

## **Sistemas de Soporte para Celdas Producción Flexible**

### **Sistema Autónomo Inteligente**

Bogado Sebastián, García Marra Alejandro

**Abstract.** El objetivo del presente trabajo es la implementación de un sistema autónomo inteligente para la resolución del juego conocido como Angry Birds, detallando la implementación, decisiones de diseño y evolución.

## **1 Introducción**

Definimos un agente como aquel capaz de percibir su medioambiente con la ayuda de sensores, procesar tales percepciones y actuar en su entorno de manera racional utilizando actuadores. La racionalidad de la acción se refiere a la intención de maximizar el resultado esperado.

El aprendizaje del agente ocurrirá cuando no sólo actúe en base a las percepciones en un instante dado, sino que pueda procesar sus experiencias previas y compararlas con los sentidos actuales, buscando si ya existe una acción que optimice el resultado esperado. En caso de encontrar una teoría cuyas precondiciones sean acordes a las percepciones actuales, se tomará la acción por ella descrita y se evaluarán los resultados de haber tomado dicha acción. Estos resultados pasarán a formar parte de la base de conocimiento del agente como postcondiciones de una teoría nueva y podrán ser utilizadas para decidir sobre futuras acciones.

Las postcondiciones de una acción son también las precondiciones del siguiente evento, pues las percepciones del entorno para instantes consecutivos resultan iguales.

En el caso donde no se encuentre ninguna experiencia previa en la base de conocimiento del agente cuyas precondiciones sean acordes al sentido actual, se utilizará un mecanismo de “curiosidad”, donde el agente propone una acción en base a un cálculo aleatorio y procesa las postcondiciones, creando así una teoría nueva con las precondiciones iniciales.

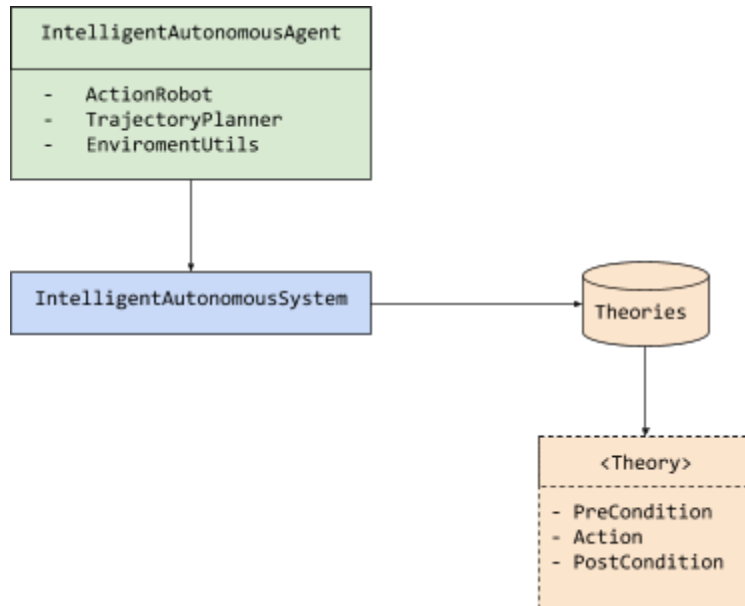
## **2 Descripción del Sistema**

Tomando el sistema base utilizado para la competencia de AIB [1], reemplazamos el agente naïve provisto, por uno propio capaz de aprender a modo de prueba, error y generando teorías. Nuestro agente intenta siempre tomar una decisión inteligente sobre el objetivo al cual pegarle. Para eso, modelamos

