

PRIORIZACIÓN DE OBJETIVOS: BÚSQUEDA EN ESPACIO DE SOLUCIONES EN LEAGUE OF LEGENDS. (septiembre, 2023)

Daza Pereira Juan Pablo y Camargo Sanchez Juan Sebastian

Fundamentos de la Inteligencia Artificial

Escuela Colombiana de Ingeniería Julio Garavito

Resumen- Priorizar objetivos en juegos como League of Legends es sustancial para optimizar el rendimiento y aumentar las posibilidades de tener una victoria. Esto implica la toma de decisiones estratégicas sobre qué objetivos, como los dragones o torretas, que buscan alcanzar los jugadores en función de la situación del juego. Este proceso se asimila a la búsqueda en espacio de soluciones sin adversario utilizada en la inteligencia artificial. El algoritmo de búsqueda A* será empleado para encontrar la ruta más eficiente para alcanzar los objetivos del juego, considerando varios factores. Al evaluar constantemente las situaciones y adaptaciones estratégicas en los objetivos, los jugadores pueden acumular ventajas, controlar el mapa y aumentar en un gran porcentaje sus posibilidades de ganar.

Palabras clave- Adaptación táctica, Búsqueda A*, Control del mapa, Coordinación estratégica, Estrategia de juego, Priorización de objetivos, Rutas eficientes, Toma de decisiones, Ventajas estratégicas.

I. INTRODUCCIÓN

League of legends (LoL), es uno de los videojuegos más populares a nivel mundial en la categoría de eSports, y es conocido por su complejidad estratégica y táctica.

En este escenario competitivo, la priorización estratégica de objetivos es el pilar fundamental para obtener éxito en el juego. Este artículo de investigación se va a centrar en el tema de **priorización de objetivos** con un enfoque innovador usando el algoritmo de búsqueda A*, en un escenario reducido, sin adversarios. El objetivo principal para destacar en este artículo es cómo la utilización de A* puede llevar a la optimización de decisiones estratégicas, exclusivamente en la priorización de objetivos como los súbditos, torretas y monstruos de la jungla, que son los elementos u objetivos esenciales para encontrar una victoria en el juego.

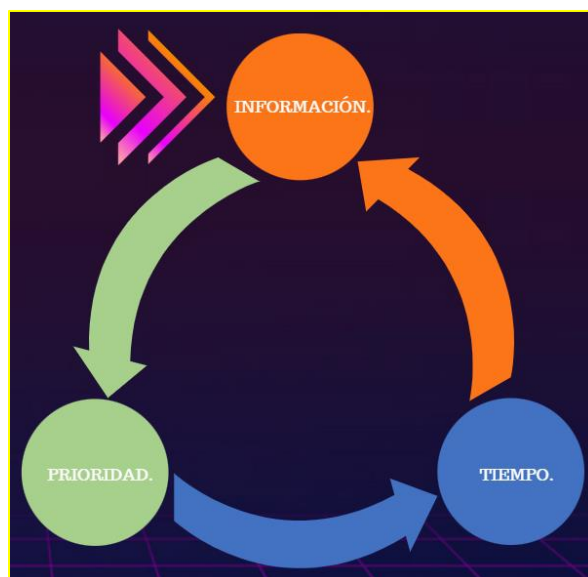
En esta abstracción, veremos cómo A* puede ser una valiosa herramienta para maximizar y explotar, la eficiencia de recursos y control del mapa en un solo carril, lo que, a futuro, teniendo esas expectativas, podrá informar a los jugadores y equipo en entornos competitivos acerca de su efectividad en la partida. Esta investigación busca fuentes sobre la importancia de la priorización de objetivos y cómo el uso de algoritmos de búsqueda puede contribuir a este maravilloso juego como lo es League of Legends.

