMAL	Maximum spring's normal Value: 400.0
UP	Direction constant: UP Value: 1
DOWN	Direction constant: DOWN Value: 2
LEFT	Direction constant: LEFT Value: 3
RIGHT	Direction constant: RIGHT Value: 4

4.2 Class Node



A node is a mass circle which acts gravity, air resistance. It has a position and a velocity and can be connected by springs

4.2.1 Methods

```
__init__(self, pos=(0, 0), vel=(0, 0), acc=(0, 0))
```

Creates a node with a start position, velocity and acceleration

Overrides: object.__init__

```
refresh(self, atr=0.01, grav=(0, 0.3), elast=0.6)
```

Refreshes node: changes position based on velocity and velocity based on acceleration. Also applies gravity and air resistance

```
colideWall(self, limit, side, atr_normal=0.6, atr_surface=0.5, min_vel=0.99, radius=8)
```

Colides this node with a wall, if possible. Applies surface friction and changes velocity

```
colide(self, other, radius=8)
```

Colides this node with another one, if possible

```
__repr__(self)
```

repr(x)

Overrides: object.__repr__ extit(inherited documentation)

Inherited from object

```
__delattr__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__setattr__(), __str__()
```

4.2.2 Properties

continued on next page

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4.3 Class Spring

object —
 springbots.gear.Spring

Spring: An object that connect nodes, and can only exist if connecting two diferent nodes. A spring has a normal lenght state and if pulled or pushed it applies a force on the nodes it connects to restabilish it's state. Springs can also move like muscles by changing it normal lenght by a sine wave frequency.

4.3.1 Methods

<code>__init__(self, a, b, amplitude=0, offset=3.14159265359, normal=None)</code>
Creates a spring connecting nodes A and B, with an start amplitude, offset and normal lenght.
Overrides: object.__init__

<code>refresh(self, elast=0.5, ang=None, visc=0)</code>
Refreshes a spring by applying forces on surrounding nodes and creating buoyancy forces if there are viscosity on the enviroment.

<code>__repr__(self)</code>
<code>repr(x)</code>
Overrides: object.__repr__ extit(inherited documentation)

Inherited from object

`__delattr__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`

4.3.2 Properties

Name	Description
<i>Inherited from object</i>	<code>--class--</code>

5 Module `springbots.latimname`

This module has a function to generate a latim like name, but not latim at all

5.1 Functions

<code>latimname(sil=5)</code>

Creates a latim like name with N sylabes.

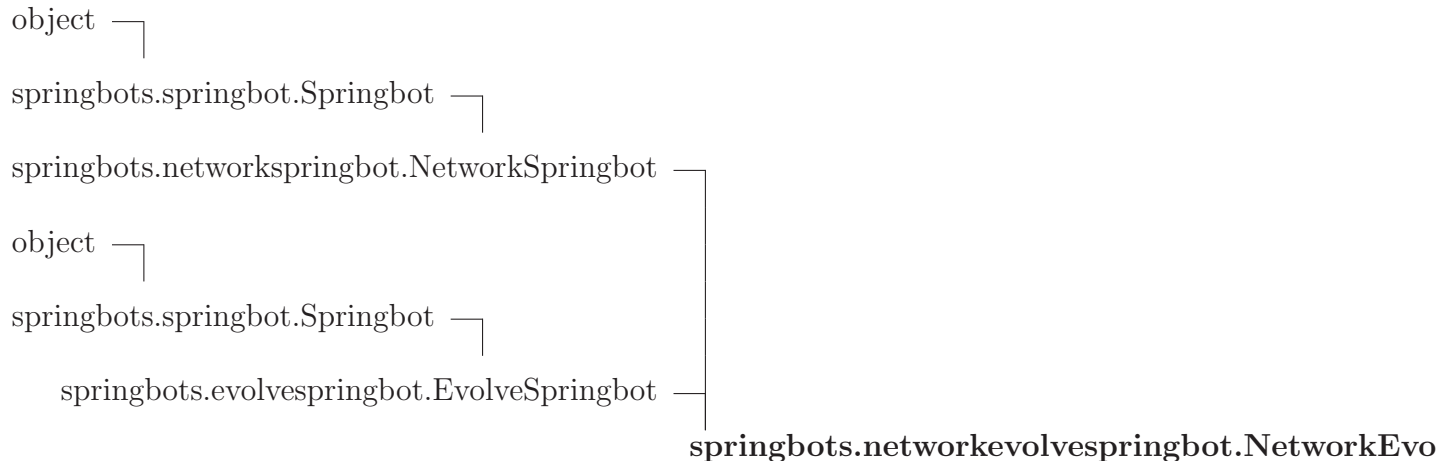
5.2 Variables

Name	Description
VOG	Latim vogals sylabes Value: ['a', 'e', 'i', 'o', 'u', 'ae', 'ia', 'au', 'io', 'oi']
CONS	Latim consonants sylabes Value: ['b', 'c', 'd', 'e', 'f', 'g', 'h', 'j', 'l', 'm', 'p', '...']
MIDCONS	Latim consonants sylabes no to appear first Value: ['mm', 'pp', 'cc', 'tt', 'mn', 'rs', 'll']
TERM	Latim optional terminations for words Value: ['um', 'em']

