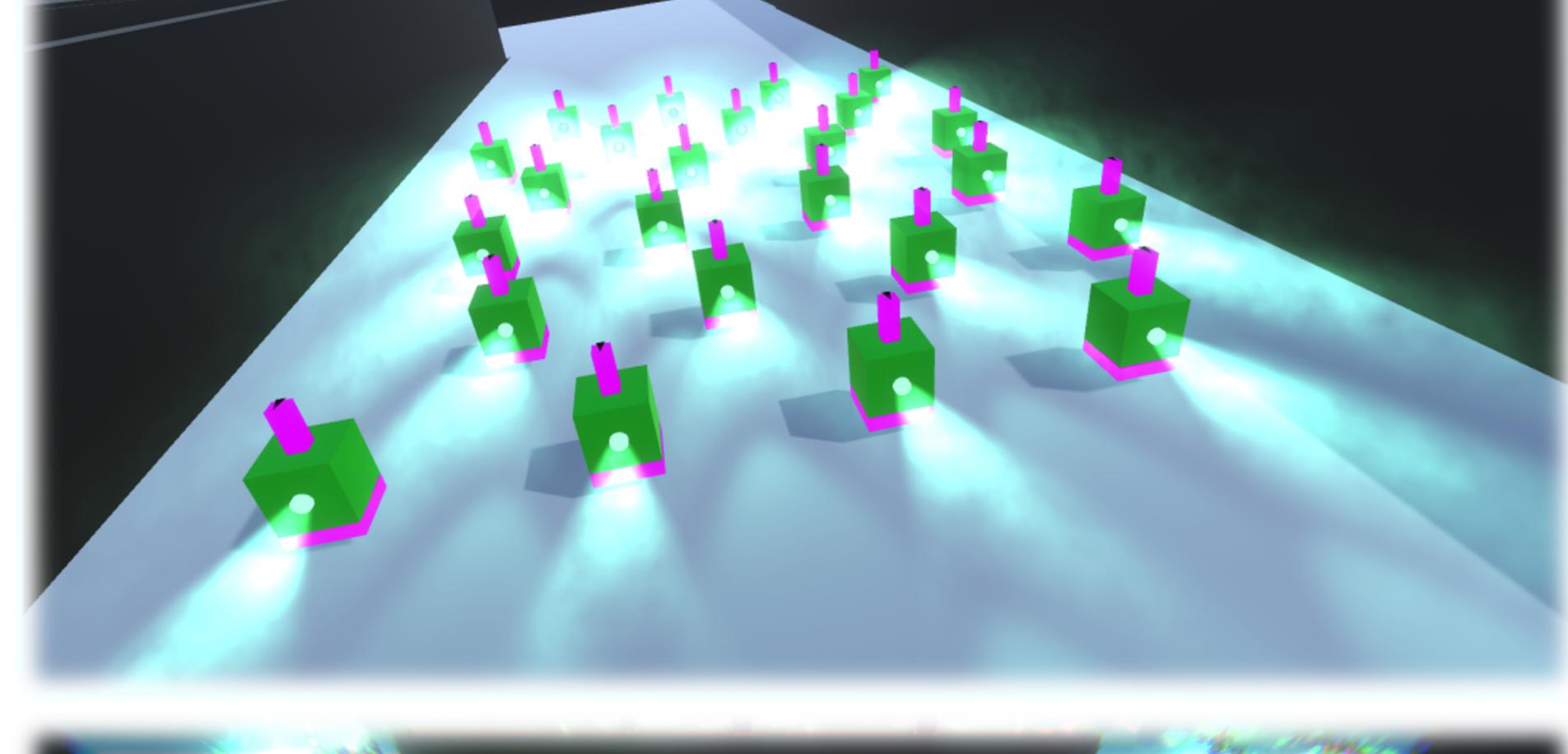


## Conductores Autónomos Libres para Inteligencia Artificial de Juegos en Tiempo Real

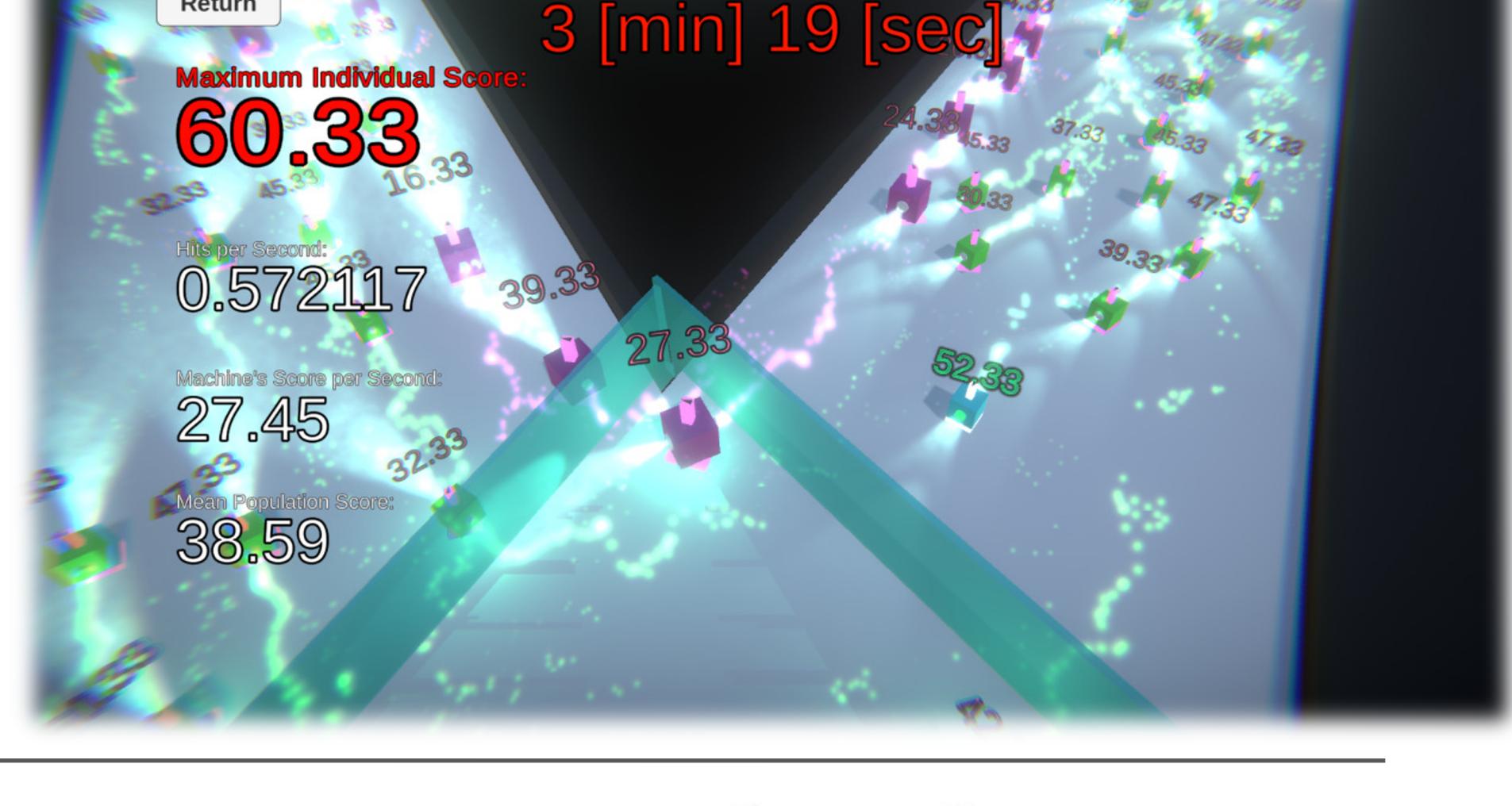
## PROBLEMA

- Es fácil para los jugadores discernir patrones en agentes creados con las técnicas tradicionales de inteligencia artificial para juegos, aburriéndose. Dado que la inmersión de los juegos es tan frágil, ¿podemos crear, utilizando aprendizaje por refuerzo, agentes autónomos para un juego de carreras que continuamente sorprendan a los jugadores?



