

Análisis y aplicación de técnicas inteligentes a un jugador automático en videojuegos tácticos.

Resumen

Partiendo de la noción de lo que es un videojuego táctico en el marco de la teoría de juegos, siendo este un juego de dos jugadores de suma cero en el que existen unas unidades colocadas en un tablero discreto, pertenecientes a uno de dos bandos enfrentados, en turnos discretos para cada uno de los jugadores, y de la necesidad que hay de que tengan jugadores automáticos se presenta, a continuación, un programa inédito, ubicado en el repositorio github.com/granates/tactical_game, un caso de estudio, el cual es base para probar una serie de técnicas inteligentes en aras de distinguir cuál tiene mejor aplicabilidad. Luego se recaban los datos referentes a los enfrentamientos entre las técnicas inteligentes, es decir, la proporción de victorias de cada método frente a cada uno de los demás, se analizan estadísticamente y se valoran críticamente. Asimismo, las técnicas que se han elegido para test y para probar empíricamente son dos técnicas Minimax (una versión modificada de la poda alfa-beta y también Negascout) y un algoritmo basado en la búsqueda de la casilla/enemigo con mayor daño promedio a un enemigo en este turno y menor a ella en el siguiente. Se concluye que, con un muestreo amplio en todos los casos ($n > 200$), no hay evidencias estadísticamente significativas de que alguna de las técnicas probadas tenga una proporción de victorias mayor a otra con los parámetros utilizados y que, por tanto, haría falta mejorar el rendimiento de estas aún más para poder afirmar o descartar si alguna es superior, amén de testear otras técnicas que no se hayan implementado para tener una visión más amplia.

Resultados

Valores Z del enfrentamiento de la técnica X vs Y (ratios ganadas/jugadas)

