



INSTITUTO TECNOLÓGICO DE BUENOS AIRES

SISTEMAS DE INTELIGENCIA ARTIFICIAL

Trabajo Práctico 2

Algoritmos Genéticos

3 de septiembre de 2024

Índice general

Narrativa	2
Resumen	2
Variables Desconocidas	2
Variables a definir en la creación	2
EVE	3
Clases	3
Atributos	3
Altura	4
Ataque y Defensa	4
Enunciado	5
Implementación	6
Forma de entrega	7
Fecha de entrega	7

Narrativa

Para distraernos y descomprimir, un amigo nos aconseja probar un nuevo juego de rol de fantasía medieval: **ITBUM ONLINE**. En este juego, deberemos luchar batallas para defender nuestras tierras de enemigos, y nuestro rol en esta batalla será azarosamente el de un Guerrero, un Arquero, un Guardián o un Mago.

En la creación del personaje, podremos gastar una cantidad de puntos limitada para definir los atributos (fuerza, destreza, inteligencia, vigor y constitución) de nuestro personaje, y adicionalmente podremos definir que tan alto será.

Su amigo les advierte que la creación de personaje será única por usuario, y que la deberán realizar en un breve tiempo limitado también, o el sistema creará el personaje aleatoriamente. Para complicar las cosas, no sabemos exactamente qué clase de personaje (guerrero, etc...) nos otorgará el sistema, no sabemos la cantidad de puntos que podremos asignar (será variable según la suerte que tengamos) y no tenemos conocimiento de cómo distribuir dichos puntos.

Deciden no rendirse al azar, e investigando un poco encuentran en los lugares más recónditos de internet un ‘Entrenador Virtual Experto’ EVE (/i-’vi-i/). No saben cómo opera EVE, pero saben que a partir de cada uno de los atributos, la altura y la clase de personaje, nos arrojará un número de desempeño. Mientras más alto, mejor.

Aún así, en el tiempo limitado y con tantas variables, será muy difícil elegir la mejor combinación en un tiempo acotado! Deciden entonces implementar un motor de algoritmos genéticos utilizando a EVE, con el cuál podrán definir estas variables en un tiempo acotado.

Resumen

Variables Desconocidas

Estas serán las variables (no hiperparámetros) que deberán ingresar al motor de algoritmos genéticos

- Clase de personaje
 - Guerrero
 - Arquero
 - Guardián
 - Mago
- Total de puntos a distribuir (100 - 200)
- Tiempo límite (10” - 120”)

Variables a definir en la creación

- Altura (1.3m - 2.0m)
- Atributos (distribuir total entre...)
 - Fuerza
 - Destreza

- Inteligencia
- Vigor
- Constitución

$$Fuerza_{puntos} + Agilidad_{puntos} + Inteligencia_{puntos} + Vigor_{puntos} + Constitución_{puntos} = Total_{puntos}$$

EVE

La implementación de EVE se basa en las siguientes fórmulas. Deberán implementar pero asumir desconocido al momento de definir los hiperparámetros, ya que EVE en la narrativa funciona como una caja negra.

Clases

En el juego actualmente existen 4 tipos de personajes: Guerreros, Arqueros, Guardianes y Magos. 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. **Guardián:** 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. **Mago:** Este personaje posee mayormente hechizos ofensivos, pero deberá estar preparado para defender y auxiliar a sus tropas en la defensa.
Desempeño = $0.8 * Ataque + 0.3 * Defensa$

Atributos

Los puntos elegidos definirán un coeficiente, redefiniendo los atributos finales del personaje en base a los puntos asignados, de la siguiente manera:

- $Fuerza_{total} = 100 * \tanh(0,01 * Fuerza_{puntos})$
- $Destreza_{total} = \tanh(0,01 * Destreza_{puntos})$
- $Inteligencia_{total} = 0,6 * \tanh(0,01 * Inteligencia_{puntos})$
- $Vigor_{total} = \tanh(0,01 * Vigor_{puntos})$
- $Constitución_{total} = 100 * \tanh(0,01 * Constitución_{puntos})$

Altura

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

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

- $DEM = 2 + (3h - 5)^4 - (3h - 5)^2 - h/2$

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

Ataque y Defensa

El ataque y la defensa quedará definido como:

- $Ataque = (Destreza_{total} + Inteligencia_{total}) * Fuerza_{total} * ATM$

- $Defensa = (Vigor_{total} + Inteligencia_{total}) * Constitución_{total} * DEM$

Enunciado

Ejercicio 1.1

- Cómo definirían los hiperparámetros de su motor de algoritmos genéticos?
 - Muestren evidencia de su investigación / experimentación
 - Elaboren su respuesta, justificando la decisión de sus hiperparámetros

Ejercicio 1.2

Semanas luego de comenzar el juego, quieren ayudar a algunos amigos a que creen sus personajes también. Esta vez, los desarrolladores del juego subieron el tiempo límite a 30 minutos. Quieren que sus amigos puedan crear personajes con buen desempeño, pero que el motor corra la menor cantidad de tiempo.

- Cómo elegirían los hiperparámetros?
 - Definan supuestos sobre el ‘trade-off’ entre esperar y optimizar dentro de estos 30 minutos.

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 según se defina.
 - **MultiGen**
Cada gen de cada individuo creado tiene la posibilidad 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 permitir utilizar uno o dos métodos al menos, en un porcentaje definido, y los mismos podrán ser diferente entre la selección de padres y la selección para el reemplazo al generar una nueva generación.

Por ejemplo, podrían definir porcentajes A (padres) y B(reemplazo) [0.0-1.0], y métodos de selección 1 y 2 (padres) y 3 y 4 (reemplazo), de manera que al seleccionar padres el total sea $A * (\text{método1}) + (1-A) * (\text{método2})$ - y el reemplazo podrá ser $B * (\text{método3}) + (1-B) * (\text{método4})$

Atención! Dado que son demasiados hiperparámetros, se pide un archivo de configuración externo que lea TODOS los hiperparámetros (no deberá cambiar el código para correr con diferentes hiperparámetros, analizar diferentes clases de personaje o diferente tiempo límite).

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 hiperparámetros del motor de Algoritmos Genéticos

Fecha de entrega

- Martes 17 de Septiembre 14:59hs entrega digital
- Martes 17 de Septiembre 15:00hs (según timeslot asignado) defensa con presentación