**INTRODUCCIÓN AL VIDEO (6 seg.)**

Vamos a aplicar el algoritmo Q-Learning para que una nave espacial aprenda a aterrizar por sí sola.

**EXPLICACIÓN DEL JUEGO (10 seg)**

El juego sobre el que vamos a trabajar se llama Lander y el objetivo consiste en aterrizar una nave sobre una plataforma esquivando los obstáculos y tratando de realizar un aterrizaje suave.

La nave y sus movimientos (7 seg)

Los movimientos que puede realizar la nave son:

* Rotar hacia la izquierda o hacia la derecha.
* Acelerar.
* No hacer nada.

Físicas del juego (35 seg)

Durante el juego, la nave es atraída hacia abajo debido a la gravedad. Si no hacemos ningún movimiento, la nave no se queda en suspensión, sino que empieza a descender ganando velocidad hasta estrellarse.

Además, la nave acelera y desacelera de forma progresiva, es decir, si se desplaza muy rápido en una dirección y quiere corregir su trayectoria, no puede hacerlo de forma instantánea, sino que tiene que rectificar poco a poco.

Estos aspectos son importantes a la hora de plantear el problema porque la nave necesitará tiempo y espacio para reaccionar a cambios de movimiento y debemos tenerlos en cuenta.

**DEFINICIÓN DE ESTADOS. VERSIÓN 1. (1 min 30 seg)**

Lo primero que debemos hacer para utilizar un algoritmo de aprendizaje por refuerzo como es Q-Learning es definir los estados en los que se puede encontrar la nave a lo largo de la partida.

Según cómo definamos los estados, podemos tener un aprendizaje mejor o peor. Sin embargo, en problemas como este, tendremos que ir definiendo los estados mediante aproximaciones sucesivas, es decir, vamos a partir de una definición muy general, vamos a ponerla a prueba y vamos a estudiar si es buena o no.

Para la definición de estados, vamos a considerar los siguientes aspectos:

* En primer lugar, sabemos que la nave tiene una orientación, es decir, hacia dónde apunta la cabeza de la nave. Este valor lo vamos a discretizar de manera que solo pueda tomar 8 valores donde cada valor se diferencia del siguiente en 45º.
* Por otro lado, el hecho de que la nave esté orientada hacia arriba, no quiere decir que se esté desplazando hacia arriba porque la gravedad tira de la nave hacia abajo. Necesitamos conocer hacia dónde se está desplazando la nave en cada instante. Para ello, vamos a guardar el desplazamiento también con 8 valores donde cada valor se diferencia del siguiente en 45º.
* Para que la nave sepa hacia dónde está la plataforma de aterrizaje, vamos a utilizar una brújula que le va a indicar si se debe desplazar hacia el norte, sur, este u oeste.
* Por último, es importante saber si la nave va rápido o no y si hay peligro de impacto o no. Si la nave va rápido y trata de aterrizar, se considerará impacto aunque caiga sobre una plataforma de aterrizaje.

Teniendo en cuenta todas las variables de nuestro problema, el número de estados en los que se puede encontrar la nave son 1024.

**FUNCIÓN DE RECOMPENSA. VERSIÓN 1 (40 seg)**

Cuando llegó el momento de implementar la función de recompensa para esta definición de estados, empezamos por recompensas muy sencillas:

* Ir despacio se recompensaba.
* Reducir la distancia con la plataforma de aterrizaje se recompensaba.
* Ganar se recompensaba.
* Morir se castigaba.

Sin embargo, con estas definiciones no llegamos a conseguir que la nave aprendiera y no aterrizó bien nunca. Cuando veíamos lo que hacía, la nave realizaba movimientos muy bruscos e inestables que hacían que se estrellara. Resultaba un reto conseguir que la nave se mantuviera estable e implementar funciones para que la nave se estabilizara era un trabajo muy tedioso y poco eficaz.

Así que, modificamos la definición de estados inicial para tratar de conseguir que la nave aprendiera a estabilizarse por sí misma.

**DEFINICIÓN DE ESTADOS. VERSIÓN 2 (30 seg)**

La nueva definición de estados se centra en mantener a la nave en zonas de estabilidad. Para ello, dividimos los posibles valores de orientación y desplazamiento de manera que la orientación se mantiene sobre la parte alta y el desplazamiento sobre la parte baja.