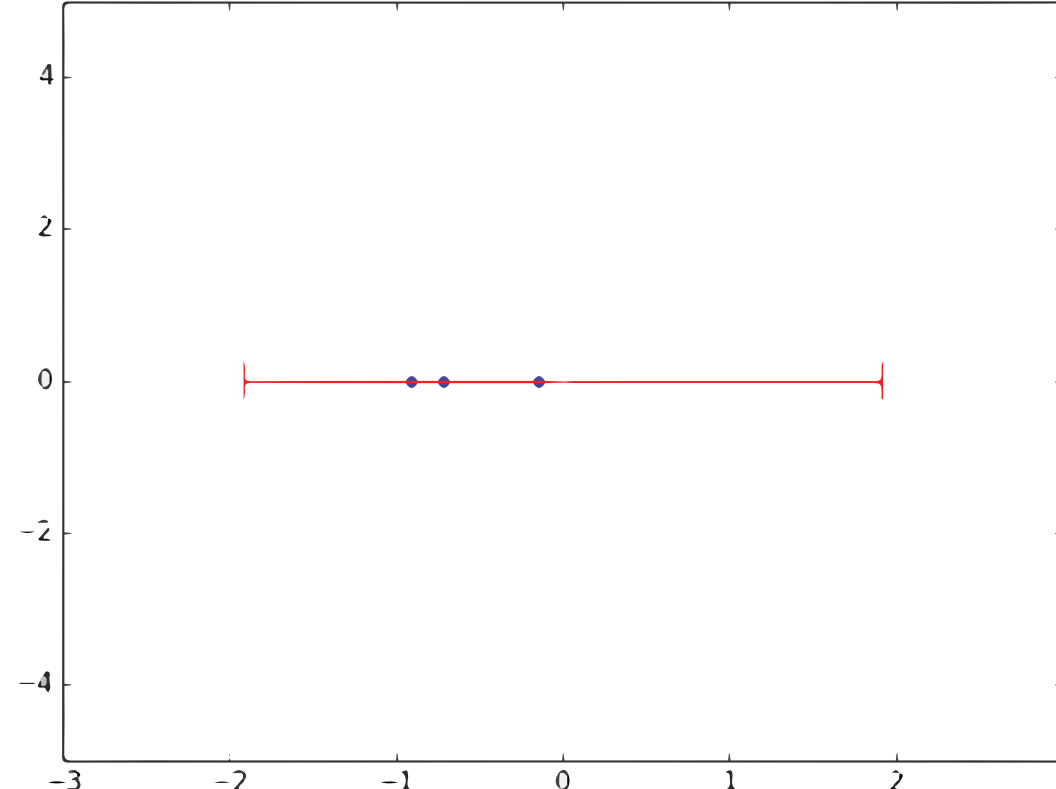


Figura R1: Resultados estadísticos, 1

Valores z del ratio ganar/partida de X vs Y contra Y vs X

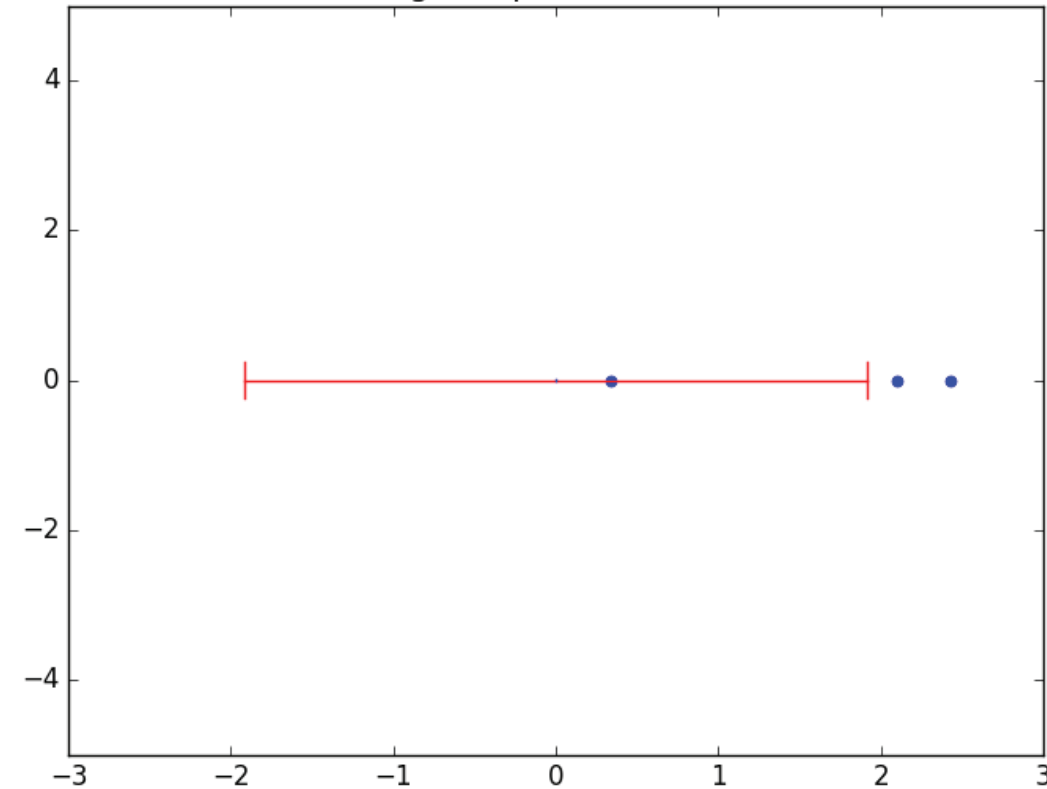


Figura R2: Resultados estadísticos, 2

El **gráfico R1** refleja el valor z del test de proporciones sobre las proporciones de victorias globales de una cierta técnica comparada con otra, para ver si la diferencia entre las dos proporciones es estadísticamente significativa, o lo que es lo mismo, si hay alguna mejor que otra; como se observa, en ningún caso lo ha sido. **Por otra parte, R2** compara la proporción de victorias si una empieza primera con la recíproca, es decir, con la proporción si tiene el segundo turno. Es una diferencia significativa en los casos en que participa el script del daño promedio (vs. Negascout y Alfa-beta, los dos puntos fuera del rango rojo)

