

Introducción

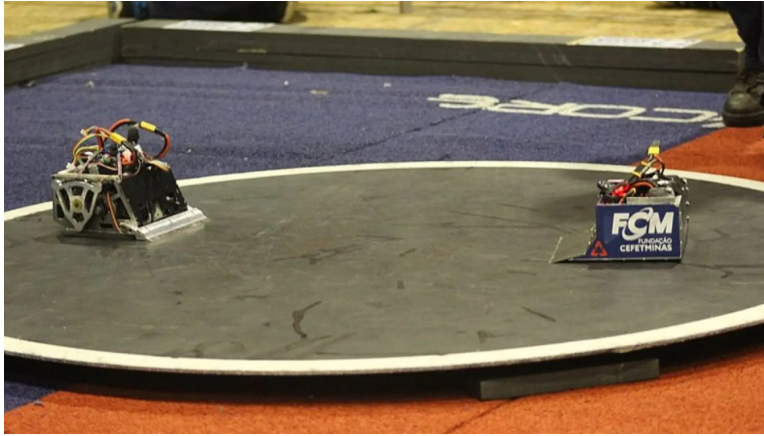


Figura 1: Robot Sumo

El “Robot Sumo” es un deporte real, que incluso se juega a nivel profesional. Análogo al Sumo entre humanos, este consiste en sacar al oponente del campo, solamente empujándolo. A veces se juega con roboceros autónomos, y a veces a control remoto. En este trabajo se presenta una simulación de lo que se podría considerar una generalización de este deporte en su modalidad autónoma. En particular: en lugar de haber solo dos roboceros, hay varios equipos de roboceros que se enfrentan todos contra todos, y la forma del campo puede variar durante la partida, lo que generalmente se traduce en que a partir de cierto momento, el campo comienza a reducirse, hasta que desaparece. De esta simulación se sacarán conclusiones acerca de qué aptitudes físicas y “mentales” (o sea, de la IA) son más importantes y/o mejores para ganar una partida.

Detalles de la simulación

Por simplicidad, la simulación se realiza en 2D. Para simular la física (colisiones, fuerzas, etc.) se utiliza el motor de físicas Box2D. Para la visualización se usa la biblioteca Raylib. Toda acción que realiza un robot (atacar, posicionarse, rotar, etc.) se simula siempre aplicando fuerzas físicas sobre el mismo. Los robots se representan como conjuntos de polígonos, y pueden tener forma arbitraria. Estos poseen además otras propiedades físicas, entre ellas: masa, restitución, fuerza (separada en fuerza lineal y rotacional), precisión, rango de visión y frecuencia de procesamiento (cuantas veces por segundo analiza la situación actual para decidir qué hacer). Hay distintas “personalidades” de robot disponibles, cada una con sus propias preferencias y prioridades a la hora de jugar. Los robots pueden moverse libremente por el campo, pero para hallar su camino de un lugar a otro (esquivando obstáculos)

utiliza una discretización (idealmente fina) del campo, en forma de grafo. Respecto al campo, los parámetros más importantes con los que se puede experimentar son el radio, a los cuántos segundos comienza a reducirse, y a qué velocidad se reduce. El código de los jugadores, su cerebro, y la lógica del campo se pueden encontrar respectivamente en `fighter.py`, `brain.py` y `scene.py`. Se presenta una visualización de la batalla, que se puede desactivar para que la simulación vaya más rápido (útil a la hora de sacar estadísticas). Con la configuración con la que realizamos los experimentos, 30 frames representan un segundo dentro de la simulación(aunque esto es un parámetro configurable). Esto no significa que la simulación vaya a 30 frames por segundo constantes, sino que cada vez que pasan 30 frames, pasa un segundo de la simulación. Así, si el programa va a 210 fps, a cada segundo de la vida real están pasando 7 segundos dentro de la simulación.

La arquitectura de los agentes

Para la inteligencia de los agentes (los jugadores) se decidió optar por una arquitectura BDI. Los “Belief” serían todos los datos que tiene el agente para tomar decisiones, que mayormente corresponde a las propiedades del agente, y de los agentes que están en su campo de visión, además de sus posiciones y velocidades. Los “Desires” son planes relativamente abstractos, como “Atacando al agente X” o “Ejecutando la fase 2 del ataque X”. Los “Intentions” corresponden a qué se debe hacer específicamente para ejecutar esos planes, como por ejemplo “Cargando hacia la posición (x,y)”, o “Rotando en el lugar”, o “Frenando”.