II. METODOLOGÍA

En League of Legends, los jugadores asumen el papel de invocadores en la grieta del invocador, donde controlan campeones con habilidades únicas, en batallas de equipos 5 vs 5. El objetivo principal del juego es destruir el Nexa enemigo, que se ubica en la base enemiga. Para lograr esto, los jugadores deberán trabajar en equipo, luchar contra los campeones enemigos, derribar las torretas defensivas, matar los monstruos de la jungla, adquirir oro y experiencia para mejorar a los campeones. League of Legends al ser un juego altamente competitivo que combina estrategia, trabajo cooperativo y habilidad individual, y que su popularidad es gracias a su enfoque competitivo y de su evolución constante, hemos decidido, para esta investigación, reducir ese mundo tan grande a un mundo abstracto que nos

permitirá cumplir con el objetivo. Como se ha dicho a lo largo de este artículo, priorizar objetivos en League of Legends es algo esencial para el éxito en el juego, pero también puede ser un problema debido a varios factores que los jugadores deben considerar. Los principales problemas que los jugadores se encuentran a la hora de priorizar objetivos son la información limitada, los cambios en la prioridad y la presión del tiempo. A continuación, daremos una breve explicación de cada problema mencionado anteriormente:

- **Información limitada:** Los jugadores casi siempre tienen información limitada sobre el mapa, los monstruos de la jungla y la forma de adquirir oro para comprar ítems y poder mejorar a los campeones. Esto puede dificultar la toma de decisiones y a la vez hace perder el objetivo principal del juego.
- **Cambios en la prioridad:** La prioridad de los objetivos va cambiando a lo largo de la partida. Por ejemplo, al inicio de cada partida, los súbditos son una prioridad, mientras que ya en una fase media de la partida, los objetivos serán las torretas, luego el Barón Nashor y los dragones. Por ello los jugadores deben adaptarse a estos constantes cambios.
- **Presión del tiempo:** En League of Legends, el tiempo es un recurso muy importante y limitado, y tomar demasiado tiempo para priorizar un objetivo puede conllevar a perder oportunidades valiosas en la partida. Por ello es necesario aprender a priorizar ese tiempo y a tomar decisiones rápidas bajo presión.



Resolver estos tres problemas, es fundamental por varias razones:

- Optimización de recursos: Los recursos como el oro y la experiencia son limitados y deben hacerse de manera efectiva. La priorización de objetivos adecuada garantiza que el jugador está maximizando su capacidad de adquirir recursos, para poder mejorar a su campeón.
- Ventajas estratégicas: La elección de buenos objetivos de acuerdo con la estrategia planificada puede llevar a una ejecución efectiva y aumentar la probabilidad de ganar la partida.
- Prevención de pérdidas innecesarias: El no priorizar bien los objetivos puede llevar a grandes pérdidas, como la muerte de nuestro campeón o la pérdida de torretas defensoras. Es decir, una mala elección de objetivos puede hacer que la partida sea más difícil de ganar.

Como se dijo antes, la idea de solución que tuvimos frente a este problema fue tomar un mundo tan grande de muchas estrategias, tácticas y formas de ganar, y lo redujimos a un mundo más pequeño, un mundo abstracto del juego. Lo primero fue tomarlo como un juego sin adversario (jugador vs bot),

reducirlo en un solo carril (Top, Mid o Bot) y priorizar los objetivos principales (Súbditos, Torretas, Monstruos de la jungla, Inhibidor y Nexo).

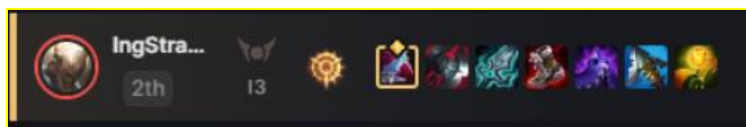


Jugador vs bots en un solo carril.



Carriles y objetivos.