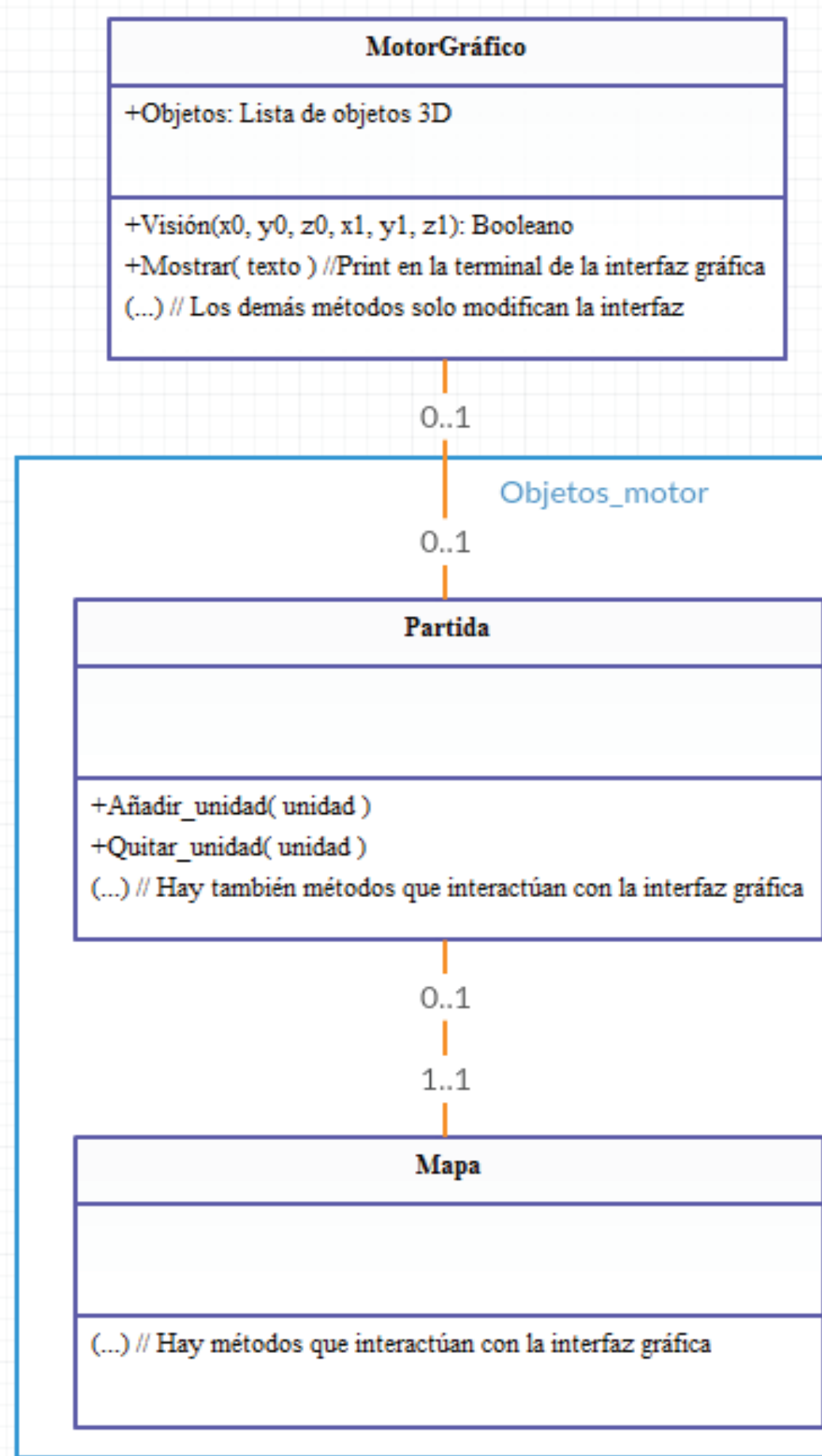