Detalles de la inteligencia de los agentes

Lo primero que hace un agente es inferir qué está haciendo el resto de los agentes dentro de su campo de visión (atacándolo, huyendo de él, etc.). Esto se logra mediante un modelo oculto de markov (con memoria) por cada agente dentro del campo de visión, cuya entrada son los belief del agente. Una vez un jugador infiere los estados del resto, queda tomar decisiones al respecto. Esto se hace de forma heurística. Dentro de nuestra implementación, se puede escoger entre 3 funciones distintas para esto, que son como las “personalidades” de los agentes, ya que cada una de ellas prioriza un aspecto distinto a la hora de jugar. Las opciones son:

-IA estándar: intenta llegar a un balance entre ataque y defensa. Se arriesga, pero no mucho. Su ataque es el estándar (cargar directamente contra el enemigo, rotando).

-IA precavida/paranoica: prioriza la defensa. Trata de mantenerse muy alejada del borde del campo. Si hay muchos jugadores con una probabilidad relativamente alta de estarla atacando, huye, o empieza a rotar descontroladamente para alejarlos. Su ataque es el estándar.

-IA planificadora: Prefiere un ataque más complejo, de dos fases: se aleja para tomar impulso, y luego carga contra aquel al que decidió atacar. Si en la segunda fase pasa mucho tiempo, puede decidir dejar ese plan y pasar a otro. También cuenta con el ataque básico, pero lo elige con menos frecuencia.

Hay dos comportamientos que sí tienen en común las 3 opciones: si están muy cerca del borde del campo (que significa “muy cerca” lo define cada una), priorizar alejarse, y si no hay nadie en su radio de visión, explorar.

Experimentos

A continuación se describen (y dan resultados de) algunos de los experimentos realizados en la simulación. En los experimentos, los agentes usan la IA estándar, a no ser que se diga lo contrario.

Acerca de las distintas IAs de los agentes

¿Cuál de las IAs que proponemos para los agentes es más efectiva? o equivalentemente ¿Qué forma de jugar es más efectiva? Es difícil hacer un ranking exacto de las IAs, ya que su efectividad varía de acuerdo al tamaño del campo, o algunos atributos físicos del robot (precisión o campo de visión por ejemplo). Así que fijemos un campo de batalla estándar (10 unidades de radio, 20 segundos antes de que el campo se empiece a achicar, y el radio del campo de achicca a 2 unidades por segundo), y unos equipos estándar (sus propiedades exactas se encuentran en el fichero de python que se usó para este experimento), en este caso idénticos en todo sentido (excepto que los robots de cada equipo los controla una IA distinta), y veamos los resultados.

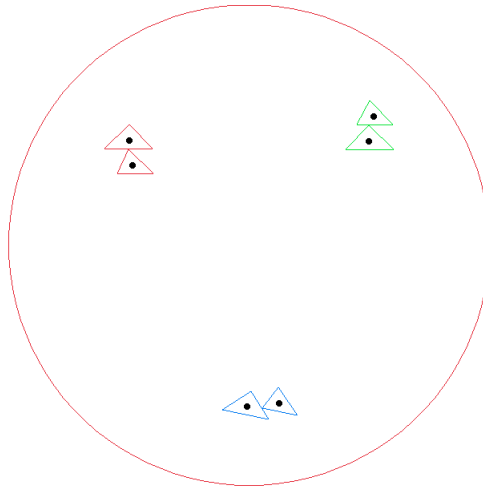


Figura 2: El campo para este experimento. Los equipos Rojo (R), Verde (G) y Azul(B) tienen las IAs estándar, paranoíca, y planificadora respectivamente.

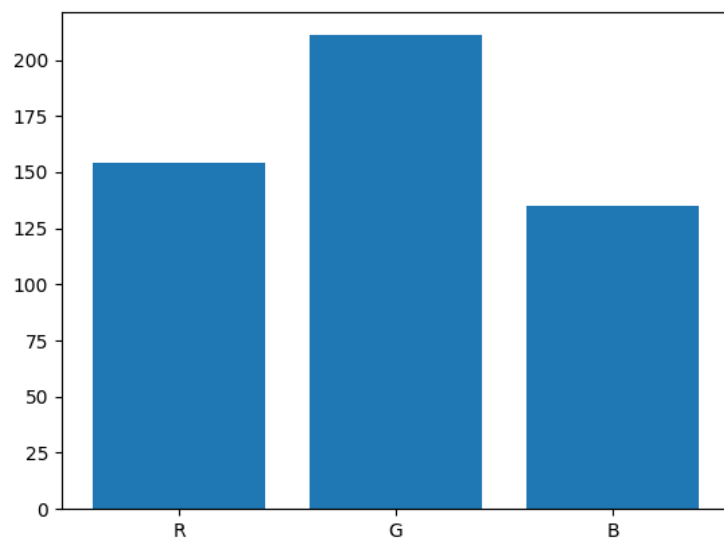


Figura 3: Los resultados del experimento. La IA paranoica (precavida) gana. Las entre las otras dos hay diferencia también, pero no realmente significativa.

