

Inteligencia Artificial (IS) 2017/18

Propuesta de trabajo

*Decidir jugadas de **TETRIS** mediante Búsqueda y Planificación*

Luis Valencia Cabrera

1. Introducción y objetivo

Los algoritmos de búsqueda y planificación constituyen un amplio área de estudio dentro de la *Inteligencia Artificial*, con múltiples aplicaciones desde la robótica hasta el diseño de estrategias ganadoras en juegos o en retos de todo tipo, desde lo más conceptual a la resolución práctica de problemas reales.

El ámbito de los juegos es de gran interés, no solo por su vertiente lúdica sino por otra serie de beneficios asociados. En particular, los juegos de lógica pueden promover (según el caso) el desarrollo de la agilidad mental, aspectos geométricos, reconocimiento de patrones, visión estratégica, etc.

El juego del Tetris se creó originalmente en los años 80 en la (ya extinta) Unión Soviética. En [1] puede consultar los hechos principales relacionados con su historia y no entraremos en detalle aquí. Desde entonces se han desarrollado múltiples versiones y variantes (una de las más populares en figura 1), y continúa suscitando interés unas cuantas décadas después de su creación.

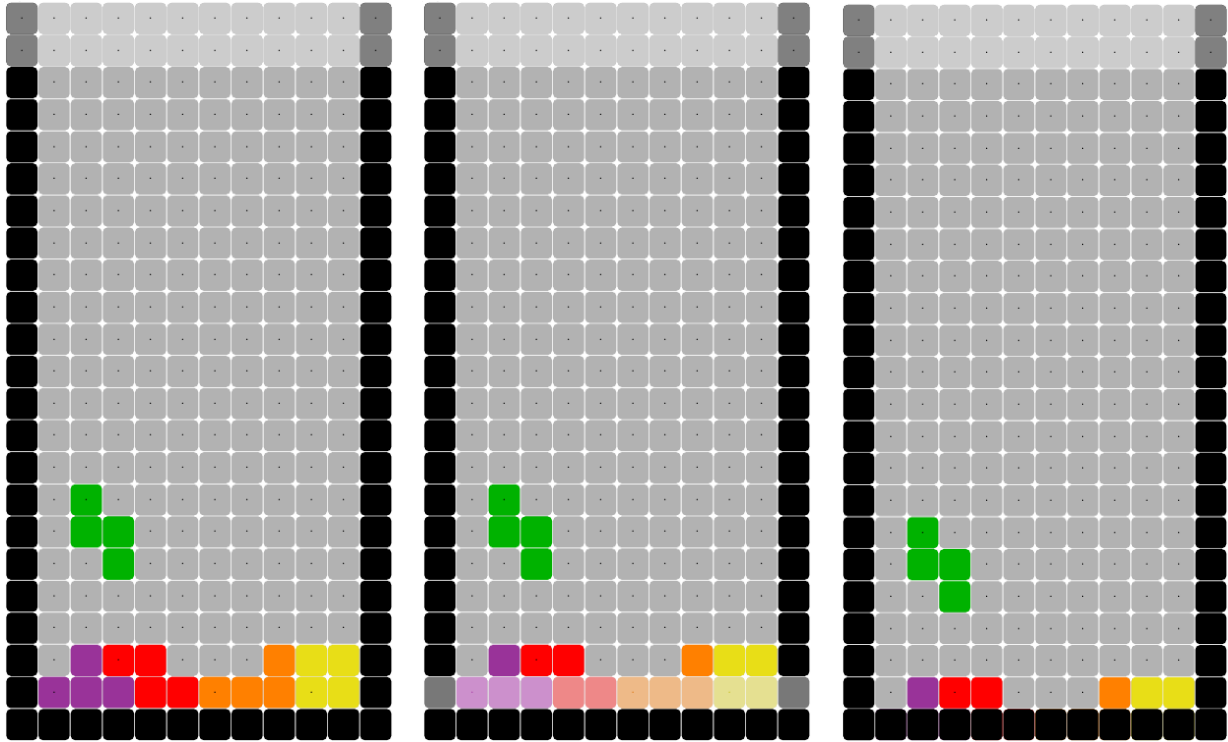


Figura 1: Pantalla inicial del Tetris de Atari Games.

El **objetivo principal** de este trabajo será el diseño y desarrollo en Python de un asistente para la decisión de la mejor jugada, dada una situación del juego, empleando las técnicas de representación y búsqueda y/o planificación vistas en clase (con las bibliotecas empleadas en las prácticas y bibliotecas generales de apoyo). Pasemos a detallar en qué consiste el juego y a desglosar los objetivos del trabajo.

