Instituto Tecnológico de Buenos Aires

Sistemas de Inteligencia Artificial

Trabajo Práctico 2

Algoritmos Genéticos

29 de agosto de 2023

Índice general

Objetivos

Se desea implementar un motor de algoritmos genéticos para obtener las mejores configuraciones de personajes de un juego de rol.

Descripción del problema

El juego consiste en personajes que tienen cierta clase, ciertas propiedades y cierto equipamiento. El objetivo es lograr la mejor configuracion de ellas para optimizar el desempeño del personaje en el juego. Cualquier supuesto no descripto de este problema no será válido sin antes consultarlo con la cátedra (en un escenario real, sería "el cliente").

Clases

En el juego actualmente existen 4 tipos de personajes: Guerreros, Arqueros, Defensores y Asesinos. Cada personaje debe lograr diferentes objetivos en cuanto a su desempeño en el *ataque* y la *defensa*.

1. **Guerrero**: Este personaje estara en el frente de batalla junto a los defensores. Al estar casi tan expuesto como ellos, deberá tener un desempeño equilibrado en cuanto al ataque.
   Desempeño = 0.6 * *Ataque* + 0.4 * *Defensa*
2. **Arquero**: Este personaje se ubicará en el final del batallón, atacando desde la distancia. Como tal, no precisa de técnicas de defensa, sino que su mayor prioridad es un ataque efectivo.
   Desempeño = 0.9 * *Ataque* + 0.1 * *Defensa*
3. **Defensor**: Este personaje deberá interponerse entre los ataques de sus enemigos y sus colegas. Su función primordial es defender a sus aliados.
   Desempeño = 0.1 * *Ataque* + 0.9 * *Defensa*
4. **Infiltrado**: Este personaje se infiltrará en las tropas enemigas, pasando desapercibido y realizando ataques furtivos desde adentro. Idealmente, no precisará de defenderse, pero deberá estar preparado para ello en el momento en que revelen su identidad para poder huir.
   Desempeño = 0.8 * *Ataque* + 0.3 * *Defensa*

Equipamiento

Las piezas de equipamiento (items) contienen ciertas características (stats): Fuerza, Agilidad, Pericia, Resistencia y Vida.

El personaje al elegir su equipamiento, deberá asignar 150 puntos distribuídos entre todas las características mencionadas.

$$Fuerza_{items} + Agilidad_{items} + Pericia_{items} + Resistencia_{items} + Vida_{items} = 150$$

Atención! Los valores para cada una de las características pueden ser valores decimales.

Estos puntos definirán un coeficiente, redefiniendo las características del personaje en base a las características de sus items, de la siguiente manera:

- $Fuerza_p = 100 * tanh(0{,}01 * Fuerza_{items})$
- $Agilidad_p = tanh(0{,}01 * Agilidad_{items})$
- $Pericia_p = 0{,}6 * tanh(0{,}01 * Pericia_{items})$
- $Resistencia_p = tanh(0{,}01 * Resistencia_{items})$
- $Vida_p = 100 * tanh(0{,}01 * Vida_{items})$

Altura

Aquí vemos modificadores de ataque (ATM) y de defensa (DEM). Estos se definirán en base a la altura (h) de la siguiente manera:

- $ATM = 0{,}5 - (3h - 5)^4 + (3h - 5)^2 + h/2$
- $DEM = 2 + (3h - 5)^4 - (3h - 5)^2 - h/2$

$1{,}3m \leq h \leq 2{,}0m$

Ataque

El ataque y la defensa quedará definido como:

- $Ataque = (Agilidad_p + Pericia_p) * Fuerza_p * ATM$
- $Defensa = (Resistencia_p + Pericia_p) * Vida_p * DEM$

Implementación

El motor de algoritmos genéticos deberá implementar:

- Operadores genéticos
  - Cruce
    - Cruce de un punto
    - Cruce de dos puntos
    - Cruce uniforme
    - Cruce anular
  - Mutación (puede ser GEN o MULTIGEN y al mismo tiempo UNIFORME o NO UNIFORME)
    - **Gen**
      Si por el azar un individuo muta, uno de sus genes muta segun se defina.
    - **MultiGen**
      Cada gen de cada individuo creado tiene la posiblidad de mutar, independiente del resto.
    - **Uniforme**
      Se mantiene la probabilidad de mutación en todas las generaciones
    - **No Uniforme** (opcional)
      La probabilidad de mutación cambia dependiendo de la generación
- Selección y reemplazo*
  - Elite
  - Ruleta
  - Universal
  - Boltzmann
  - Torneos (ambas versiones)
  - Ranking
- Métodos de reemplazo (todos los vistos en clase). Recordar los parámetros de dichos métodos.
- Criterios de corte
  - Máxima cantidad de generaciones
  - Estructura
  - Contenido
  - Entorno a un óptimo (parametrizable)

(*) La selección deberá ser A*(método1) + (1-A)*(método2). El reemplazo deberá ser B*(método3) + (1-B)*(método4). De esta forma, A,B definen porcentajes [0.0-1.0] de selección entre diferentes métodos.

Atención! Dado que son demasiados parámetros, se pide un archivo de configuración externo que lea TODOS los parámetros (no deberá cambiar el código para correr con diferentes configuraciones o diferentes personajes).

Deberán hacer el análisis y estimar la mejor configuración para cada una de las clases.

Entrega

Forma de entrega

Toda la entrega será digital. Deberá contener:

- Código fuente del trabajo
- El documento utilizado para realizar la presentación
- Un README.md o README.txt o simplemente README con una descripción del procedimiento necesario para compilar y ejecutar el programa
- La implementación deberá funcionar sobre un sistema operativo Linux. Explicar los pasos para configurar el entorno, de ser necesario.
- Un archivo de configuración con todos los parámetros del motor de Algoritmos Genéticos

Fecha de entrega

- Lunes 11 de Septiembre entrega digital
- Martes 12 de Septiembre defensa con presentación