



INSTITUTO TECNOLÓGICO DE BUENOS AIRES

SISTEMAS DE INTELIGENCIA ARTIFICIAL

Trabajo Práctico 2

Algoritmos Genéticos

25 de agosto de 2020

Índice general

Objetivos	2
Descripción del problema	2
Clases	2
Equipamiento	2
Altura	3
Ataque y Defensa	3
Datasets	3
Implementación	3
Repositorio	4
Forma de entrega	5
Presentación oral	5
Fecha y horario de entrega	5

Objetivos

Se desea implementar un motor de algoritmos genéticos para obtener las mejores configuraciones de un personaje (altura y equipamiento), dado como “input” la clase del personaje de un juego de rol.

Descripción del problema

Un personaje de un juego de rol se compone de:

1. Su clase (guerrero, arquero, defensor o infiltrado).
2. Su altura física.
3. Su equipamiento (arma, casco, etc...).

El objetivo es lograr la mejor configuración de altura y equipamiento para optimizar el desempeño de un personaje en el juego. Cualquier supuesto no descrito 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 e Infiltrados. Cada personaje debe lograr diferentes objetivos en cuanto a su desempeño en el *ataque* y la *defensa*.

1. **Guerrero:** Este personaje estará en el frente de batalla junto a los defensores. Al estar casi tan expuesto como ellos, deberá tener un desempeño equilibrado.
 $\text{Desempeño} = 0.6 * \text{Ataque} + 0.6 * \text{Defensa}$
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.
 $\text{Desempeño} = 0.9 * \text{Ataque} + 0.1 * \text{Defensa}$
3. **Defensor:** Este personaje deberá interponerse entre los ataques de sus enemigos y su propio batallón. Su función principal es defender a sus aliados.
 $\text{Desempeño} = 0.3 * \text{Ataque} + 0.8 * \text{Defensa}$
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.
 $\text{Desempeño} = 0.8 * \text{Ataque} + 0.3 * \text{Defensa}$

Equipamiento

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

Estos puntos definirán un coeficiente, redefiniendo las características del personaje como:

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

También se encuentran los archivos para cada set de items, en formato **tsv** (Tab-separated Values).

Altura

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

- $ATM = 0,7 - (3h - 5)^4 + (3h - 5)^2 + h/4$
- $DEM = 1,9 + (2,5h - 4,16)^4 - (2,5h - 4,16)^2 - 3h/10$

Cada individuo inicial deberá poseer cierta altura, con una distribución uniforme $h \in [1,3m - 2,0m]$.

Ataque y Defensa

El ataque y la defensa quedará definido como:

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

Datasets

Se disponibilizaran dos datasets, “testdata” (un dataset reducido para que puedan utilizar en etapa de desarrollo del motor) y “fulldata” (el dataset real sobre el que tienen que hacer el análisis).

Implementación

El motor de algoritmos genéticos deberá implementar:

- Operadores genéticos
 - Cruce
 - Cruce de un punto
 - Cruce de dos puntos
 - Cruce anular
 - Cruce uniforme
 - Mutación

- Gen
 - Multigen Limitada
 - Multigen Uniforme
 - Completa
- Selección de padres y reemplazo de individuos*
 - Elite
 - Ruleta
 - Universal
 - Boltzmann
 - Torneos (ambas versiones)
 - Ranking
- Implementación
 - Fill-All
 - Fill-Parent
- Criterios de corte
 - Tiempo
 - Cantidad de generaciones
 - Solución aceptable
 - Estructura
 - Contenido
- Gráficos en tiempo real de, al menos, Fitness Mínimo y Fitness Promedio por cada generación. Idealmente, también graficar la diversidad genética.

(*) La selección de padres deberá ser $a \cdot (\text{método1}) + (1-a) \cdot (\text{método2})$. El reemplazo de individuos deberá ser $b \cdot (\text{método3}) + (1-b) \cdot (\text{método4})$. De esta forma, a,b definen porcentajes [0.0-1.0] de selección entre diferentes métodos. De existir otros métodos híbridos, parametrizarlos de la misma forma.

Atención! Dado que son demasiados parámetros, se pide un archivo de configuración externo que lea TODOS los parámetros. El dataset deberá ser leído de una ruta de archivo parametrizable.

Repositorio

La forma de entrega fué explicada en un anuncio en la materia.

Atención! No subir los archivos de dataset al repositorio.

Forma de entrega

- Crear el tag **TP2** (respetar las mayúsculas) en el repositorio con el trabajo completo. Los contenidos deben contar con:
 - Todo lo mencionado en la sección Descripción del problema.
 - Código fuente del trabajo.
 - Binarios ejecutables (incluyendo librerías, plugins y todo lo necesario para su funcionamiento).
 - Documento de la presentación oral (ppt, pps, pdf, etc.)
 - Un **README** con la descripción detallada de los procedimientos para compilar y ejecutar el sistema; y al menos dos configuraciones de ejecución que se crean oportunas.

Presentación oral

Cada grupo realizará una presentación oral de 20 minutos (como máximo), donde resumirá el trabajo realizado, detallará los resultados obtenidos y explicará las conclusiones a las que llegó. Además, deberán responder las preguntas que los docentes formulen. Los docentes podrán pedir que se hagan corridas en vivo modificando cualquier parámetro del programa. No incluir la descripción del juego en la presentación. **Todos los alumnos del grupo deberán participar significativamente de la presentación.**

Fecha y horario de entrega

Entrega digital: 08 de Septiembre del 2020 a las 08:59hs

Presentación oral: 08 de Septiembre del 2020 a las 09:00hs