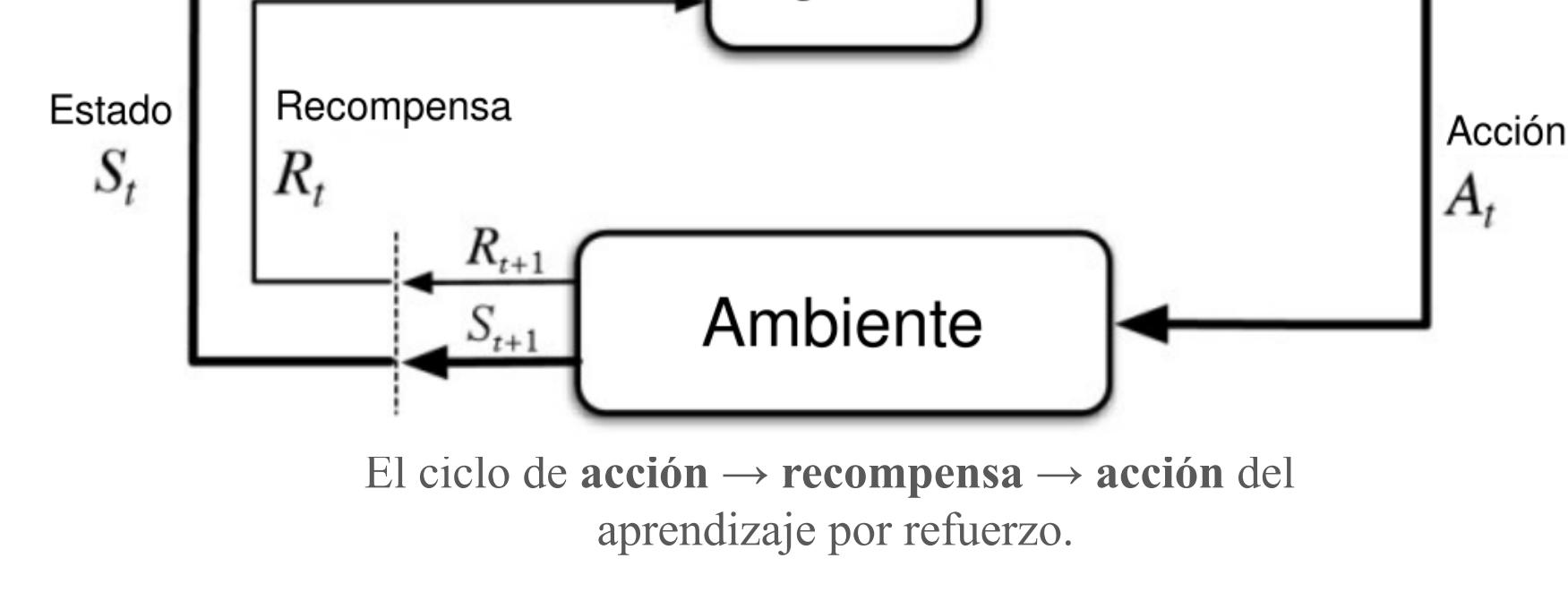