Al cumplir cada uno de los objetivos, el jugador irá adquiriendo oro que le va a servir para comprar ítems que volverán a su campeón más fuerte y poderoso. Esos ítems a medida que se van comprando restan oro, pero suman experiencia y poder en las habilidades del campeón.



Items del campeón para alcanzar el objetivo (NOTA: no tienen que ser exactamente los ítems de la imagen presentada).

Cabe destacar que en nuestra situación entran todos los campeones disponibles en League of Legends, a excepción de los campeones “Soporte”. Los campeones soporte no los tomamos en cuenta por dos razones. La primera es porque al ser soportes, van a depender siempre de su adc (compañero del carril Bot) ya que este es el encargado de cumplir con los objetivos de ese carril. Y la segunda razón es que, al depender de su compañero de carril, en la idea que estamos aplicando, le será mucho más difícil sumar objetivos porque son campeones con un nivel de habilidades menor.



Campeones disponibles para este problema.

Antes de dar a conocer los resultados, si no ha quedado claro a lo largo del artículo, la heurística que vamos a usar es “Priorización de objetivos”.

III. RESULTADOS

En la implementación de la solución se planteamos lo siguiente para nuestro heurístico y nuestro costo de alcance:

- Recompensa: Cada objetivo, asesinato, asistencia del juego tiene un valor en oro que nos dará si lo cumplimos.
- Tiempo: el tiempo que se requiere para cumplir el objetivo para el caso de los asesinatos y las asistencias es diferente (ya que pueden ocurrir en cualquier momento), se calcula según su KDA (Kills-Deaths-Assitence) sobre su promedio de duración de partida.
- Riesgo: Clasificamos del 1 al 10 que tan arriesgado es cumplir con el objetivo.

Por último, le asignamos a cada una de estas variables un peso ya que algunas variables son más importantes a la hora de hacer una estrategia. Por ejemplo, decir que la recompensa tiene más peso que el tiempo. El riesgo sería un peso negativo que usaremos para equilibrar nuestra estrategia.




Como ambos, costo de alcance y heurístico, usarán las mismas variables se diferenciarán porque variables usamos. Habíamos planteado usar estadísticas que nos provee el juego como es el KDA y la duración de las partidas. Para el costo de alcance usaremos las estadísticas del jugador del personaje que este jugando en la partida, junto con el nivel de juego (Hierro, Bronce, Plata, Oro, etc.). Para el costo heurístico serán estas mismas estadísticas, pero a nivel general.

Las variables no van a ser estáticas para el árbol estas van a ir cambiando a través del tiempo, ósea van a ir cambiando con la profundidad del árbol de caminos representando el cambio del tiempo. Esto con el fin de que con el tiempo se le van a ser más fácil llegar a cumplir objetivos que inicialmente eran inalcanzables.

Creemos que aplicando esto llevara a que la toma de decisiones del jugador sea orientada a su nivel de juego junto a su desempeño en partida a través del tiempo.

