

PROGRAMACIÓN I

PROYECTO

El pescador y el pez

Érase una vez un pescador que llevaba semanas sin pescar nada. Cuando recogía las redes, se dio cuenta que había un pez tan pequeño que el pescador no sabía si echarlo al mar otra vez:

- *Si me comes ahora no podré calmar tu apetito. Pero si me devuelves al mar, podré crecer* - dijo el pequeño pez.

En esto oyó las voces de otros peces que buscaban al pequeño. Era su familia, que le llamaba angustiada.

Tomó en sus manos al pequeño pez, y lo arrojó al mar. Ahora el pequeño pez debía seguir viviendo junto a los suyos y su amada familia que lo buscaba. De esta forma, los otros peces quisieron mostrar su agradecimiento al joven pescador.

Y le llenaron las redes de succulentas ostras, almejas y pulpos. El pescador se alegró mucho de ver reunida a la familia del pequeño pez, y los despidió deseándoles una larga vida a ellos y toda su familia.

El modelo básico del comportamiento de un cardumen de peces

El pescador, al tener esta conexión con el pequeño pez, quedó maravillado con el comportamiento de los cardúmenes y comenzó a desarrollar un estudio junto a ingenieros de la Universidad Andrés Bello, para ello, establecieron las siguientes condiciones:

- Se simulará un mundo en dos dimensiones ($2D$).
- El movimiento de cada pez se describe utilizando dos componentes: La posición y la velocidad.
- La interacción entre peces se basará será descrita a través de 3 reglas.
- No se considerarán alteraciones al medio, no se evaluarán las condiciones del clima, la existencia de depredadores, ni periodos de apareamiento.

Boids de Reynolds

El método estudia el comportamiento emergente de un conjunto de animales a partir de tres reglas sencillas. Fue ideado en 1986 por Craig Reynolds, quien consideraba su algoritmo como parte de los sistemas de partículas. Un sistema de partículas es un conjunto de un gran número de partículas, en el cual cada una de ellas tiene su propio comportamiento; Reynolds diseñó un sistema de partículas que interactúan entre sí, en donde cada partícula en lugar de ser solamente puntos o figuras simples, son objetos que tienen una forma geométrica, comportamiento y orientación. Estas partículas fueron llamadas *boids*.

Reynolds desarrolló tres reglas sencillas para explicar el comportamiento de los boids:

Cohesión Esta regla se refiere a que los miembros del grupo deben intentar mantenerse lo más cerca del centro posible. Esto se logra manteniéndose cerca de sus vecinos más próximos. El pseudocódigo se encuentra en el algoritmo 1.

Algoritmo 1: Pseudocódigo para la regla de cohesión

Input : Individuo P
Lista CARDUMEN

```

1  $x \leftarrow 0$ ;
2  $y \leftarrow 0$ ;
3  $VECINDAD \leftarrow obtenerVecindad(P, CARDUMEN)$ ;
4 para todo  $v$  en  $VECINDAD$  hacer
5    $x \leftarrow P.x - v.x$ ;
6    $y \leftarrow P.y - v.y$ ;
7  $x \leftarrow x/largoVecindad$ ;
8  $y \leftarrow y/largoVecindad$ ;
9  $P.velocidadX \leftarrow P.velocidadX/largoVecindad$ ;
10  $P.velocidadY \leftarrow P.velocidadY/largoVecindad$ ;
11 devolver  $P$ ;
```

Alineación Esta regla mantiene a los miembros del grupo en la misma dirección que el grupo. El pseudocódigo se encuentra en el algoritmo 2.

Algoritmo 2: Pseudocódigo para la regla de alineación

Input : Individuo P
Lista CARDUMEN

```

1  $x \leftarrow 0$ ;
2  $y \leftarrow 0$ ;
3  $VECINDAD \leftarrow obtenerVecindad(P, CARDUMEN)$ ;
4 para todo  $v$  en  $VECINDAD$  hacer
5    $x \leftarrow P.velocidadX$ ;
6    $y \leftarrow P.velocidadY$ ;
7  $x \leftarrow x/largoVecindad$ ;
8  $y \leftarrow y/largoVecindad$ ;
9 devolver  $P$ ;
```

Separación Ser refiere a evitar chocar con los otros miembros del grupo y se logra conociendo la distancia que hay entre ellos. El pseudocódigo se encuentra en el algoritmo 3.

