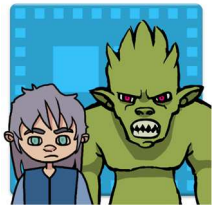
Resolución de Wappo mediante técnicas de búsqueda

Inteligencia Artificial (IS) 2018/19 – Propuesta de trabajo

Luis Valencia Cabrera

1. Introducción y objetivo

El juego denominado "Wappo Game"[1] es un juego desarrollado para Android, aunque su primera versión data de 2003 (creado por *Cingular Media* para *Siemens*). Está inspirado en juegos pre-existentes como Teseo y el minotauro[2]. Se enmarca en el contexto de los puzles lógicos, con reglas tan sencillas como soluciones a menudo complicadas.



El objetivo del juego consiste en, partiendo de un mapa de tamaño prefijado (generalmente una cuadrícula 6x6), lograr llegar a una casa situada fuera del mapa, accesible directamente desde una de las casillas del mismo. Podemos ver un ejemplo en la siguiente imagen:

Como se puede apreciar, no solamente hay una cuadrícula y una serie de casillas, sino también componentes para dificultar y/o animar el juego:

[1] https://play.google.com/store/apps/details?id=com.pinkzero.wappogame&hl=es_419
[2] https://www.logicmazes.com/theseus.html, https://www.microsiervos.com/archivo/juegos-y-diversion/teseo-y-el-minotauro.html

- Monstruos, que si nos atrapan nos hará caer derrotados.
- Trampas, en las que si caemos perdemos también automáticamente, pero si hacemos caer en ellas a los monstruos les hacemos perder tres turnos.
- Muros, que impiden a los monstruos pasar.

En cada turno del jugador humano, movemos una casilla hacia arriba, abajo, izquierda o derecha. Automáticamente, el turno de los monstruos que pueda haber en la pantalla quedará determinado siguiendo sus reglas básicas de comportamiento. En su versión básica:

- El humano mueve una sola casilla
- El monstruo mueve hasta un total de 2, teniendo en cuenta que:
    - Siempre trata de ir directo siguiendo el camino más corto. Lo lleva al extremo, de modo que cada paso que da sigue el criterio siguiente:
        - Si se encuentra en nuestra fila, trata de ir hacia nosotros, y si hay un muro que lo impida no avanza, no busca alternativas[3].
        - Si se encuentre en nuestra columna, trata de seguirnos, y de nuevo se para si se encuentra un muro.
        - Si se encuentra en una fila y columna distinta, intenta en primer lugar moverse de manera horizontal, y si no es posible lo hace en vertical.

Para más información sobre la dinámica del juego se recomienda descargar el juego gratuitamente del enlace anterior, así como visualizar vídeos de Youtube en los que se puede ver la solución de partidas superando distintos niveles[4].

El núcleo del trabajo se refiere al "Modo clásico" del juego. Existe adicionalmente un "Modo infierno", así como otras versiones del juego como el "Wappo II", que en principio estarían fuera de los objetivos planteados; únicamente si se superaran todos los objetivos y se dispusiera de tiempo para extender el desarrollo a cubrir total o parcialmente esos otros modos y variantes podría ser interesante analizar qué nuevos retos plantean esas extensiones, y se valorarían como tal.

El **objetivo principal** que se debe acometer dentro del trabajo propuesto es la resolución de cualquier puzle del *Wapo Game* (modo clásico) mediante **técnicas de búsqueda** de modo que dado una situación inicial nos diga qué secuencia de movimientos debemos hacer como personaje para llegar al destino sin ser derrotados. Para ello, se debe comenzar con la **representación del problema como espacio de estados**, si bien se deja la opción de basarse, si se prefiere, en el *formalismo PDDL y enfocarlo como problema de planificación*. Téngase presente que la elección de las técnicas de búsqueda adecuadas y las definiciones de funciones de coste y sobre todo heurísticas idóneas para el problema en cuestión serán fundamentales para guiar la búsqueda. Se da la opción, si se desea, de emplear técnicas auxiliares, como meta-heurísticas o técnicas de aprendizaje automático, para tratar de ajustar parámetros para las funciones de coste y heurísticas, si bien no constituiría el núcleo de su trabajo sino un añadido interesante en su caso.