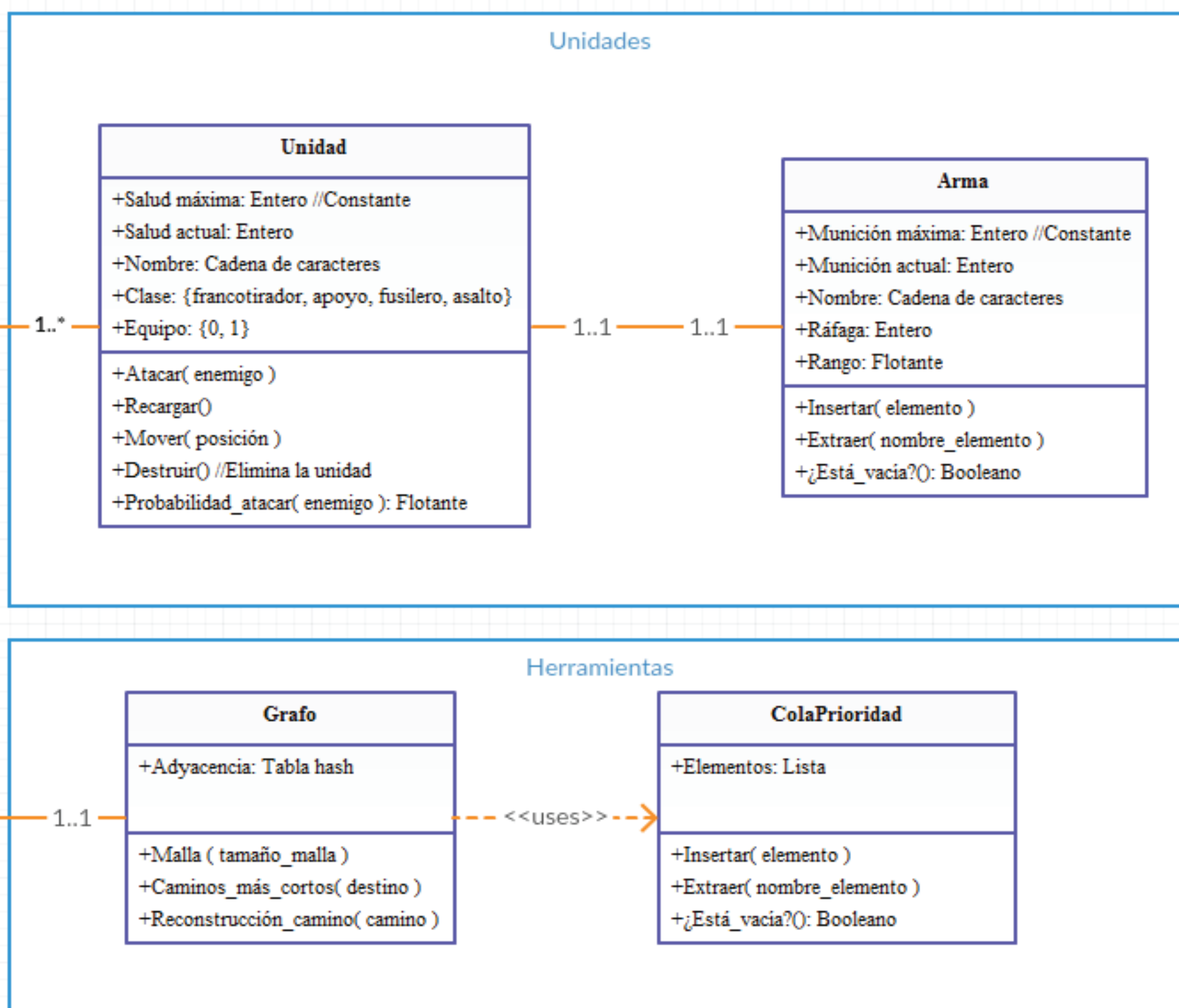