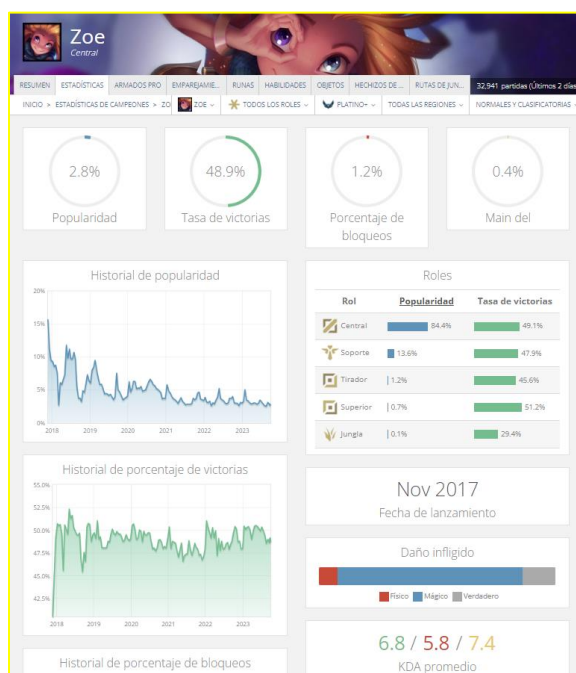
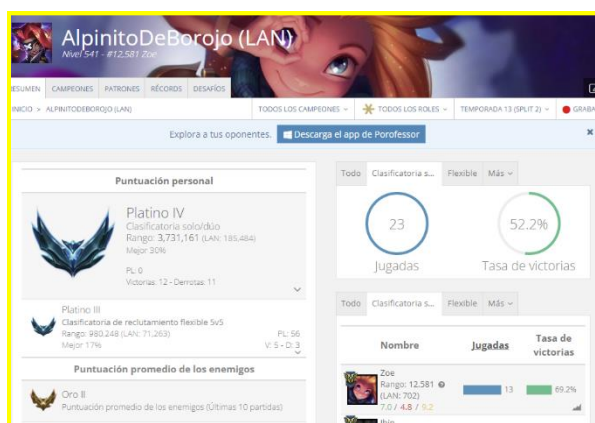
Ya implementado el código, los resultados no fueron exitoso.

Como entrada insertamos el perfil de un jugador que lleva el personaje de Zoe y tiene las siguientes estadísticas de KDA :




























Campeón	Jugadas	Tasa de victorias	KDA	Súbditos / min	Oro / min	Pentakills	Rango
 Zoe	 22	 72.7%	8.5 / 6.2 / 13.2	4.1	435.8	0	12,565 (LAN: 701)

Se calculo la duración de estadísticas promedio tomando las ultimas 10 partidas de este jugador.

Ahora para las estadísticas generales vemos que es de nivel Platino en el juego y buscamos el nivel general de Zoe en Platino:



Ahora para las estadísticas generale la estadística de duración de partida de un Platino:

Puntuación	<u>Duración de la partida Incl. rendiciones</u>	<u>Duración de la partida Excl. rendiciones</u>
 Hierro	 30:16	 32:57
 Bronce	 30:01	 32:27
 Plata	 29:41	 31:53
 Oro	 29:08	 31:15
 Platino	 28:47	 30:48
 Esmeralda	 28:06	 30:07
 Diamante	 26:50	 29:05
 Maestro	 25:43	 28:18
 Gran maestro	 25:21	 28:09

Código del cálculo de valor heurístico y cálculo de valor de alcance:

```
def heuristicCost(e, pasoTiempo):
    tiempoValue = 0

    if e.tipo == "O":
        aux = generaPlayerStat.duracion - e.tiempo - pasoTiempo
        if 0 > aux:
            aux = 0
        tiempoValue = aux * 0.5

    if e.tipo == "K":
        aux = (generaPlayerStat.duracion / generaPlayerStat.kill)
        tiempoValue = aux * 0.5
    elif e.tipo == "A":
        aux = (generaPlayerStat.duracion / generaPlayerStat.kill)
        tiempoValue = aux * 0.5

    recompensaValue = e.oro * 0.5
    riesgoValue = e.riesgo * 0.0
    print(tiempoValue, recompensaValue, riesgoValue)
    print(tiempoValue + recompensaValue - riesgoValue)
    return tiempoValue + recompensaValue - riesgoValue
```

```
def actionCost(e1,e2,pasoTiempo):
    tiempoValue=0;

    if e2.tipo == "O":
        aux = playerStat.duracion-e2.tiempo - pasoTiempo
        if 0 > aux:
            aux = 0
        tiempoValue = aux * 0.5

    if e2.tipo=="K":
        aux = (playerStat.duracion/playerStat.kill)-pasoTiempo
        tiempoValue=aux*0.5
    elif e2.tipo=="A":
        aux = (playerStat.duracion /playerStat.kill)-pasoTiempo
        tiempoValue = aux * 0.5

    recompensaValue = e2.oro * 0.5
    riesgoValue = e2.riesgo *5
    print(tiempoValue, recompensaValue, riesgoValue)
    print(tiempoValue + recompensaValue - riesgoValue)
    return tiempoValue + recompensaValue - riesgoValue
```