6 Module `springbots.networkevolvespringbot`

This module contains the `NetworkEvolveSpringbot` class, it inherits both `NetworkSpringbot` and `EvolveSpringbot` to create a springbot which is evolvable and network aware

6.1 Class `NetworkEvolveSpringbot`



This is a double heritage of a network and an evolvable springbot into a class which does both

6.1.1 Methods

```
__init__(self, *args, **kargs)
```

Creates a brand new creature or a descendent from a parent

Overrides: `object.__init__` extit(inherited documentation)

Inherited from `springbots.networkspringbot.NetworkSpringbot`(Section 7.1)

`marshal()`, `unmarshal()`

Inherited from `springbots.evolvepringbot.EvolveSpringbot`(Section 2.3)

`addBloodline()`, `crossover()`, `generations()`, `mutate()`

Inherited from `springbots.springbot.Springbot`(Section 8.3)

`__getitem__()`, `__iter__()`, `__len__()`, `__repr__()`, `__setitem__()`, `add()`, `boundingBox()`, `centerGround()`, `colide()`, `colideWall()`, `draw()`, `getNode()`, `loadXML()`, `massCenter()`, `refresh()`, `remove()`, `removeUnconnected()`, `storeXML()`, `unconnected()`

Inherited from object

`__delattr__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__setattr__()`, `__str__()`

6.1.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

7 Module `springbots.networkspringbot`

This module extends the Springbot class to allow its objects to be sent via xmlrpc or other methods over network by translating (two ways) the object itself to a simpler structure based on dictionary and tuples.

7.1 Class `NetworkSpringbot`



Known Subclasses: `springbots.networkevolvespringbot.NetworkEvolveSpringbot`

Extends evolvable springbot to add marshal and unmarshal methods to send more easily through network like xmlrpc

7.1.1 Methods

marshal (<i>self</i>)
Transforms the springbot into a dictionary

unmarshal (<i>self</i> , <i>dic</i>)
Reads a dictionary into this springbot

Inherited from `springbots.springbot.Springbot` (Section 8.3)

`__getitem__()`, `__init__()`, `__iter__()`, `__len__()`, `__repr__()`, `__setitem__()`, `add()`, `boundingBox()`, `centerGround()`, `colide()`, `colideWall()`, `draw()`, `getNode()`, `loadXML()`, `massCenter()`, `refresh()`, `remove()`, `removeUnconnected()`, `storeXML()`, `unconnected()`

Inherited from `object`

`__delattr__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`

7.1.2 Properties

Name	Description
<i>Inherited from object</i>	

continued on next page

Name	Description
__class__	

8 Module springbots.springbot

This module implements the springbot, a creature builded with spring and nodes.

8.1 Functions

<code>store_xml(<i>springbots</i>, <i>outfile=sys.stdin</i>)</code>

Writes a set of springbots into a xml

<code>isfloat(<i>s</i>)</code>

Tests if a string is representing a float

<code>load_xml(<i>arg=sys.stdin</i>, <i>limit=None</i>)</code>
--

Reads an xml into a set of springbots

8.2 Variables

Name	Description
HAS_PYGAME	Value: True
AIR_RESISTANCE	Value: 0.01
ANGLE_STEP	Value: 0.025
DOWN	Value: 2
ELASTICITY	Value: 0.6
GRAVITY	Value: (0, 0.3)
LEFT	Value: 3
MAX_NORMAL	Value: 400.0
NODE_WEIGHT	Value: 1.26
RADIUS	Value: 8
RIGHT	Value: 4
UP	Value: 1
VISCOSITY	Value: 0.05

8.3 Class Springbot

object —
 springbots.springbot.Springbot

Known Subclasses: springbots.networkspringbot.NetworkSpringbot, springbots.evolvepringbot.EvolveS

Springbot: Creature builded with nodes, which are masses, connected by springs which act like links between nodes pulling them together. Springs may also behave in a cicle muscle pulling and pushing the nodes it connects giving the creature some movement.