## 2.1 Arquitectura

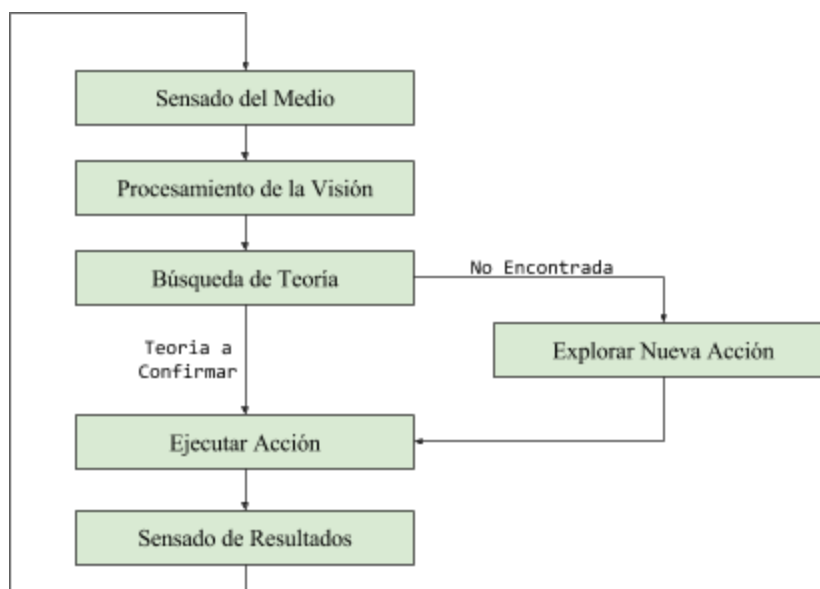
El sistema se compone por un único Agent, el cual interactúa con su entorno por medio del ActionRobot, encargado de sensado del medio y ejecución de acciones, y un TrajectoryPlanner el cual, junto con el EnviromentUtils, son utilizados para interpretar la Vision sensada por el ActionRobot y traducirla a elementos del dominio del agente.

Se cuenta también con un IntelligentAutonomusSystem, en quien recae la responsabilidad del aprendizaje al encargarse de la creación, comparación, selección y confirmación de las Teorías. Conociendo todas las Teorías de su base de conocimiento, tiene acceso a todas las Preconditions, Accions y Postconditions experimentadas por el Agent.



**Fig. 1.** Esquema básico de la arquitectura donde se pueden apreciar los principales componentes, encargados de sensar, procesar, decidir racionalmente y actuar sobre el entorno.

## 2.2 Flujo de Aprendizaje



**Fig. 2.** Flujo de aprendizaje recorrido por el Agent en un ciclo de vida. Se repite de manera continua hasta completar todas las iteraciones propuestas por todos los niveles encontrados.

## 2.3 Capacidades

En el modelado del entorno se considera una cantidad limitada de elementos influyentes en la toma de decisiones. Sobre el contexto del agente se pueden reconocer:

- Piso
- Colina
- Gomera
- Pájaros (Rojo, Amarillo, Azul, Negro, Blanco)
- Cerdo
- Hielo
- Madera
- Piedra
- TNT

Cada uno de estos elementos posee una semántica propia en el juego, con un conjunto de propiedades relevantes para la toma de decisiones cuando se tiene por objetivo ganar un nivel realizando la mayor cantidad de puntos posibles. El Agente debe ser capaz de interpretar la diferencia entre golpear el Piso y golpear una Madera, por medio de la evaluación de las postcondiciones luego de una serie de iteraciones. De la misma forma, con el correr de los niveles se presentan distintos tipos de Pájaros, los cuales experimentan reacciones distintas ante la misma acción (por ejemplo, un Pájaro explosivo golpeando un Hielo produce un efecto distinto al ocasionado por un Pájaro común).

Dentro de las restricciones del modelo, podemos considerar las acciones disponibles sobre las cuales se realiza la toma de decisiones. En el modelo actual, sólo se considera la posibilidad de Golpear en el centro el elemento más cercano a la Gomera en su lado izquierdo, variando únicamente el tipo del objetivo a golpear.

Si este modelo se extendiese para incorporar una mayor cantidad de acciones, como por ejemplo decidir a que altura se desea golpear al elemento, entonces crecería rápidamente la cantidad de Teorías que podrían probarse, permitiendo un mejor acercamiento a la decisión óptima.

## 3 Evolución del Sistema

A continuación se describirán tres hitos que cumplió sistema a lo largo del desarrollo.

### 3.1 Primera Muestra de Inteligencia

En esta primera etapa, el sistema trabajó constantemente sobre el primer nivel para que aprendiera lo básico del juego. Las teorías eran olvidadas cada vez que cerrábamos y abríamos el sistema porque no teníamos la certeza de que valieran la pena. Sin embargo, en un punto el sistema intentó varias cosas y encontró una forma determinista de ganar el primer nivel, descartando teorías que no hacían nada (es decir, no sumaban puntaje). Ver imágenes.