Aquí implementamos el código como se plantemos a un inicio. Hacemos la ejecución con ya implementados el camino más corto A:

```
2.2058823529411766 150.0 0.0
152.2058823529412
25 25
50
[['inicio', 'Dragon'], ['Dragon', 'Dragon']]
Traceback (most recent call last):
  File "C:\Users\juanc\PycharmProjects\IA\src\python\Proyecto.py", line 257, in <module>
    a(root)
  File "C:\Users\juanc\PycharmProjects\IA\src\python\Proyecto.py", line 255, in a
    return a(aux,path,pasoTiempo,acum)
  File "C:\Users\juanc\PycharmProjects\IA\src\python\Proyecto.py", line 255, in a
    return a(aux,path,pasoTiempo,acum)
  File "C:\Users\juanc\PycharmProjects\IA\src\python\Proyecto.py", line 251, in a
    path.append([e.name, aux.name])
AttributeError: 'str' object has no attribute 'name'
```

Analizando el resultado vimos que crear un árbol con las diversas posibilidades que tiene es mucho trabajo y muy extenso. Ahora el resultado del algoritmo nos dice que el dragón es la mejor opción para obtener oro, pero la como jugadores sabemos que esto no es correcto. La mejor decisión para tomar en los primeros momentos de una partida es ir por los Súbditos, que son la fuente más segura y provechosa para el jugador. Viendo los resultados y el cálculo los valores heurísticos y de alcance nos dimos cuenta de que los pesos y las variables no están relacionadas por algo que las una y que podamos decir que es una unidad

de medida. Por lo que creemos que tocaría reformular el cálculo y el sistema de pesos que nos lleven mejor abstracción.

IV. CONCLUSIONES

En el recorrido de esta investigación, exploramos el uso del algoritmo A* en un entorno como lo es League of Legends, donde la priorización de objetivos rige como un elemento principal e importante para la victoria del juego. Los resultados obtenidos nos han permitido tener una eficaz metodología en la optimización de la toma de decisiones estratégicas, especialmente en la priorización de súbditos, torretas y monstruos de la jungla.

Estudiar este problema abstracto nos ha permitido revelar que la aplicación de A* bien aplicada puede brindar a los jugadores una ventaja significativa en la parte de optimización de recursos y control del juego en un solo carril. Esto también muestra que la heurística de priorización de objetivos respaldada por un algoritmo A* podría ser una herramienta valiosa para mejorar a futuro en el entorno competitivo de League of Legends.

Creemos que con una buena abstracción de la toma de decisiones estratégicas en el juego se podría crear un mapa de buenas decisiones nos direccionen a tener ventajas en campos competitivos del juego. Además de crear un ejemplo en la abstracción de escenarios donde la toma de decisiones estratégicas sea lo principal para el éxito.

A medida que vayamos explorando más del mundo de la Inteligencia Artificial y de sus métodos de búsqueda para aplicarlos en el contexto de League of Legends y los eSports, es fundamental aprender y reconocer cómo todas estas herramientas pueden contribuir en nuestra investigación.

V. REFERENCIAS

[1] Riot Games, League of Legends y PvP.net. Disponible:

<https://www.leagueoflegends.com>

[2] Stuart Russell y Peter Norvig, “Inteligencia Artificial, un enfoque moderno”, segunda edición 2004.

[3] Blitz.gg (*lugar donde se tomaron estadísticas e información*). Disponible:

<https://blitz.gg/>

[4] Para las estadísticas del juego tomamos la página <https://www.leagueofgraphs.com/es/> , donde se pueden encontrar las estadísticas que planteamos.