ROBOCODE

PRÁCTICA 3

LUIS ENRIQUE ZAMUDIO CERVANTES

No. CTA. 307293136

Robocode

"Robocode is a programming game, where the goal is to develop a robot battle tank to battle against other tanks in Java or .NET. The robot battles are running in real-time and on-screen." [1]

Robocode, permite crear una abstracción de un tanque de guerra, lo cual implica tener movimientos hacia adelanta, atrás, girar, disparar, detectar un tanque enemigo, etc, todas estas características se encuentran en la documentación oficial de robocode [2].

Antes de crear nuestro tanque, haremos la distinción entre métodos y acciones de nuestro robot.

Llamaremos "*método*" a la capacidad que tiene el tanque para poder ejecutar acciones, estos métodos se ejecutaran mediante eventos, es decir, cada vez que nuestro tanque choque con pared, vea a un robot enemigo, etc, se ejecutan un conjunto de acciones bien definidas. En mi caso usare 4 métodos:

- run() : método encargado del comportamiento inicial del tanque, si sus acciones se encuentran entre un ciclo while (true) estas acciones se ejecutaran durante el tiempo de vida de nuestro tanque.
- onScannedRobot(ScannedRobotEvent e) : método que se ejecuta cuando nuestro tanque "ve" a otro robot.
- onHitByBullet(HitByBulletEvent e) : si una bala toca a nuestro robot, se ejecutara este método.
- onHitWall(HitWallEvent e) : evento que se ejecuta cuando nuestro robot choca con un muro del campo de batalla.

Las "*acciones*" son las propiedades que tiene nuestro tanque para combatir al enemigo (disparar, girar, ir hacia atrás, etc), y solo se podrá ejecutar una acción a la vez. Usare un conjunto total de 10 acciones:

- turnLeft(grados) : gira el tanque a la izquierda la cantidad de grados (de 0 a 360) que reciba como parámetro.

---

[1] http://robocode.sourceforge.net/

[2] http://robocode.sourceforge.net/docs/robocode/robocode/Robot.html

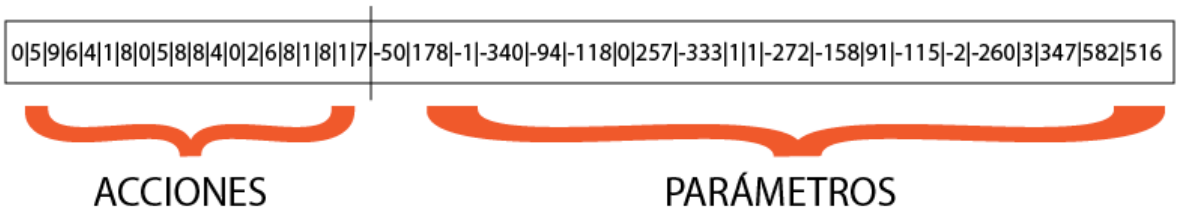
- turnRight(grados) : gira el tanque a la derecha la cantidad de grados (de 0 a 360) que reciba como parámetro.
- turnRadarRight(grados) : gira el radar del tanque a la derecha la cantidad de grados (de 0 a 360) que reciba como parámetro.
- turnRadarLeft(grados) : gira el radar del tanque a la izquierda la cantidad de grados (de 0 a 360) que reciba como parámetro.
- turnGunLeft(grados) : gira el arma del tanque a la izquierda la cantidad de grados (de 0 a 360) que reciba como parámetro.
- turnGunRight(grados) : gira el arma del tanque a la derecha la cantidad de grados (de 0 a 360) que reciba como parámetro.
- ahead(distancia) : camina hacia delante la distancia que reciba como parámetro.
- back(distancia) : camina hacia atrás la distancia que reciba como parámetro.
- fire(power) : dispara una bala con potencia igual a power.
- doNothing() : el tanque no genera ninguna acción.

ALGORITMO GENÉTICO

CODIFICACIÓN

Cada uno de nuestros individuos (tanques de robocode) esta codificado como una cadena de números enteros, donde la primera mitad contiene las acciones a ejecutarse en cada uno de nuestros 4 métodos y en la segunda mitad encontraremos los parámetros que necesita cada acción, en el caso de que tengamos una la acción doNothing(), ya que no recibe ningún parámetro, tendremos el número -1 como constante.