En un primer momento, su forma de describir el mundo era:

- cantidad de bloques de madera
- cantidad de bloques de hielo
- cantidad de bloques de piedra
- cantidad de cerditos

Y su única acción:

- elige un tipo de elemento aleatoriamente y de esos, pegarle al que está más a la izquierda

A continuación, los pasos que seguía para ganar el nivel

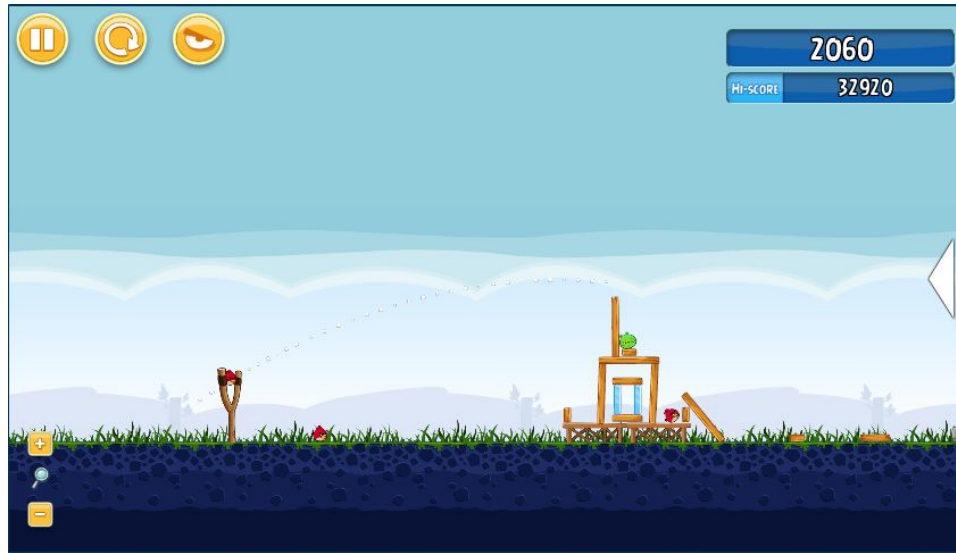


Fig. 3. Siempre le pega a la piedra primero.

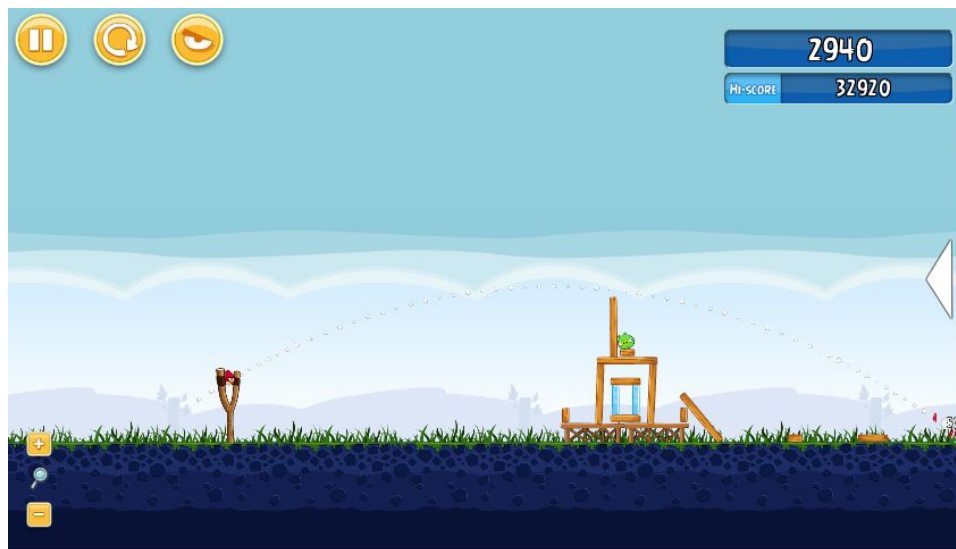


Fig. 4. Luego, a la piedra de vuelta para romperla.

