

Prácticas Movimiento avanzado

Práctica 1

Implementar un Path following. Los puntos del path se especifican en un fichero path.txt junto con el fichero params.txt.

Formato path.xml:

```
<root>
  <points>
    <point x="-100" y="-100"></point>
    <point x="0" y="0"></point>
    <point x="100" y="0"></point>
    <point x="100" y="100"></point>
    <point x="100" y="200"></point>
    <point x="400" y="300"></point>
  </points>
</root>
```

Formato params.xml:

```
<root>
  <params>
    <max_velocity value="100"></max_velocity>
    <max_acceleration value="180"></max_acceleration>
    <arrive_radius value="60"></arrive_radius>
    <targetPosition x="400" y="300"></targetPosition>

    <max_angular_velocity value="10"></max_angular_velocity>
    <max_angular_acceleration value="250"></max_angular_acceleration>
    <angular_arrive_radius value="20"></angular_arrive_radius>
    <targetRotation value="90"></targetRotation>

    <look_ahead value="40"></look_ahead>
    <time_ahead value="4"></time_ahead>
  </params>
</root>
```

Look_ahead es la distancia en el path que hay que sumar a la posición más cercana en el path.

time_ahead es el tiempo que se usa para calcular la posición futura. Sólo se usa en caso de implementar el algoritmo de path following predictivo (opción 2).

Se debe pintar el path en la función DebugDraw.

Consejo: Pintad también la posición a la que voy a hacer seek, así como la posición más cercana y la posición predicha si es el caso.

Práctica 2

Implementar un Obstacle avoidance. Los obstáculos se especifican en un fichero obstacles.txt con el fichero params.txt.

Formato obstacles.txt:

```
<root>
  <obstacles>
    <obstacle x="-100" y="-100" r="30"></obstacle>
    <obstacle x="100" y="-200" r="12"></obstacle>
  </obstacles>
</root>
```

Formato params.txt:

```
<root>
  <params>
    <max_velocity value="100"></max_velocity>
    <max_acceleration value="180"></max_acceleration>
    <arrive_radius value="60"></arrive_radius>
    <targetPosition x="400" y="300"></targetPosition>

    <max_angular_velocity value="10"></max_angular_velocity>
    <max_angular_acceleration value="250"></max_angular_acceleration>
    <angular_arrive_radius value="20"></angular_arrive_radius>
    <targetRotation value="90"></targetRotation>

    <char_radius value="4"></char_radius>
  </params>
</root>
```

Se deben pintar los obstáculos como círculos en blanco en la función DebugDraw. Podéis utilizar la función SetCircle para ello. Podéis crear tantos Sprite círculos en el level como obstáculos tengáis para simplificar y no tener que crear Actores desde el código.

También debéis pintar los obstáculos con los que está colisionando en rojo.

Consejo: pintad también al menos la velocidad actual y la aceleración que se está aplicando.

Configurad una velocidad inicial en BeginPlay para poder probar este steering de forma aislada de los demás

Sólo tenéis que aplicar el steering de obstacle avoidance. En esta práctica no es necesario tener ningún otro para mirar a donde se mueve, ni para hacer seek. Simplemente se mueve según la velocidad actual salvo que encuentre obstáculos.

Práctica 3

Unid las dos prácticas anteriores para tener un personaje que sigue un path evitando obstáculos que encuentre a su paso.