La clara ganadora es la IA más precavida. Esto se debe mayormente a que la regla de “el campo comienza a reducirse hasta que desaparece” convierte a este juego en uno de supervivencia, lo que premia las tácticas defensivas y el control del centro del campo. Los resultados para las otras dos son no presentan una diferencia tan marcada entre sí, pero una posible explicación para ella es que en un juego rápido, los movimientos compuestos se demoran demasiado, y resultan penalizados (entonces el ataque de dos fases de la IA planificadora sería sub-óptimo).

Acerca de las formas de los robots

¿Importa la forma de los robots? Volvamos a tomar un campo estandar y equipos con propiedades idénticas (ahora salvo la forma), y veamos:

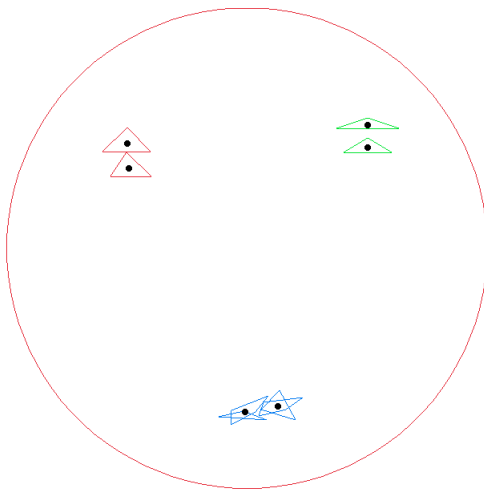


Figura 4: El campo para este experimento. Los equipos Rojo (R), Verde (G) y Azul(B) tienen forma normal, alargada y compuesta/cóncava respectivamente.

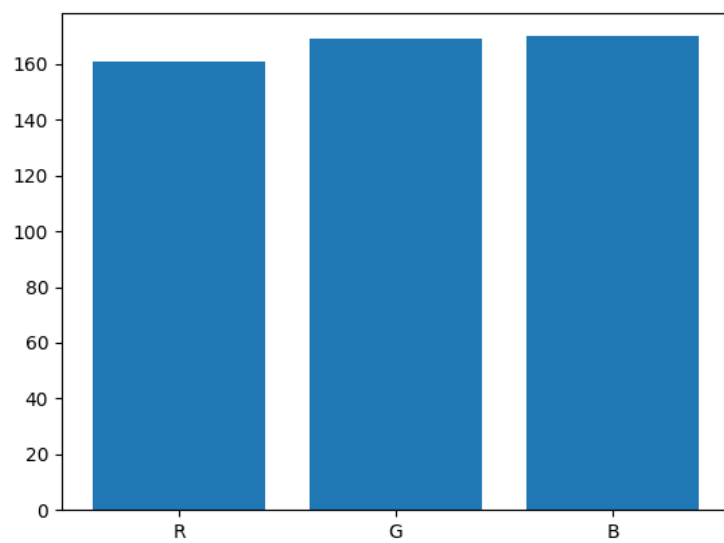


Figura 5: La diferencia es despreciable

Resulta que no hay mucha diferencia de rendimiento entre las formas de los robots, siempre y cuando tengan tamaños similares.

Acerca de los tamaños de los robots

Bueno, entonces... ¿Y el tamaño? Veamos:

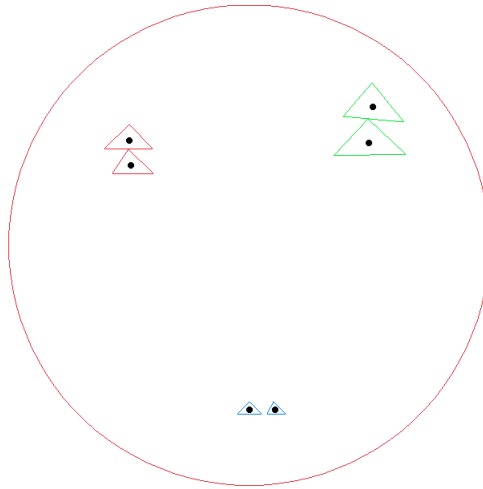


Figura 6: El campo para este experimento. Los equipos Rojo (R), Verde (G) y Azul(B) son de tamaño normal, grande y pequeño respectivamente. La forma de los robocitos son iguales entre los equipos.