Cada método ejecutara un conjunto de 5 acciones por lo que nuestra cadena tiene una longitud de 40 posiciones.



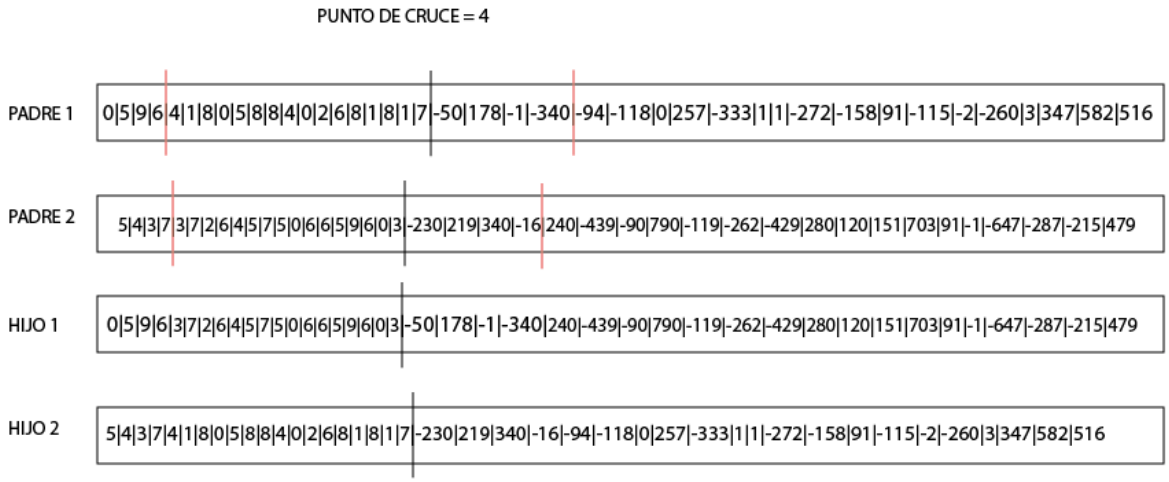
SELECCIÓN

Cada población consta de 100 individuos, el método de selección es el método de Vasconcelos, donde se toma al mejor y al peor individuo, luego al segundo mejor y al penúltimo peor individuo y así sucesivamente.

CRUZA

La probabilidad de cruza es del 90% además se usa elitismo, es decir, el mejor individuo de la generación n pasa directamente a la generación n+1.

Cada pareja tiene 2 hijos, estos se obtienen haciendo cruza en 1 punto, a comparación de la cruza “normal”, la cruza se genera respetando las posiciones de las acciones y de sus parámetros como se ilustra en la siguiente imagen:



MUTACIÓN

La probabilidad de mutación usada es del 1% y es una mutación uniforme, es decir, se recorre la mitad de la cadena, donde se encuentran las acciones, y si esa posición tiene la probabilidad de mutación, se cambia de manera aleatoria por alguna de las 10 acciones al igual que su parámetro respetando el rango que pudiera tener por parámetro esa acción.

La población inicial se genera de manera aleatoria, los parámetros de cada acción están definidos en los siguientes intervalos:

- turnLeft(grados) => [-360 , 360]
- turnRight(grados) => [-360 , 360]
- turnRadarRight(grados) => [-360 , 360]
- turnRadarLeft(grados) => [-360 , 360]
- turnGunLeft(grados) => [-360 , 360]
- turnGunRight(grados) => [-360 , 360]
- ahead(distancia) => [-800, 800]
- back(distancia) => [-800, 800]
- fire(power) => [0, 3]

- doNothing() => -1 (no importa la constante, la acción no recibe parámetros)

FUNCIÓN DE FITNESS

El fitness de cada individuo esta dada por el score que obtiene en cada batalla, para obtener el score se generan batallas con las siguientes características:

- Dimensión del campo de batalla 800 x 600
- 3 rounds
- 5 robots por round, 4 de estos robots son el tanque “Frankenstein” que usa el algoritmo genético y el quinto tanque se genera de manera aleatoria en cada generación eligiendo entre los robots ya definidos en robocode, “Corners”, “Crazy”, “Fire”, “TrackFire” y “Walls”

Todos los resultados de cada generación se guardan en la carpeta “generaciones” con el nombre generacion_[n].txt donde n es el número de la generación.

Hasta ahora se tiene registro de la generación 2800.