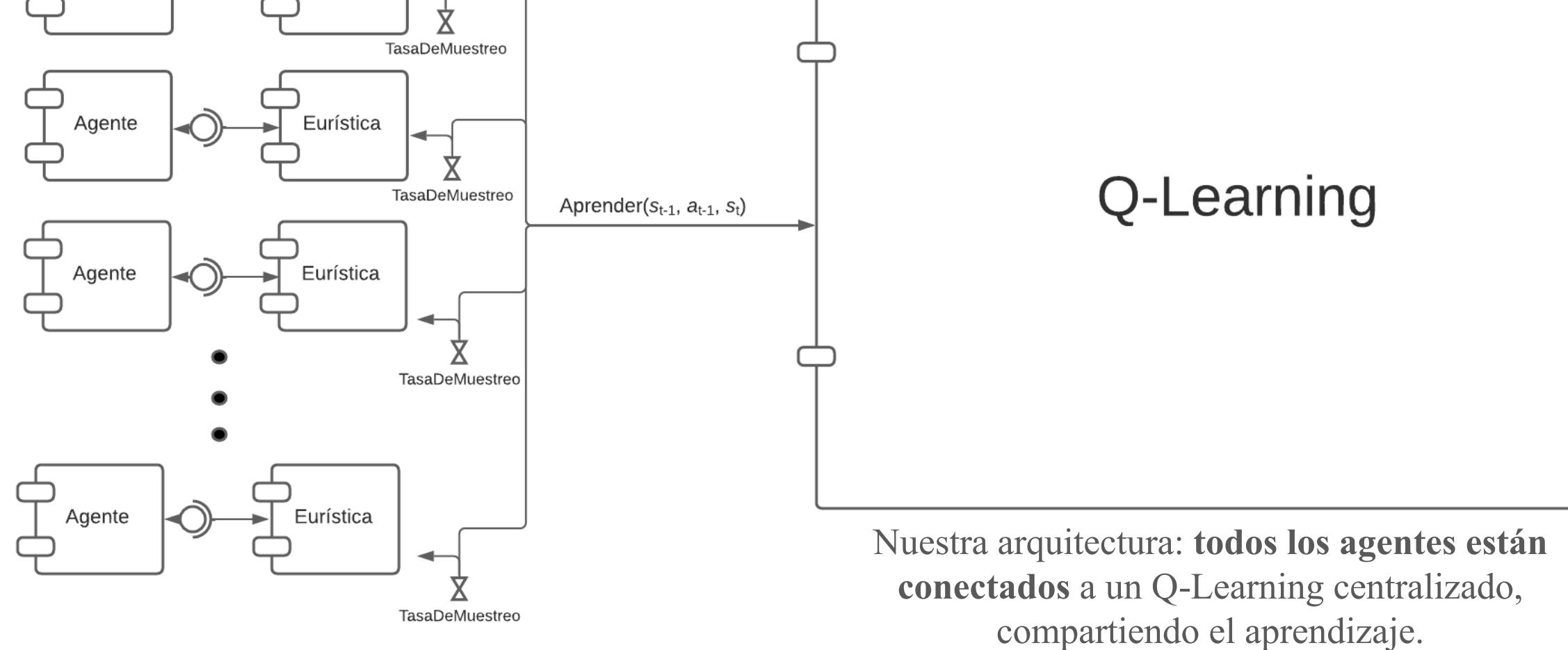
## OBJETIVO GENERAL

