

Traectorii

Unele obiecte pot avea anumite traectorii, de exemplu: un autobuz se misca liniar, o vrajitoare zboara in cercuri in jurul unui punct etc.

Tipurile de traectorii implementate vor fi:

- **linear** (liniar - deplasare pe un segment). Va fi definita prin doua puncte in spatiu.
- **line_strip** (deplasare pe o linie franta deschisa – adica pe mai multe segmente unite – sunt practic mai multe deplasari liniare unite). Va fi definita prin mai multe puncte in spatiu.
- **line_loop** (linie franta inchisa). Va fi definita prin mai multe puncte in spatiu. Este ca si line_strip, dar obiectul se deplaseaza in plus si pe segmentul care uneste ultimul punct cu primul)
- **circle** (deplasare pe un cerc). Se va defini planul in care se realizeaza deplasarea, prin intermediul a 2 vectori, si a centrului cercului. Se va defini si o raza.
- Eventual alte traectorii definite de voi.

Pentru orice traectorie se va defini in plus si **viteza**.

Miscarea pe traectorie se poate realiza:

- la infinit
- de un numar fix de ori

Se va defini de cate ori se efectueaza miscarea pe traectorie, cu ajutorul atributului **iteration-count** in XML-ul de configurare.

Miscarea poate sa fie mereu in acelasi sens, sau alternata. Se va specifica asta prin atributul **direction**, astfel:

- direction="normal" - miscare intr-o singura directie
- direction="alternate" - miscare alternata (merge intr-un sens, apoi se intoarce)

Pentru diversele tipuri de miscare, avem:

- Pentru miscarea pe linie, in directia normala, obiectul se misca de la A la B, si la urmatoarea iteratie, reapare la A si se misca spre B. In cazul miscarii alternate, pentru o iteratie se va misca A-B-A, rotindu-se cu 180 de grade cand ajunge in B (ca sa se deplaseze cu fata). Rotirea poate fi imediata, dar e de preferat o rotire animata (nu brusca), pe loc, in jurul centurului obiectului, inainte de a porni inapoi spre A.
- Pentru line_strip, in miscarea normala se va deplasa de la A(1) pana la A(k) si va reporni de la A(1). In miscare alternata, va merge A(1)-A(2)....-A(k)-A(k-1)....A(1). Cu aceleasi specificatii despre rotirea in punctul final, ca si mai sus.
- Pentru line_loop si circle, directia alternata nu mai are sens, asa ca va fi ignorat atributul direction.

In XML **in definirea obiectului**, daca vrem sa definim o traiectorie liniara cu un numar de iteratii egal cu 10 si cu miscare alternata, avem:

```
<trajectory type="linear" iteration-count="10" direction="alternate" speed="4">
  <points> <!-- observatie: puteti considera primul punct din traiectorie sa
fie pos-ul obiectului, sau puteti sa nu mai definiti pos pentru obiectele cu
traiectorie, pentru a fi mai clar xml-ul -->
    <point>
      <x>2</x>
      <y>7</y>
      <z>1</z>
    </point>
    <point>
      <x>18</x>
      <y>3</y>
      <z>5</z>
    </point>
  </points>
</trajectory>
```

Pentru line_strip si line_loop am avea mai multe puncte definite in points.

Sa consideram acum un exemplu in care vrem sa definim o traiectorie circulara cu un numar de iteratii infinit, avem:

```
<trajectory type="circle" iteration-count="infinite" speed="4">
  <center> <!-- observatie: puteti considera centrul sa fie pos-ul obiectului,
sau puteti sa nu mai definiti pos pentru obiectele cu traiectorie, pentru a fi mai
clar xml-ul -->
    <x>12</x>
    <y>3</y>
    <z>0</z>
  </center>
  <radius>10</radius>
  <rotationPlane>
    <vect>
      <x>2</x>
      <y>4</y>
      <z>5</z>
    </vect>
    <vect>
      <x>1</x>
      <y>7</y>
      <z>3</z>
    </vect>
  </rotationPlane>
</trajectory>
```

Planul de rotatie alternativ poate fi definit prin unghiurile cu care e rotit fata de un plan de referinta (sa presupunem XOZ).

```
<rotationPlane>
  <!-- elementele x,y,z cuprind aici unghiuri -->
  <x>0.5</x>
  <y>0</y>
  <z>3</z>
</rotationPlane>
```