

The background is a dark blue gradient. In the top-left corner, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the bottom-left corner, there is a circular inset showing a detailed, grayscale image of a circuit board. In the top-right corner, there is a faint, grayscale image of a circuit board with many small components.

# Algoritmos genéticos

Sistemas de Inteligencia Artificial  
Grupo 12



# Introducción

Se describe el desarrollo de un motor de algoritmos genéticos con el fin de obtener las mejores configuraciones de personajes de un juego de rol. El juego consiste en personajes que tienen cierta clase, ciertas propiedades, y cierto equipamiento. El objetivo es lograr la mejor configuración de ellas para optimizar el desempeño del personaje en el juego.

El análisis se realizará sobre el personaje **Defensor 2**



# Implementación del proyecto

El motor se divide en distintos paquetes de Java. Cada paquete incluye una interfaz y un conjunto de clases con implementaciones particulares de cada método

- Selection
- Replacement
- Mutation
- Crossover
- Stop condition

También se implementaron paquetes del modelado del problema

- Item
- Character



# Implementación del proyecto

```
while (!stopCondition.stop())  
  chromosomes = replace.apply(  
    mutate.apply(  
      cross.apply(  
        select.apply(chromosomes)  
      )  
    ), chromosomes);
```



# Implementación del modelo

Se implementó una clase `Chromosome` que referencia al `Character` que se está analizando. Cada cromosoma consiste entonces del personaje, la lista de `Items` y la altura del personaje. De esta forma, cada generación consiste de una lista de cromosomas cada uno con sus respectivas configuraciones.

## Cromosoma

### Personaje

Modificadores de  
desempeño

Lista de Items  
Altura



# Fitness

Se describe el desarrollo de un motor de algoritmos genéticos con el fin de obtener las mejores configuraciones de personajes de un juego de rol. El juego consiste en personajes que tienen cierta clase, ciertas propiedades, y cierto equipamiento. El objetivo es lograr la mejor configuración de ellas para optimizar el desempeño del personaje en el juego.

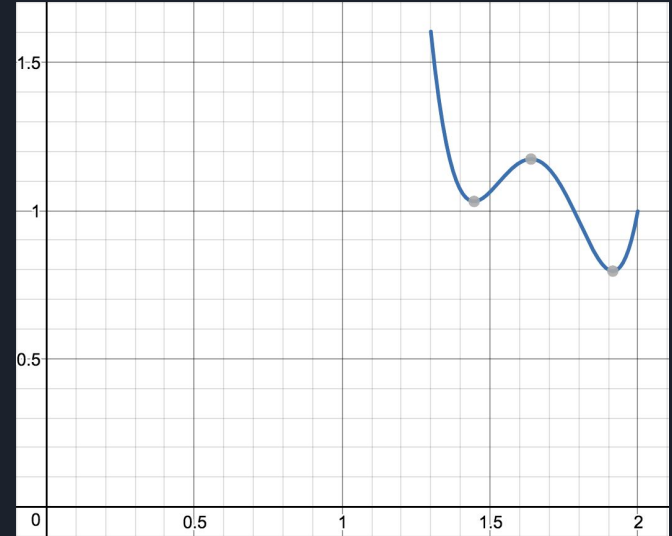
El análisis se realizará sobre el personaje **Defensor 2**

$$\text{Fitness} = 0.1 * \text{Ataque} + 0.9 * \text{Defensa}$$

# Análisis

Una vez definida la forma en la que se *calificará* el desempeño de cada personaje, pensamos comenzar analizando algunos detalles que saltan a la vista. Por ejemplo, podemos graficar el DEM en función de la altura dado a que son funciones de una sola variable, y por lo tanto fáciles de graficar, y al hecho de que son factores directos del fitness.

Dado que la defensa representa el 90% del desempeño de nuestro personaje, esperamos que la altura del personaje esté bastante cercana al máximo del DEM. De esta forma, la *Figura 1* nos indica que la altura óptima de nuestro personaje rondará 1.3 metros.





# Análisis

Para simplificar el análisis se decidió fijar los siguientes parámetros:

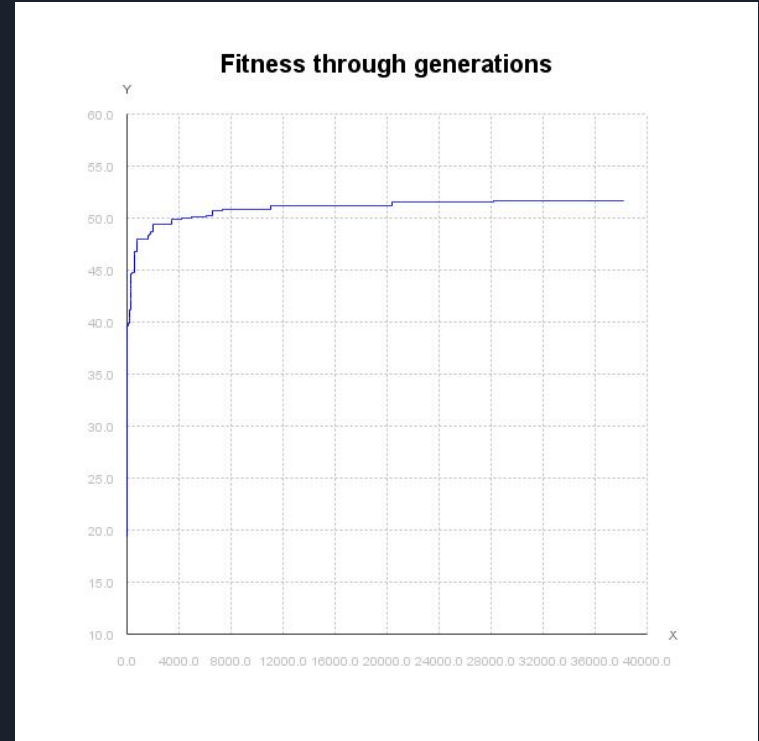
- Método de reemplazo 2
- Población de 200 cromosomas
- Generation gap de 70% (140 seleccionados)
- Condición de corte por contenido, es decir, detener el algoritmo luego de que el mejor desempeño de la generación no mejore por 10.000 generaciones.