- Desarrollar agentes autónomos para aplicaciones de juego en tiempo real.** Específicamente, buscamos diseñar agentes que eviten obstáculos en su camino cuando se les presenta un destino objetivo.
- Desarrollar**, usando diseño experimental, **un juego de carreras** que explote las fortalezas de estos agentes para transportar al jugador a una experiencia maravillosa.



## SOLUCIÓN

- Se creó un algoritmo de Q-Learning:** i) centralizado, ii) basado en muestreo, iii) en línea, iv) entrenado con sub-muestreo y v) cuyo desempeño fue mejorado post-entrenamiento con super-muestreo.
- Se usaron eurísticas para el cálculo de la dirección óptima de los agentes**, la cual Q-Learning transforma, usando información muestreada del ambiente, en la dirección real que los agentes deben tomar para evitar obstáculos. Variando la eurística, se construyeron dos tipos de agentes: voraces y vagos.



Algorithm 1: Online, Sample-Based Q-Learning

```

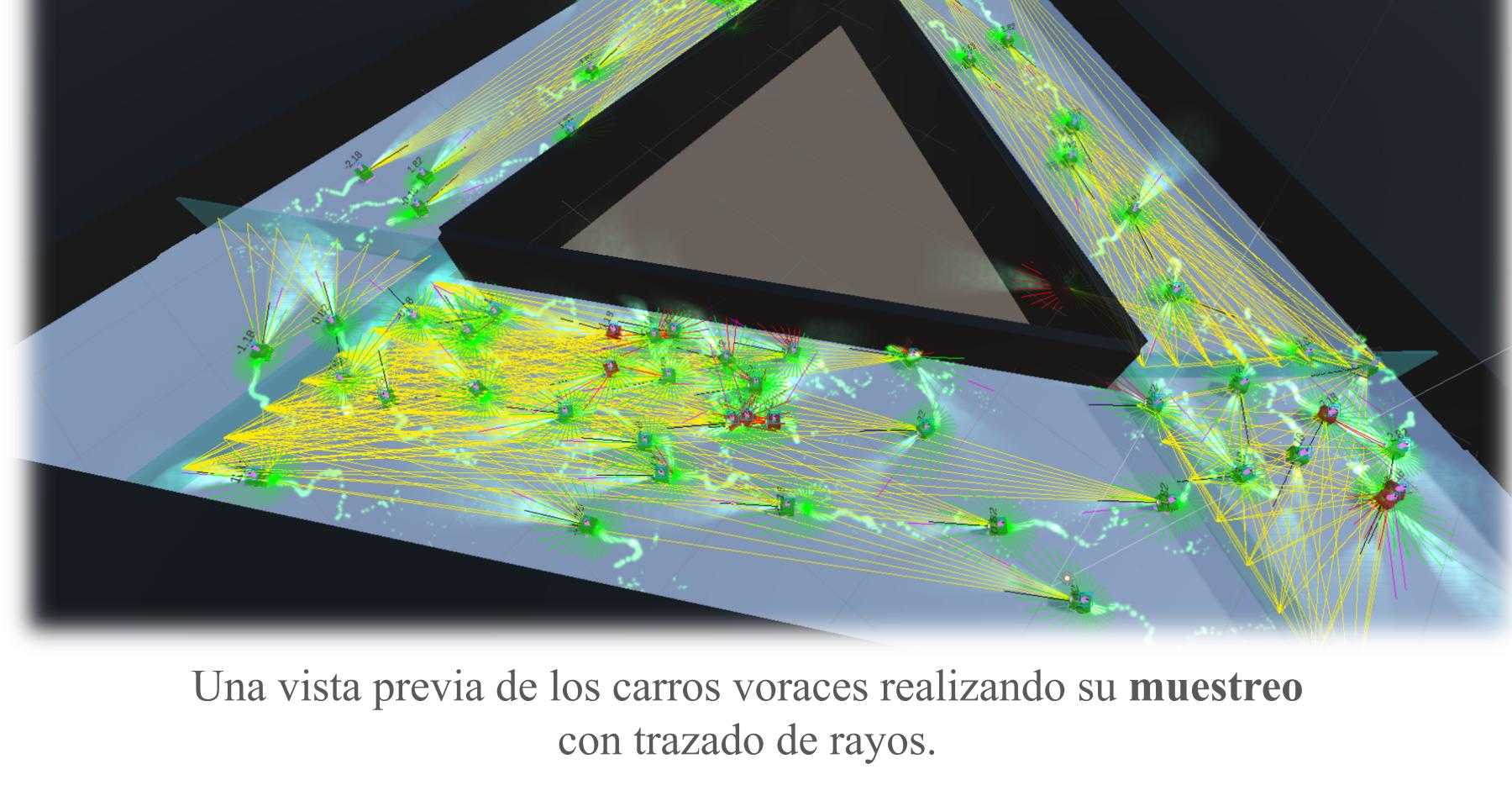
1 Function Learn( $s_{t-1}, a_{t-1}, s_t$ ):  $a_t$  is
2   if Random(0, 1) <  $\rho$  then
3     |  $a_t \leftarrow$  GetRandomAction( $s_t$ )
4   end
5   else
6     |  $a_t \leftarrow$  GetBestAction( $Q$ ,  $s_t$ )
7   end
8    $r \leftarrow$  GetReward( $s_t$ )
9    $Q(s_{t-1}, a_{t-1}) = (1 - \alpha) \cdot Q(s_{t-1}, a_{t-1}) + \alpha \cdot (r + \gamma \cdot Q(s_t, a_t))$ 
10  return  $a_t$ 
11 end

```

Nuestro algoritmo personalizado de Q-Learning.