Para ello será necesario alcanzar los siguientes objetivos **específicos**:

1. Representar el problema como espacio de estados (o mediante pddl, si opta por planificación), haciendo uso en cualquier caso de las bibliotecas vistas en clase para espacio de estados o planificación.
2. Emplear y evaluar la eficacia y eficiencia de distintas técnicas de búsqueda, vistas en clase

[3] https://www.youtube.com/watch?v=RL9fPVy7FGg
[4] https://www.youtube.com/watch?v=RgP4tALtr9E&list=PLHxjCbF9wKl_0mK7Ja1AAp1XCHcAnvdH7

y trabajadas durante las prácticas, proponiendo distintas funciones de coste y heurísticas para los algoritmos de búsqueda informada que aparecieron en las prácticas de clase.
3. Experimentar con múltiples niveles del juego, u otros creados por los miembros del grupo, más o menos sencillos, comprobando que las técnicas proporcionadas logran superar los retos planteados.
4. Dotar de facilidades de uso del programa, de modo que podamos fácilmente proporcionar un determinado mapa de entrada y seleccionar una de las búsquedas propuestas (con distintas técnicas, o bien repitiendo algunas de ellas pero cambiando funciones de coste y/o heurística).
5. Documentar el trabajo en un fichero con formato de artículo científico, explicando con precisión las decisiones de diseño en la representación del problema, los resultados obtenidos en las pruebas de evaluación (número de ejemplos resueltos y tiempo de respuesta).
6. Realizar una presentación (PDF, PowerPoint o similar) de los resultados obtenidos en la defensa del trabajo.

Además, como objetivos complementarios se valorarán las aportaciones cualitativas que se realicen en términos de facilidad de uso, visualización de la dinámica del sistema, situaciones de la partida, ayudas para seguir paso a paso una simulación de partida, claridad del código (estructura de archivos y modularidad, clases, funciones, estilo de programación, etc.), documentación del mismo y otras aportaciones que se consideren significativas desde el punto de vista de la Inteligencia Artificial, las Ciencias de la Computación y la Ingeniería del Software.

Para que el trabajo pueda ser evaluado, se deben satisfacer TODOS los objetivos específicos al completo: el trabajo debe ser original, estar correctamente implementado y funcionar perfectamente, los experimentos se deben haber llevado a cabo y analizados razonadamente, el documento debe ser completo y contener entre 8 y 10 páginas (no más, no menos), y se debe realizar la defensa con una presentación de los resultados obtenidos. Los objetivos complementarios no serán indispensables, pero sí contribuirán a la mejora de la calidad del trabajo, y por tanto influirán en la obtención de una mayor calificación.

2. Apuntes metodológicos

Algunas consideraciones metodológicas relevantes dentro del desarrollo del código se proporcionan a continuación para una mejor guía hacia la consecución de los resultados esperados.

2.1. Configuración inicial

El trabajo debe implementarse de forma parametrizada, para mapas de cualquier tamaño (tanto en número de filas como de columnas, no necesariamente iguales), dados como entrada.

2.2. Búsqueda de la solución

Parte del trabajo consiste en investigar la mejor manera de resolver el puzle mediante algoritmos de búsqueda. Por ello será necesario evaluar distintas formas de representación y/o funciones de coste y/o heurística, que proporcionen los mejores resultados en la búsqueda de la solución. En este proceso es admisible hacer modificaciones sobre los algoritmos de búsqueda implementados en la librería vista en clase, o incluso el desarrollo de nuevos algoritmos de búsqueda.

2.3.Experimentación

La experimentación con mapas parciales o situaciones más cercanas a la solución puede proporcionar ideas sobre cómo mejorar el proceso de búsqueda. Es importante obtener buenos resultados con ejemplos pequeños antes de probar con niveles avanzados. Se deben documentar en ficheros auxiliares o de código los resultados obtenidos en estas pruebas, con los distintos algoritmos de búsqueda evaluados, indicando el tiempo de respuesta y las soluciones.

2.4.Documentación

En el fichero *plantilla-trabajo·doc* se muestra una sugerencia de estructura y formato de estilo artículo científico correspondiente a la documentación del trabajo. Este formato es el del *IEEE conference proceedings*, cuyo sitio web *guía para autores* [1] ofrece información más detallada y plantillas para Word y Latex.

El artículo deberá tener una extensión **entre 8 y 10 páginas**, y la estructura general del documento debe ser como sigue: en primer lugar realizar una **introducción** al trabajo explicando el objetivo fundamental, incluyendo un breve repaso de antecedentes en relación con la temática del trabajo y con los métodos empleados (mencionar referencias bibliográficas), a continuación describir la **estructura** del trabajo, las **decisiones de diseño** que se hayan tomado a lo largo de la elaboración del mismo, y la **metodología** seguida al implementarlo (nunca poner código, pero sí pseudocódigo), y seguidamente detallar los **experimentos** llevados a cabo, **analizando los resultados** obtenidos. Por último, el documento debe incluir una sección de **conclusiones**, y una **bibliografía** donde aparezcan no sólo las referencias citadas en la sección de introducción, sino cualquier documento consultado durante la realización del trabajo (incluidas las referencias web a páginas o repositorios).