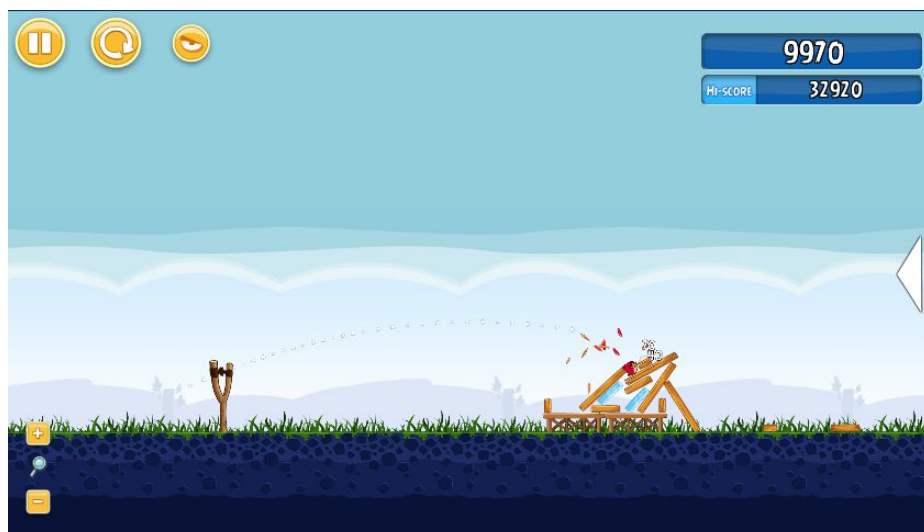


Fig. 5. Después le pega al cerdito.

Si bien el sistema aprendió a ganar el primer nivel, sabemos que no es la forma más eficiente. Así que decidimos que el sistema debería tener en cuenta el puntaje que generó cada teoría **para que la teoría de pegarle al cerdito rankee más alto.**

### 3.2 Encontrando la Teoría más Efectiva

La mejora mencionada anteriormente fue aplicada con éxito. El sistema, si bien al principio le pegaba a la piedra, aprendió que le tiene que pegar al cerdito porque da más puntos.

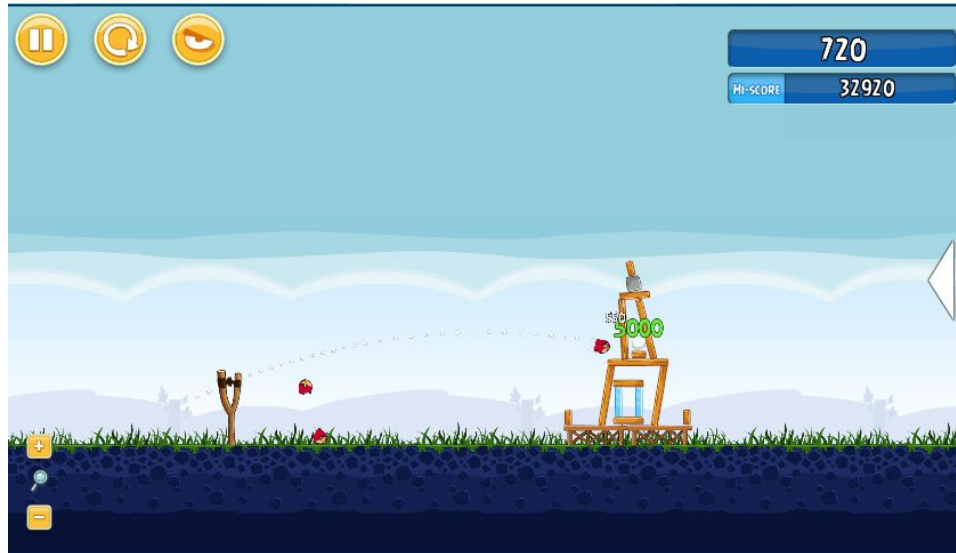


Fig. 6. El sistema aprendió que le tiene que pegar al cerdito como primer medida.

En este punto, el sistema ya mostraba algo de inteligencia, pero todavía tenía muchas limitaciones. Por más que pueda elegir correctamente entre una teoría y otra, una vez que encuentra una buena teoría, jamás intenta otras. Además, su visión del mundo es muy simple, no tiene en cuenta la posición de cada elemento, y muy estricta porque sólo tiene en cuenta cantidad exacta de elementos. Finalmente, el sistema está siempre probando en el primer nivel, su *training ground*. Al probarlo en los niveles siguientes, se encuentra en la situación de aprender todo de vuelta.

Con esto, decidimos que era momento de salir del primer nivel. Además, era estable la generación de teorías así que empezamos a persistirlas.