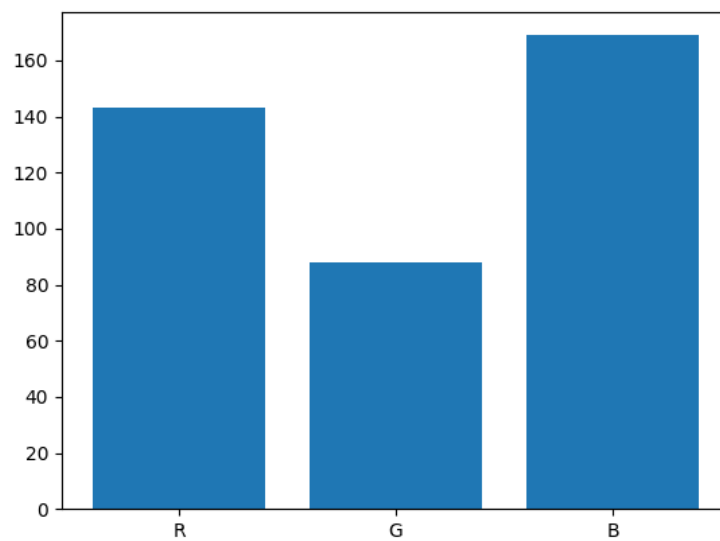


Figura 7: Los robots grandes son muy poco efectivos. Los pequeños son los más efectivos, pero solo un poquito más que los medianos.

Pues los tamaños sí importan. Los robots grandes son extremadamente poco efectivos. Esto no es porque tengan más áreas para ser considerada “fuera del campo”, ya que según las reglas de la simulación, un robot se considera fuera del campo solo cuando su centro de masas lo está, no cuando cualquier pedacito suyo lo esté. Una posible explicación para esta diferencia tan grande es que (como se observa en simulaciones individuales) los robots grandes tienen más superficie para ser golpeados por otros, y en particular por varios otros robots a la vez; que los golpeen varios a la vez por el mismo lado es equivalente a que un robot muy fuerte los golpeen, lo que los pone en desventaja. De hecho, los robots así de grandes suelen perder de primeros, cuando todavía hay muchos otros en el campo.

Acerca de las posiciones iniciales ¿Importa la posición inicial de los equipos en su rendimiento? En particular ¿Qué sucede si un equipo empieza ya en el centro? El centro es importante. Veamos, con equipos completamente idénticos:

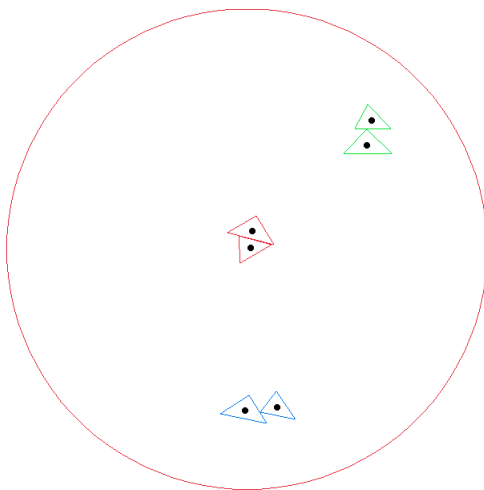


Figura 8: El campo para este experimento.

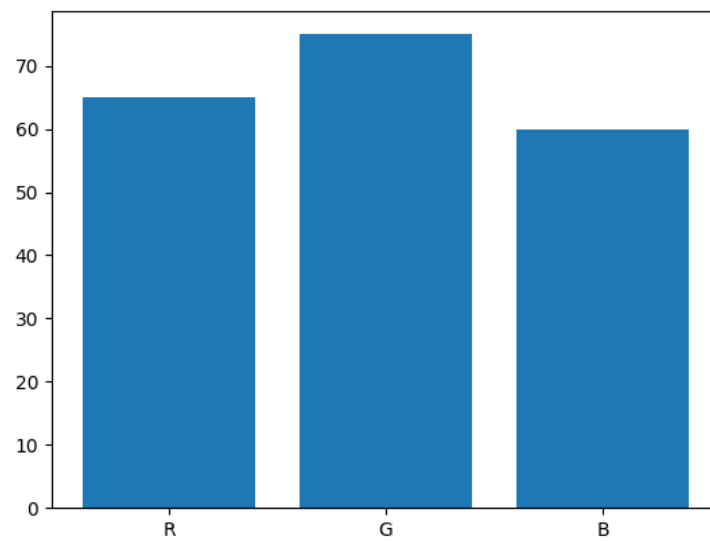


Figura 9: No hay difecencia significativa

No hay diferencia que valga. Empezar en el centro no aporta ni ventaja ni desventaja estratégica. Esto tiene sentido, porque es rápido llegar al centro, el campo es pequeño; además, el estado inicial se diluye cuando transcurre la partida.