2.5.Mejoras

Aunque no sea obligatorio, sí se tendrá en cuenta en la calificación la incorporación de un interfaz o menú amigable que facilite la experimentación o la presentación de resultados. También podrán recibir puntuación adicional otras mejoras o añadidos que se incorporen al trabajo más allá de los requisitos mínimos que se mencionan en la lista de objetivos específicos.

2.6.Presentación y defensa

El día de la defensa se deberá realizar una pequeña presentación (PDF, PowerPoint o similar) de 10-15 minutos en la que participarán activamente todos los miembros del grupo que ha desarrollado el trabajo. Esta presentación seguirá a grandes rasgos la misma estructura que el documento, pero se deberá hacer especial mención a los resultados obtenidos y al análisis crítico de los mismos. Se podrá usar un portátil (personal del alumno), diapositivas y/o pizarra. En los siguientes 10-15 minutos de la defensa, el profesor procederá a realizar preguntas sobre el trabajo, que podrán ser tanto del documento como del código fuente. Se podrán plantear cuestiones específicas a alguno de los miembros del grupo que se presenta el trabajo, y en caso de dudas razonables acerca de la autoría por parte de alguno de estos integrantes se podrá proponer una defensa separada.

3. Criterios de evaluación

Para que el trabajo pueda ser evaluado, se deberá satisfacer los objetivos concretos descritos en el apartado 1 (todos y cada uno de ellos, si no, el trabajo obtendrá una nota de suspenso). Uno de los alumnos del equipo deberá subir a través del formulario disponible en la página de la asignatura un fichero comprimido .zip, que contenga:

- **Una carpeta con el código fuente**. Dentro de dicha carpeta tiene que haber un fichero README.txt, que resuma la estructura del código fuente, e indique cómo usar la interfaz (si se ha implementado), o al menos cómo hacer pruebas con las funciones implementados, incluyendo ejemplos de uso. Asimismo se deberá indicar cómo reproducir los experimentos realizados. Es importante la coherencia de este fichero con la defensa.
- **El documento – artículo en formato PDF**. Deberá tener una extensión mínima de 8 páginas, y máxima de 10. Deberá incluir toda la bibliografía consultada (libros, artículos, technical reports, páginas web, códigos fuente, diapositivas, etc.) en el apartado de referencias, y mencionarlas a lo largo del documento.

Para la evaluación se tendrá en cuenta el siguiente criterio de valoración, considerando una nota máxima de 3 en total para el trabajo:

- **El diseño y desarrollo realizado (1.5 puntos)**: se valorará la claridad y buen estilo de programación, corrección y eficiencia de la implementación, y calidad de los comentarios. La claridad del fichero README.txt también se valorará. En ningún caso se evaluará un trabajo con código copiado directamente de internet o de otros compañeros. Se valorarán todas las aportaciones significativas sobre el trabajo básico.
- **El documento – artículo científico (1 punto)**:
  - Se valorará el estilo general del documento (por ejemplo, el uso de la plantilla sugerida).
  - Se valorará la cantidad y calidad de los experimentos, los resultados alcanzados y el análisis crítico que se haga de los mismos.
  - Se valorará la claridad de las explicaciones, el razonamiento de las decisiones, el análisis y presentación de resultados, y el uso del lenguaje. Igualmente, no se evaluará el trabajo si se detecta cualquier copia del contenido.
- **La presentación y defensa (0,5 puntos)**: se valorará la claridad de la presentación y la buena explicación de los contenidos del trabajo, así como, especialmente, las respuestas a las preguntas realizadas por el profesor.
- **Mejoras**: Se valorarán hasta con 0.5 puntos extra sin superar el máximo de 3 puntos totales del trabajo.

IMPORTANTE: Cualquier **plagio**, **compartición de código** o uso de material que no sea original y del que no se cite convenientemente la fuente, significará automáticamente la **calificación de cero** en la asignatura para **todos los alumnos involucrados**. Por tanto, a estos alumnos **no se les conserva**, ni para la actual ni para futuras convocatorias, **ninguna nota** que hubiesen obtenido hasta el momento. Todo ello sin perjuicio de las correspondientes **medidas disciplinarias** que se pudieran tomar.

4. Referencias

[1] Plantilla IEEE. https://www.ieee.org/conferences_events/conferences/publishing/templates.html