Figura D: Diagrama UML del programa



Métodos

Las técnicas Minimax se han modificado con respecto a las originales: estas analizan solo algunos de los movimientos siguientes, en lugar de todos, mediante un muestreo basado en algunos parámetros sencillos, y suponen que, una vez ejecutado un movimiento, se dará la situación más probable; es decir, que elaboran un plan. Para el caso en que se dé una situación no contemplada (por ejemplo, atacar una unidad ya eliminada), se entra en un estado, llamado “pánico”, donde el siguiente movimiento se calcula con otro algoritmo, preferiblemente uno con menor coste temporal (para este caso se ha elegido el método del daño promedio). La figura M1, por otra parte, muestra un diagrama UML de las clases que interactúan con Partida; tanto el método del daño promedio como las dos técnicas Minimax tienen una dependencia con Partida y, por extensión, el sistema que se ve en la figura. Finalmente, la heurística que las técnicas Minimax utilizan es:

$$h(n) = -(C_{salud} \cdot \sum_{unidades} salud_{enemigos} + C_{muerte} \cdot \sum_{unidades} unidades_{enemigas})$$

Figura M: Heurística de los Minimax de este proyecto

Es decir, que, intuitivamente, lo que falta para llegar al nodo final son las unidades enemigas restantes y su salud, promediadas por unas constantes.

Conclusiones

Los resultados no sugieren una prevalencia de una técnica sobre otra con los parámetros utilizados (anchura = 16, profundidad = 2), tan solo que quién tiene el primer turno es importante en los casos que involucren el algoritmo basado en maximizar el daño promedio (el “script fijo”), no para los dos de tipo Minimax; sería necesario optimizar las técnicas para probarlas con más profundidad y muestreo en anchura para ver si alguna llega a superar significativamente en proporción de victorias a otra.

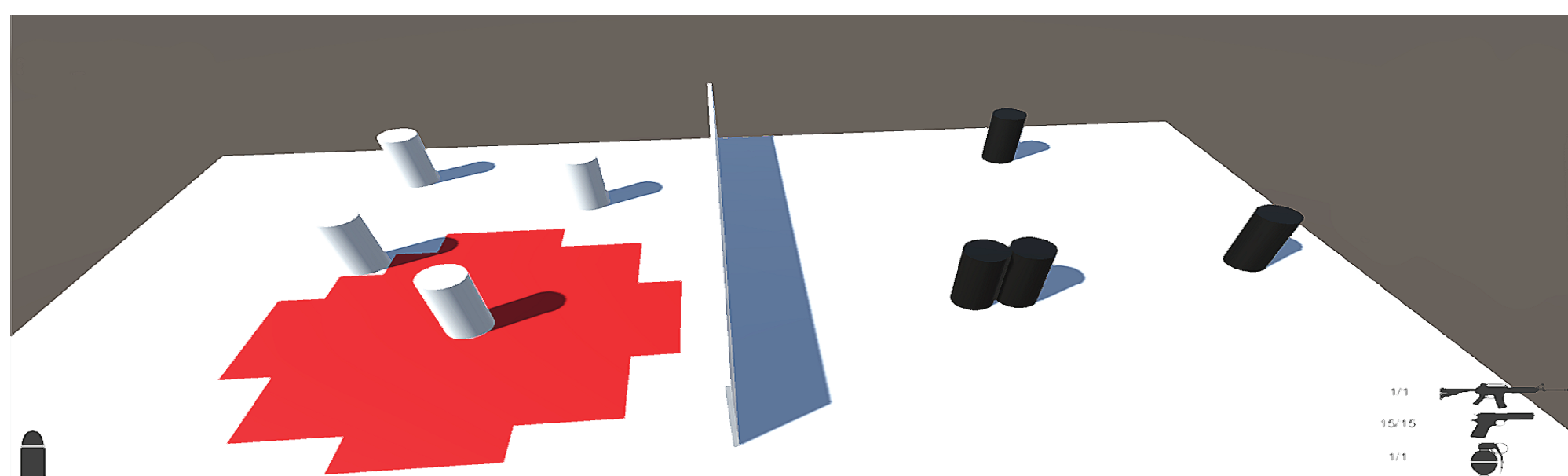


Figura C: Captura de pantalla de la interfaz gráfica.