8.3.1 Methods

`__getitem__`(*self*, *key*)

Gets an info inten like a dictionary

`__init__`(*self*, *parent*=None, *name*='Unnamed', *startangle*=0)

Creates a brand new creature or a descendent from a parent

Overrides: `object.__init__`

`__iter__`(*self*)

Iterate between `_info` values

`__len__`(*self*)

Lenght(number of nodes)

`__repr__`(*self*)

`repr(x)`

Overrides: `object.__repr__` `exitit`(inherited documentation)

`__setitem__`(*self*, *key*, *value*)

Gets an info inten like a dictionary

`add`(*self*, *objeto*)

Adds an object: node or spring

`boundingBox`(*self*, *radius*=8)

Calculates its bounding box limits

`centerGround`(*self*, *height*)

Center springbot width and touch ground

colide(*self*, *other*, *radius*=8)

Colides this springbot to another

colideWall(*self*, *limit*, *side*, *atr_normal*=0.6, *atr_surface*=0.5, *min_vel*=0.99, *radius*=8)

Colides this springbot with a straight wall

draw(*self*, *screen*, *ticks*=None, *track_x*=False, *track_y*=False, *backgroundcolor*=(20, 10, 0), *showText*=True, *extrainfo*=None)

Draws Springbot using pygame engine. Use this function to show the springbot

getNode(*self*, *id*)

Get node from id, None if id does not exists

loadXML(*self*, *file*=sys.stdin)

Reads a xml into this springbot

massCenter(*self*)

Calculates the center of mass

refresh(*self*, *atr*=0.01, *grav*=(0, 0.3), *elast*=0.6, *visc*=0, *moving*=True)

Refreshes creature's state

remove(*self*, *objeto*)

Remove an object: node or spring

removeUnconnected(*self*)

Remove all nodes and springs unconnected

storeXML(*self*, *file*=sys.stdin)

Store this springbot into a xml file

unconnected(*self*)

Gets all unconnected nodes

Inherited from object

`--delattr--()`, `--getattrattribute--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,

`__setattr__()`, `__str__()`

8.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

9 Module `springbots.vector`

This module implements the Vector object. Which has several of 2D vector features such dot product, adding, project and angle. It also overloads many common math operator

9.1 Class Vector

object `springbots.vector.Vector`

Vector class: Objects of this class have several of 2D vector implementations such dot product, adding, project and angle. It also overloads many common math operator

9.1.1 Methods

<code>__init__(self, a=0, b=0)</code>

Creates a new vector

Overrides: <code>object.__init__</code>

<code>__getitem__(self, index)</code>

<code>__add__(self, other)</code>

<code>__sub__(self, other)</code>

<code>__mul__(self, other)</code>

<code>__rmul__(self, other)</code>

<code>__div__(self, other)</code>

<code>__neg__(self)</code>

<code>__abs__(self)</code>

`--repr--(self)`

`repr(x)`

Overrides: `object.__repr__` extit(inherited documentation)

`--str--(self)`

`str(x)`

Overrides: `object.__str__` extit(inherited documentation)

`--pow--(self, y, z=None)`

`dot(self, vector)`

Return the dot product of two vectors

`cross(self, vector)`

Return the cross product of two vectors

`length(self)`

Return the vector's length

`perpendicular(self)`

Return the vector that is perpendicular of this

`unit(self)`

Return the vector's unit(length one)

`projection(self, vector)`

Return the projection of this vector on another

`angle(self, vector=None)`

Return the angle created by this vector and another of the vector (1,0) if given None

`angle_in_degrees(self, vector=None)`

Return the angle created by this vector and another of the vector (1,0) if given None. Angle in degrees

Inherited from object

`__delattr__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__setattr__()`

9.1.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

Index

- springbots (*package*), 2
 - springbots.evolvespringbot (*module*), 3–4
 - springbots.evolvespringbot.EvolveSpringbot (*class*), 3–4
 - springbots.evolvespringbot.random_springbot (*function*), 3
 - springbots.fitness (*module*), 5
 - springbots.fitness.equilibrium (*function*), 5
 - springbots.fitness.height (*function*), 5
 - springbots.fitness.jump (*function*), 5
 - springbots.fitness.swim (*function*), 5
 - springbots.fitness.walk (*function*), 5
 - springbots.gear (*module*), 6–9
 - springbots.gear.Node (*class*), 6–8
 - springbots.gear.Spring (*class*), 8–9
 - springbots.latimname (*module*), 10
 - springbots.latimname.latimname (*function*), 10
 - springbots.networkevolvespringbot (*module*), 11–12
 - springbots.networkevolvespringbot.NetworkEvolveSpringbot (*class*), 11–12
 - springbots.networkspringbot (*module*), 13–14
 - springbots.networkspringbot.NetworkSpringbot (*class*), 13–14
 - springbots.springbot (*module*), 15–18
 - springbots.springbot.isfloat (*function*), 15
 - springbots.springbot.load_xml (*function*), 15
 - springbots.springbot.Springbot (*class*), 15–18
 - springbots.springbot.store_xml (*function*), 15
 - springbots.vector (*module*), 19–21
 - springbots.vector.Vector (*class*), 19–21