# Análisis

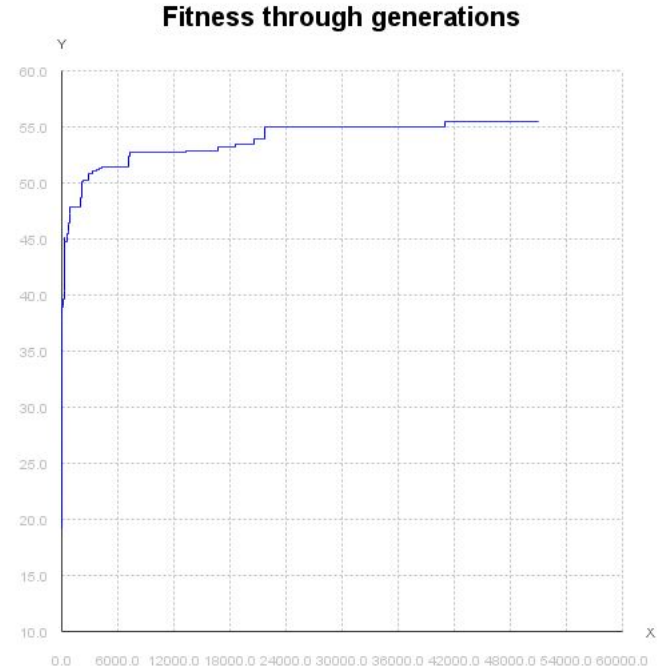
Comenzamos con una cruce en un punto con una probabilidad del 80%, una mutación uniforme con una probabilidad del 1%, y un método de selección de Ruleta. El método de reemplazo cuenta con la misma selección.

La evolución del mejor fitness por cada generación puede verse en la *Figura 3*. Con estos parámetros logramos un fitness de 51.63 luego de 39,000 iteraciones.



# Análisis

Para seguir realizando pruebas decidimos mantener los mismos parámetros y cambiar únicamente el método de selección por uno de Elite. Como puede verse en la *Figura 4* Se alcanzó un fitness de 55.42 luego de 52,000 iteraciones.



# Análisis

Si bien el fitness mejoró, podríamos estar en un caso de convergencia prematura. Es por esto que decidimos utilizar un método de selección mezclada entre Boltzmann (Con una temperatura inicial de 400 grados) y Elite. Manteniendo los mismos parámetros, le otorgamos un peso de 80% a Boltzmann y uno de 20% a Elite para evitar la convergencia prematura. Tras haber realizado los cambios podemos ver que el fitness alcanza 54.70 luego de 50,000 iteraciones (*Figura 5*).

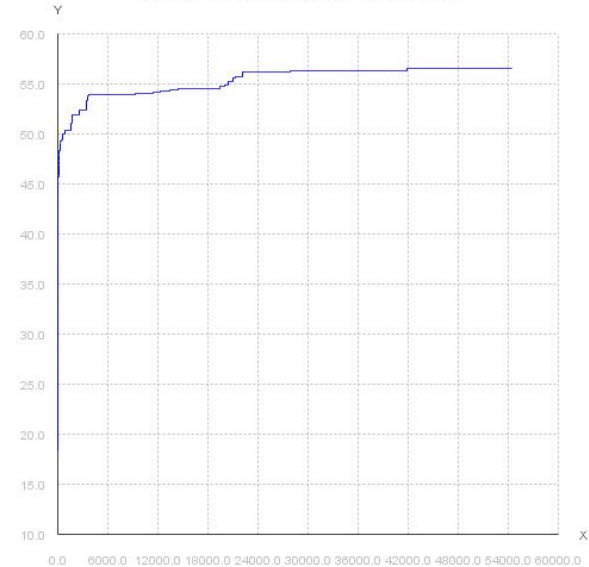


# Análisis

Tomamos entonces una cruce uniforme con una probabilidad del 50% y, una mutación uniforme con una probabilidad del 10%.

Aplicando el algoritmo con los parámetros descritos se obtienen los resultados de la *Figura 6* y un fitness de 57.23 luego de 54,000 iteraciones.

**Fitness through generations**





# Conclusión

Item	Id
Botas	801166
Casco	516677
Armadura	96822
Guantes	382596
Arma	37856
Altura	1.300 metros

Fitness: 57.23



Demo

¡Muchas gracias!