Algoritmo 3: Pseudocódigo para la regla de separación

Input : Individuo P
Lista CARDUMEN
entero minDist

```

1  $x \leftarrow 0$ ;
2  $y \leftarrow 0$ ;
3 numeroCercanos  $\leftarrow 0$ ;
4  $VECINDAD \leftarrow obtenerVecindad(P, CARDUMEN)$ ;
5 para todo  $v$  en vecindad hacer
6   si  $distancia(P, v) < minDist$  entonces
7     numeroCercanos  $\leftarrow$  numeroCercanos + 1;
8      $diff.x \leftarrow P.x - v.x$ ;
9      $diff.y \leftarrow P.y - v.y$ ;
10    si  $diff.x \leq 0$  entonces
11       $diff.x \leftarrow \sqrt{minDist} - diff.x$ ;
12    en otro caso
13       $diff.x \leftarrow -\sqrt{minDist} - diff.x$ ;
14    si  $diff.y \leq 0$  entonces
15       $diff.y \leftarrow \sqrt{minDist} - diff.y$ ;
16    en otro caso
17       $diff.y \leftarrow -\sqrt{minDist} - diff.y$ ;
18     $x \leftarrow x + diff.x$ ;
19     $y \leftarrow y + diff.y$ ;
20 si no hay peces cercanos entonces
21   devolver  $P$ 
22  $P.velocidadX \leftarrow -P.velocidadX / largoVecindad$ ;
23  $P.velocidadY \leftarrow -P.velocidadY / largoVecindad$ ;
24 devolver  $P$ 

```

Cada uno de los elementos del grupo se actualizan en forma individual y respetando estar dentro de los márgenes establecidos luego de aplicar las reglas se debe establecer la nueva posición. El algoritmo que permite obtener la nueva posición de cada individuo se describe en el algoritmo 4, para asegurarse de que los individuos se encuentren dentro de los márgenes establecidos es preciso corregir su posición.

Algoritmo 4: Pseudocódigo para recorrer una lista

Input : Individuo P

```

1
2 si  $P.velocidadX > velocidadMaxima$  y  $P.velocidadY > velocidadMaxima$  entonces
3    $escala \leftarrow velocidadMaxima / \max(P.velocidadX, P.velocidadY)$ ;
4    $P.velocidadX = P.velocidadX + escala$ ;
5    $P.velocidadY = P.velocidadY + escala$ ;
6  $P.x = P.x + P.velocidadX$ ;
7  $P.y = P.y + P.velocidadY$ ;
8 devolver  $P$ 

```

El método busca explicar el comportamiento del grupo. Aunque a simple vista pareciera un movimiento caótico, con el algoritmo de Boids se demuestra que no lo es del todo.

El algoritmo se extiende para permitir a los Boids los siguiente comportamientos:

- Evitar obstáculos: Al estar en movimiento debe ser capaz de evitar chocar contra los objetos que encuentre en su camino. De la misma forma en que evitan chocar con un compañero, los Boids pueden medir la distancia entre ellos y los obstáculos y de esta forma evitarlos.

- Huir de un cazador: Este comportamiento se realiza dándole la capacidad a los Boids de distinguir entre sus compañeros y el cazador y además cuidar la distancia entre ellos y el cazador. Así mismo, el cazador debe intentar acercarse lo más posible a los Boids.
- Perseguir una presa: Al igual que el comportamiento anterior, los Boids deben ser capaces de distinguir entre sus compañeros y una presa y de esa forma intentar que la distancia entre ellos sea mínima mientras la presa realiza lo contrario.

Actividad

Deberá implementar las funciones que se listan a continuación:

1. `vecindad` : Permite obtener la vecindad del individuo.
2. `moveCloser` : Corresponde al algoritmo de cohesión.
3. `MoveWith` : Corresponde al algoritmo de alineamiento.
4. `moveAway` : Corresponde al algoritmo de separación.
5. `corregir` : Corrige la posición del individuo para mantenerlo dentro de los márgenes.
6. `move` : Obtiene la nueva posición del individuo.

Condiciones generales

- Debe utilizar `Python 3.x` y el módulo `pygame` para desarrollar el proyecto. Deberá investigar sobre el módulo `pygame` y su instalación.
- El trabajo se podrá desarrollar en grupos de hasta tres personas.
- Debe modularizar su proyecto. La entrega de un solo script conlleva a la nota mínima.
- Copia entre grupos conlleva a la nota mínima.
- En el laboratorio, cada profesor interrogará a los alumnos respecto al código con el fin de verificar que el programa fue desarrollado por el grupo sin intervención de terceros. El test de salida equivale al 25 % de la nota del proyecto.
- El trabajo será entregado mediante un archivo comprimido, en formato `zip` o `tar.gz`. El nombre del archivo comprimido se construirá a partir de los apellidos, en minúscula, de los integrantes del grupo separados por un guión medio.

Ejemplo: si los integrantes son Tyrion Lannister , John Smith y Camilo Sesto el archivo tendrá por nombre `lannister-smith-sesto.zip` o `lannister-smith-sesto.tar.gz`.