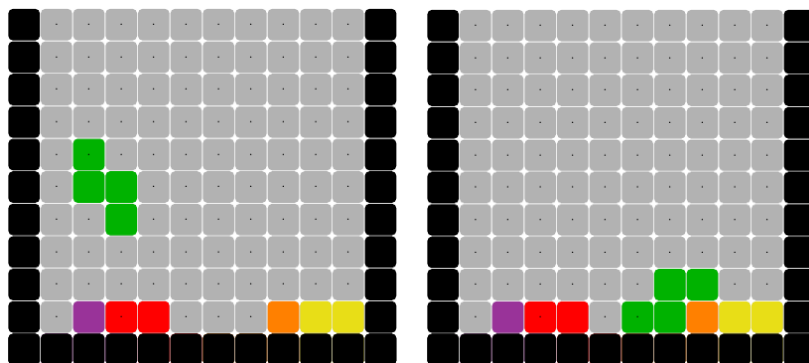
2. Descripción del juego

El **cometido principal** del jugador en el juego es la colocación adecuada de una serie de piezas que van cayendo sobre una pantalla vertical (generalmente de 20 filas por 10 columnas). Además, 2 filas adicionales se suelen mantener ocultas para que vayan apareciendo progresivamente las fichas por la parte superior de la pantalla (como aparecen sombreadas en la parte superior de la imagen). Las piezas caen hasta que lleguen a la base o se encuentren con algún obstáculo, sobre el que quedan posadas. Cada vez que se completa una fila, ésta desaparece, como vemos en estas dos primeras imágenes:



Esto deja más despejada la zona superior de la pantalla, de modo que tenemos más tiempo para situar las piezas donde corresponda, de modo que ocupen el menor espacio posible.

Las piezas irán cayendo una a una hasta asentarse sobre la primera casilla sobre la que colisionen, sin perder la forma de la pieza en cuestión. Durante la caída, el jugador podrá hacer movimientos de **rotación a derecha o izquierda**, así como **desplazamientos** también a **derecha o izquierda**, mientras baja la pieza por la propia gravedad hasta colocarse en el sitio deseado.



Piezas

Las piezas del juego se conocen como tetrominós o tetriminós, que viene de la palabra *tetraminó* [2], como fichas de dominó que constan de 4 fragmentos iguales (cuadrados unitarios). Existen 7 formas diferentes, con los nombres y formas definidas que vemos en la imagen:

Estados Rotación		0	R	2	L
Tipos Piezas	I				
	J				
	L				
	O				
	S				
	T				
	Z				

Su forma por defecto, digamos canónica, es la que aparece en la primera columna de figuras, y se denota por el estado de rotación cero (0). Las restantes resultan al llevar a cabo, respecto a la forma canónica, las rotaciones correspondientes a derecha (R), izquierda (L) o dos rotaciones consecutivas a derecha o izquierda (2). En todo caso, se debe cumplir que la realización de 4 giros a derecha o izquierda, en ausencia de la gravedad propia del descenso de las piezas en el juego, debe dar como resultado la misma forma original. Podemos colocarnos en cualquiera de las 4 columnas de arriba, elegir un sentido (derecha o izquierda) y desplazarnos 4 veces consecutivas de una columna a otra para ver el efecto.

Este es el funcionamiento básico de las rotaciones, pero en muchos casos podemos estar demasiado cerca de una pared o una pieza, lo que puede dar lugar a que no podamos realizar la rotación deseada. En su forma más simple, bastaría con no admitir movimientos que produzcan la colisión. No obstante, con carácter general, de acuerdo con las indicaciones de los reguladores del juego, existen técnicas para evitar los muros. Más concretamente, como se explica en detalle en [3], debemos considerar, según el tipo de pieza que se trate, entre uno y cinco desplazamientos respecto a las posiciones donde en principio debería quedar la figura si la rotáramos sin más de acuerdo con las figuras de arriba.

3. Desarrollo detallado del trabajo

El **objetivo principal** de esta propuesta es, como se indicó al inicio, *el diseño y desarrollo en Python de un asistente para la decisión de la mejor jugada, dada una situación del juego, empleando las técnicas de representación y búsqueda y/o planificación vistas en clase* (con las bibliotecas empleadas en las prácticas y bibliotecas generales de apoyo necesarias). Para ello será necesario alcanzar los siguientes objetivos **específicos**:

- 1) Dado un estado de la pantalla, obtener la **secuencia de movimientos para colocar la pieza desde su lugar actual hasta un lugar específico indicado** (por ejemplo, la secuencia para pasar de la imagen de la izquierda a la de la derecha al final de la página 2).
 - Escoger el formalismo que considere adecuado para resolver el problema, bien como **espacio de estados**, enfocando el problema como en la práctica 2, o bien como un **problema de planificación**, enfocándolo como en la práctica 3, haciendo uso de PDDL.
 - Representar el problema de acuerdo con la decisión anterior, incorporando todos los elementos necesarios como hemos visto en clase. *Debido a las restricciones temporales que tenemos, se recomienda comenzar con una versión simplificada del problema, escogiendo solamente algunas de las piezas y no considerando las diferentes tentativas del objeto al chocar con algún obstáculo (si no se puede realizar el movimiento sin desplazamiento posterior, no se intenta ninguno de los desplazamientos alternativos, y se considera el problema irresoluble sin más.)*
 - Incorporar elementos relacionados con el coste de aplicación de los operadores, y heurísticas que traten de mejorar la forma en que nos acercamos a la solución del problema.
 - Realizar pruebas conforme vaya desarrollando los puntos centrales y las sucesivas mejoras sobre la solución propuesta. *En caso de que tenga tiempo, vaya incorporando nuevas piezas, las tentativas para evitar obstáculos, etc.*
 - Documentar el trabajo en un fichero con formato de artículo científico (ver fichero adjunto, *plantilla-trabajo.doc*).
 - Realizar una presentación de los resultados obtenidos en la defensa del trabajo.
- 2) **Dada una situación de la pantalla y una pieza, decidir el lugar adecuado donde colocarla**, y llevar a cabo el movimiento correspondiente para hacerlo. *Partiría del apartado anterior, donde sabemos cómo llevar una pieza de un lugar a otro específico, pero ahora deberemos asistir a la decisión sobre cuál puede ser un lugar adecuado para hacerlo.*
 - Lleve a cabo un proceso similar al del apartado anterior para ir refinando el resultado que vaya obteniendo.
- 3) **Dada una partida**, dada por una **secuencia de piezas** determinada (que van cayendo desde una determinada columna de entrada de la pantalla) que se pase como parámetro, y, **a partir de la pantalla inicial vacía, determinar la secuencia de movimientos para ir colocando las piezas lo mejor posible**, de modo que se liberen lo más posible las filas que corresponda.
 - De nuevo, vaya haciendo uso de los apartados anteriores y vaya refinando sobre ellos.

Documentación

Se ha dejado una plantilla de ejemplo que se podría usar para documentar el trabajo (no es obligatorio su uso, pero puede usarla tal cuál o bien le puede ser de utilidad como referencia de aspectos que suelen ser interesantes cuando se publica un trabajo de desarrollo e investigación). Se trata del fichero *plantilla-trabajo.doc*, en el que se puede ver la estructura y el formato del artículo

correspondiente. Este formato, para *IEEE conference proceedings*, se detalla en su *guía para autores* [5], con plantillas para Word y Latex.

El artículo deberá tener una extensión entre 8 y 10 páginas. La estructura general del documento debería incluir una introducción al trabajo (objetivos principales), posible repaso de antecedentes (trabajo relacionado en la literatura), descripción de los métodos empleados, descripción del diseño y desarrollo realizado, experimentos llevados a cabo y análisis de resultados, y conclusiones.

Presentación y defensa

El día de la defensa se deberá **realizar una pequeña presentación de 10 minutos** (si el grupo se compone de dos alumnos, 5 minutos cada uno). En esta presentación seguirá la misma estructura que el documento, y se hará especial mención a los resultados obtenidos. Se podrá usar un portátil (personal del alumno), diapositivas y/o pizarra. En los siguientes 10 minutos de la defensa (20 en total), el profesor procederá a realizar preguntas sobre el trabajo, que podrán ser tanto del documento como del código fuente.

4. Criterios de evaluación

Para que el trabajo pueda ser evaluado, se deberá satisfacer los objetivos concretos descritos en el apartado. El alumno debe de entregar 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 el código desarrollado, con ejemplos, etc. Asimismo, se deberá de 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*. Además, deberá de incluir toda la bibliografía consultada (libros, artículos, *technical reports*, códigos fuente, diapositivas, etc.) en el apartado de referencias, y mencionarlás 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 código fuente (0,8 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.
- **El documento – artículo científico (1,5 puntos):**
 - Se valorará la calidad de los experimentos, el razonamiento de ellos, y los resultados alcanzados.
 - Se valorará la claridad de las explicaciones, el razonamiento de las decisiones, el análisis y presentación de resultados, y el uso del lenguaje. Se podrá realizar en castellano o inglés. Igualmente, no se evaluará el trabajo si se detecta cualquier copia del contenido.
- **La presentación y defensa (0,7 puntos):** se valorará la claridad de la presentación y la buena explicación de los contenidos del trabajo. También será importante responder correctamente a las preguntas realizadas por el profesor sobre cualquier aspecto del trabajo. Además, independientemente de la nota de este apartado, servirá para clarificar la autoría y

nivel de comprensión del trabajo por parte de cada uno de los integrantes del grupo.

Para que el trabajo pueda ser evaluado, se deben satisfacer los objetivos específicos indicados en los apartados anteriores: **el código debe ser original, estar correctamente implementado y funcionar perfectamente, los experimentos se deben de haber llevado a cabo y analizados razonadamente, el documento debe de ser completo y contener entre 8 y 10 páginas (no más, no menos), y se debe de realizar la defensa con una presentación de los resultados obtenidos.**

Cualquier **plagio, compartición de código** o uso de cualquier material que no sea original y del que no se cite convenientemente la fuente; que se detecte, 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.

5. Referencias

- [1] Tetris en Wikipedia. <https://es.wikipedia.org/wiki/Tetris>
- [2] Tetrominós. <https://es.wikipedia.org/wiki/Tetromin%C3%B3s>
- [3] Guía sobre el sistema de rotación. https://tetris.wiki/SRS#Wall_Kicks
- [4] S. Russel, P. Norvig. Inteligencia Artificial, un enfoque moderno. *Prentice Hall*, 2010.
- [5] Plantilla IEEE. https://www.ieee.org/conferences_events/conferences/publishing/templates.html