## RESULTADOS

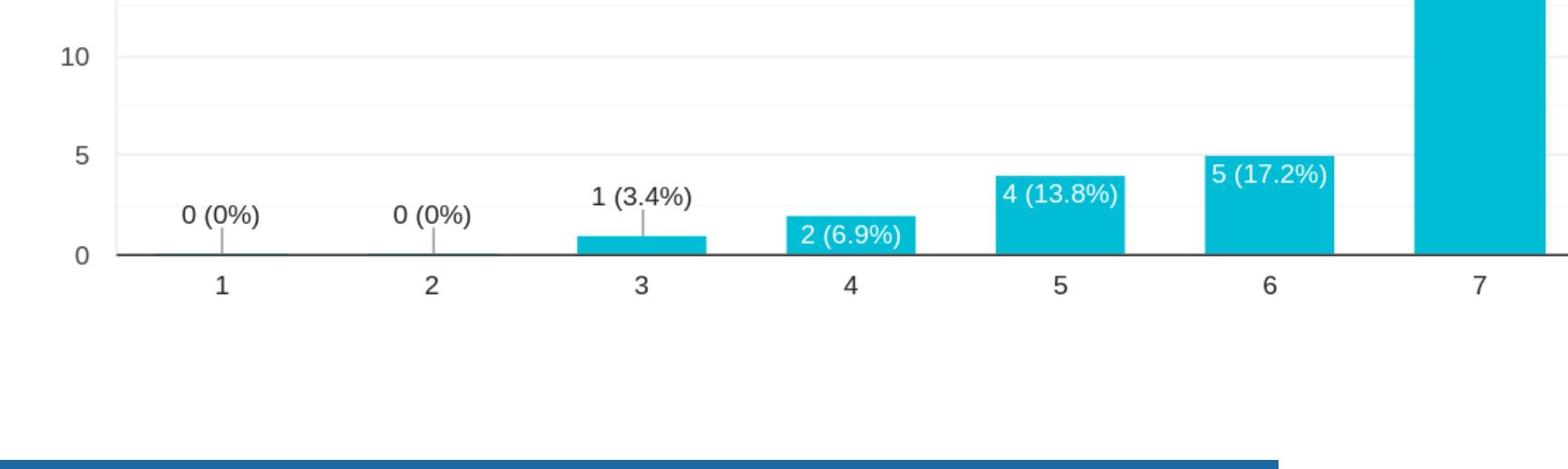
- Sin Q-Learning, usando solo la eurística para el cálculo de las direcciones, el desempeño obtenido fue de 17.5 colisiones por segundo. **Añadiento Q-Learning con super-muestreo, obtuvimos un resultado de 1 colisión por segundo.**
- El juego fue evaluado con 29 jugadores. Calificaron la experiencia con un **puntaje de 6.2 sobre 7**, y todos los jugadores lo clasificaron como **no determinístico**.



Una vista previa de los carros voraces realizando su muestreo con trazado de rayos.

¿Qué puntaje le daría a la experiencia que brinda este juego sobre 7?

29 responses



## CONCLUSIONES

- La **cuantización**—el proceso de llevar estados continuos e infinitos hacia estados discretos y contables—es un arte delicado donde se debe balancear **resolución versus memoria**.
- Sub-muestreo** hace posible el proceso de entrenamiento de los agentes en espacios continuos cuantizados.
- Una vez entrenados, **super-muestreo** mejora el desempeño de los agentes considerablemente.
- Aprendizaje por refuerzo logró crear un juego cuyos **jugadores lo denominan como no determinístico**.
- El siguiente paso es **masificar las evaluaciones del juego** utilizando el avalado *Game Experience Questionnaire*.

## NUESTRA CONTRIBUCIÓN

- Nuestro aporte científico inédito fue el descubrimiento de una nueva técnica para el diseño de juegos: **La Mente de la Colmena y el Núcleo de la Originalidad**. En este método, se usa un algoritmo de Q-Learning centralizado para unificar el comportamiento de los agentes de la colmena, pero se influye, externamente, el espacio de estados de Q-Learning para dar distintos temperamentos a las subclases de agentes. Está emergiendo un *paper* que será candidato para el congreso indexado *Game-On 2021* de la Universidad de Aveiro, Portugal.