

# Braitenberg Vehivles

João Rafael (2008111876)      José Ribeiro (2008112181)

5 de Abril de 2011

# Conteúdo

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Breve Libraries</b>	<b>4</b>
2.1	Constructors with parameters . . . . .	4
2.2	Object distance . . . . .	4
2.2.1	Point - Sphere . . . . .	4
2.2.2	Point - Box . . . . .	4
2.3	Activators . . . . .	4
2.4	Sensor rotation and initialization . . . . .	5
2.5	Multibody collision handlers (Proxies, and Real's parents ) . . . . .	5
<b>3</b>	<b>Sensors</b>	<b>7</b>
3.1	Light . . . . .	7
3.2	Distance . . . . .	7
3.3	Proximity . . . . .	7
3.4	Smell . . . . .	7
3.5	Sound . . . . .	7
<b>4</b>	<b>Vehicles</b>	<b>10</b>
4.1	Eight . . . . .	10
4.2	Ellipse . . . . .	10
4.3	Braitenberg 3c . . . . .	10
<b>5</b>	<b>Project</b>	<b>11</b>

# 1 Introduction

## 2 Breve Libraries

Como referenciado na documentação da biblioteca Breve, o código Python fornecido é obtido através da compilação de código Steve. Assim, esta biblioteca está pouco otimizada na medida em que não utiliza todas as potencialidades da linguagem. Por isso decidimos efectuar algumas alterações

### 2.1 Constructors with parameters

### 2.2 Object distance

Para implementar correctamente os sensores, é necessário calcular a distância entre dois objectos. Para o cálculo desta distância as bibliotecas originais apenas teem em conta a distância euclidiana entre os centros. No entanto esta aproximação não é suficiente quando os objectos teem dimensões elevadas.

#### 2.2.1 Point - Sphere

Por definição todos os pontos da superfície esférica estão à mesma distância do centro. Desta forma a solução para o caso das esferas é apenas considerar a distância entre os centros e subtrair o raio da esfera.

#### 2.2.2 Point - Box

Uma solução eficiente para este caso consiste em utilizar o algoritmo de Arvo como descrito em <sup>1</sup>. No entanto este algoritmo necessita que a Box esteja alinhada com os eixos. Como este não é originalmente o caso é necessário transformar as coordenadas do ponto no referencial original  $O$  para o referencial da box  $B$ . Esta transformação é obtida através do produto de matrizes:

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{bmatrix} = \begin{bmatrix} x_x & x_y & x_z \\ y_x & y_y & y_z \\ z_x & z_y & z_z \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} O_x \\ O_y \\ O_z \\ 1 \end{bmatrix} * \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$$

Onde  $x, y, z$  são os versores do referencial  $B$  em relação ao referencial  $O$  e  $O_x, O_y, O_z$  são as coordenadas da origem do referencial  $O$  nas coordenadas do referencial  $B$ .

### 2.3 Activators

O veiculos de Braitenberg como definidos na literatura apenas permitem relacionar um sensor directamente com uma roda. Esta abordagem necessita a replicação de sensores

---

<sup>1</sup>[http://www.gamasutra.com/view/feature/3383/simple\\_intersection\\_tests\\_for\\_games.php?page=4](http://www.gamasutra.com/view/feature/3383/simple_intersection_tests_for_games.php?page=4)

quando se pretende que tenha influência em mais que uma roda. Para evitar esta duplicação introduzimos o conceito de *Activador*

## **2.4 Sensor rotation and initialization**

## **2.5 Multibody collision handlers (Proxies, and Real's parents )**



### 3 Sensors

#### 3.1 Light

#### 3.2 Distance

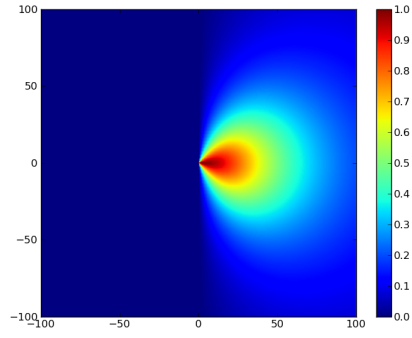


Figura 1: Light:  $bias = 50$   $\alpha = \pi/2$

#### 3.3 Proximity

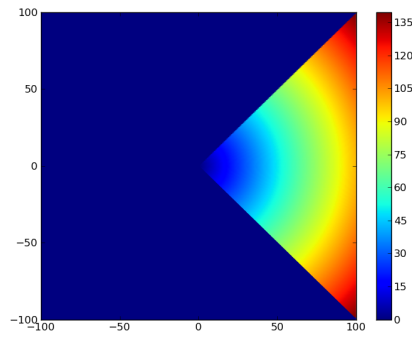


Figura 2: Distance:  $\alpha = \pi/4$

#### 3.4 Smell

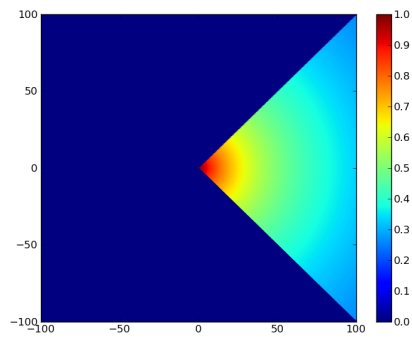


Figura 3: Proximity:  $bias = 50$   $\alpha = \pi/4$

#### 3.5 Sound

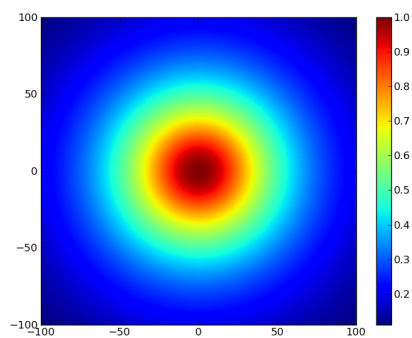


Figura 4: Smell:  $bias = 50$

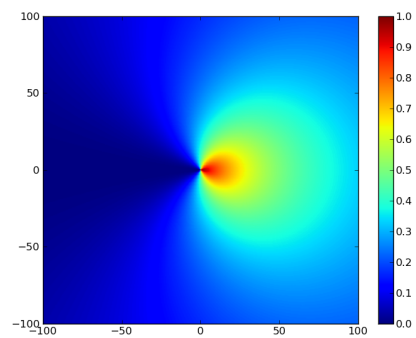


Figura 5: Sound:  $bias = 50$





## 4 Vehicles

### 4.1 Eight

### 4.2 Ellipse

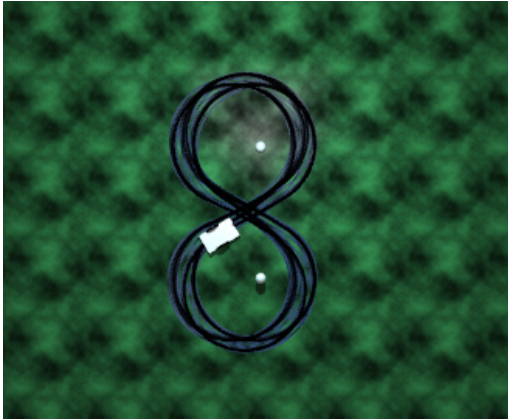


Figura 6: Trail of the eight vehicle

### 4.3 Braitenberg 3c



Figura 7: Trail of the ellipse vehicle

## 5 Project