### 3.2 La Búsqueda de un Aprendizaje Global

Al enfrentarse a los siguientes niveles, ninguna teoría era reutilizable. Dejamos el sistema entrenando durante ocho horas, en los primeros diez niveles, iterando cincuenta veces por nivel para poder conseguir un buen conjunto de las mejores teorías para cada nivel.

Luego de generar esa base de conocimiento, hicimos que el sistema jugara normalmente, una vez por nivel. Al terminar el nivel, arma una teoría general tomando las cosas en común entre la dos mejores teorías de los últimos niveles, siempre y cuando tengan una acción similar.

Como ejemplo de esta generalización presentamos el siguiente caso:

- una teoría A tiene como precondition que haya 4 bloques de madera, y otra teoría B exige que haya 2 bloques de madera.  
La nueva teoría general C tendrá como precondition que haya al menos dos bloques de madera.  
De esta manera, la teoría C se aplicará en los mismos casos que las teorías A y B. Con las poscondiciones se aplica lo mismo.

Con esto, a partir de 325 teorías específicas para los niveles, el sistema generó 39 teorías generales. De esas, 34 implicaban pegarle a un cerdito, que es una noción esencial para ganar el juego. A continuación, una de las teorías reutilizables que generó el sistema:

- si hay
  - al menos un cerdito
  - al menos un bloque de piedra
  - al menos tres bloques de madera
- entonces, pegarle al cerdito
- luego, habrá
  - al menos un bloque de piedra
  - al menos tres bloques de madera

#### **4. Conclusiones**

En el presente trabajo, desarrollamos un sistema capaz de aprender a jugar al Angry Birds, teniendo como objetivo maximizar la cantidad de puntos obtenidos por nivel.

Uno de los requerimientos para considerar al sistema exitoso es que concluya que tiene que pegarle al cerdito, ya que esto resulta siempre en una cantidad de puntos superior al resto. Este comportamiento fue finalmente observado en los sistemas más entrenados, produciendo el efecto esperado en la suma de puntos. El proceso hasta conseguir esta optimización fue el siguiente: inicialmente, no tenía noción alguna del juego, probaba tiros aleatorios, y finalmente, entendió que debía pegarle al cerdito en la mayoría de los casos.

Hicimos foco en maximizar el puntaje, pero esto tiene una consecuencia que no pudimos prever: en algunos casos, genera teorías exitosas en su predicción, pero que generan poco puntaje, y no contribuyen a ganar el nivel. Dado que estas teorías cumplen las preconditiones y las postcondiciones, se siguen ejecutando constantemente y el sistema jamás intentará otra cosa en dicho nivel.

Para evitar esto, pensamos dos posibles mejoras:

- Generar teorías mutantes con otras heurísticas para encontrar teorías potencialmente mejores
- Agregar un componente de aleatoriedad tal que siempre, incluso cuando tiene teorías realmente buenas, exista la posibilidad de encontrar una mejor. Por ejemplo, cada diez acciones correctas, generar una teoría distinta a las actuales.

Para el desarrollo futuro y la continuación de este trabajo, proponemos

- Posibilitar más acciones distintas para mayor aleatoriedad, que da la posibilidad de encontrar mejores teorías

- Considerar al rankear teorías si la teoría fue utilizada para ganar un nivel para darle más importancia a las teorías que contribuyeron al ganar un nivel
- Considerar, en la visión del mundo de cada teoría, la posición entre los elementos dado que esto es importante en el juego, y permitiría conseguir teorías más reutilizables en futuros niveles
- Verificar que la acción elegida haya sido aplicada para no descartar teorías aparentemente erróneas cuando en realidad la acción no fue llevada a cabo. Es complejo por la forma en que está hecho el módulo de visión. Por ejemplo, si hay tres bloques iguales y uno se rompe y dos se mueven, no hay forma confirmar que efectivamente se le haya pegado al bloque que se le intentó pegar

## References

1. Angry Birds AI Competition, <http://aibirds.org/>
2. Agentes Inteligentes - [Apunte de clase](#)
3. The Autonomous Intelligent System, White Paper - [Fritz, García Márquez](#)
4. Sistemas Inteligentes Autónomos, Cap 4 - García Martínez, R., Servente, M. y Pasquini