Si alguno de los valores se sale de las zonas de seguridad, consideramos que la nave se encuentra en una situación de peligro y la castigamos para que vuelva a una situación estable. La brújula y la variable de velocidad se mantienen tal y como estaban en la versión anterior y la variable de peligro se elimina.

**FUNCIÓN DE RECOMPENSA. VERSIÓN 2 (24 seg)**

La función de recompensa se reduce a premiar que la nave se mantenga en estados estables y seguros como: orientación vertical o levemente ladeada, desplazamiento vertical o levemente ladeado y velocidad baja.

Aplicando esta definición y esta función de recompensa, si la nave se encuentra justo encima de la plataforma, aterriza perfectamente y corrige las desviaciones que pueda sufrir durante el descenso.

(Vídeo de demostración)

**ASPECTOS A TENER EN CUENTA (1 min)**

Para que el aprendizaje se realice de forma adecuada, es necesario tener en cuenta ciertos aspectos:

* La nave se considera que va rápido cuando su velocidad es superior a 9 puntos. La nave necesita tiempo para poder rectificar, así que, nosotros vamos a considerar que la nave va rápido cuando su velocidad supera los 7 puntos. Sin embargo, si hacemos esto, es posible que la nave gane cuando considera que va rápido y esto no lo podemos permitir porque la estaríamos confundiendo.
* Al principio del juego, la nave selecciona uno de los bloques de plataforma para aterrizar. Concretamente, selecciona el bloque central de la plataforma más cercana. Es posible que la nave no aterrice en el bloque objetivo sino justo al lado y puede pensar que ha ganado y que lo ha hecho bien. Sin embargo, esto no es cierto, porque si la plataforma de aterrizaje tiene solo un bloque y cae al lado, la nave se estrella.

Por tanto, debemos recompensar la victoria cuando aterrice despacio y cuando haya aterrizado justo en el centro.

**¿CÓMO ESQUIVA BLOQUES ESTA DEFINICIÓN DE ESTADOS? (1 min)**

Hemos dicho que hemos eliminado la variable que indicaba si había peligro o no. El motivo es porque vamos a utilizar la brújula para indicarle a la nave cuándo está en peligro de impacto. La brújula va a actuar como una guía de vuelo.

Supongamos que la bandera se encuentra a la derecha y la nave comienza a desplazarse manteniéndose siempre dentro de lo que hemos definido como estados seguros. Como es de esperar, la nave se va a desplazar hacia abajo y levemente hacia la derecha para no entrar en la zona de inestabilidad.

Cuando esté próxima al suelo, la brújula va a indicar norte para que la nave gane altura y así se aleje del suelo. Poco a poco, la nave se aproximará a la plataforma de aterrizaje y cuando esté encima de ella, aterrizará.

Para que la nave haga esto, es necesario que en la función de recompensa se premie el hecho de desplazarse en la dirección que indica la brújula.

**PROBLEMAS CON LA DEFINICIÓN DE ESTADOS (1 min)**

Hasta ahora, la nave aterriza bien cuando se encuentra justo encima de la plataforma. Cuando incorporamos la funcionalidad de la brújula para que indicara cuándo la nave debía ganar altura, nos dimos cuenta de que no funcionaba como esperábamos.

Teníamos dos grandes problemas:

* Por un lado, la nave no se desplazaba lateralmente. Probamos con aumentar el ángulo de la zona de seguridad, pero el resultado fue una nave muy inestable que no se controlaba bien.
* Por otro lado, la nave no ganaba altura cuando la brújula le indicaba peligro. Tratamos de dar una recompensa muy alta a obedecer a la brújula, pero tampoco funcionó. La raíz del problema estaba en lo que considerábamos como “obedecer a la brújula”. Habíamos diseñado esto de la siguiente forma: si la brújula indicaba norte antes y ahora la nave está una celda por arriba, entonces ha obedecido.

Sin embargo, esto no es cierto. Recordemos que la nave necesita tiempo para rectificar su trayectoria y que la fuerza de la gravedad tira de ella. Aunque la nave hiciera la acción correcta, debía hacerla muchas veces consecutivas para conseguir corregir la trayectoria, ganar altura y recibir la recompensa.

La forma de definir la recompensa para la brújula acabó siendo muy compleja y no daba los resultados esperados y, además, la definición de estados que habíamos hecho no permitía que la nave hiciera desplazamientos laterales. Así que, había que replantear el problema.