

INTELIGENCIA ARTIFICIAL PARA JUEGOS

SESION 1

Dr. Edwin Villanueva Talavera

Contenido

- Revisión de IA
- IA en Juegos
- Agentes Inteligentes y Entornos

Bibliografía:

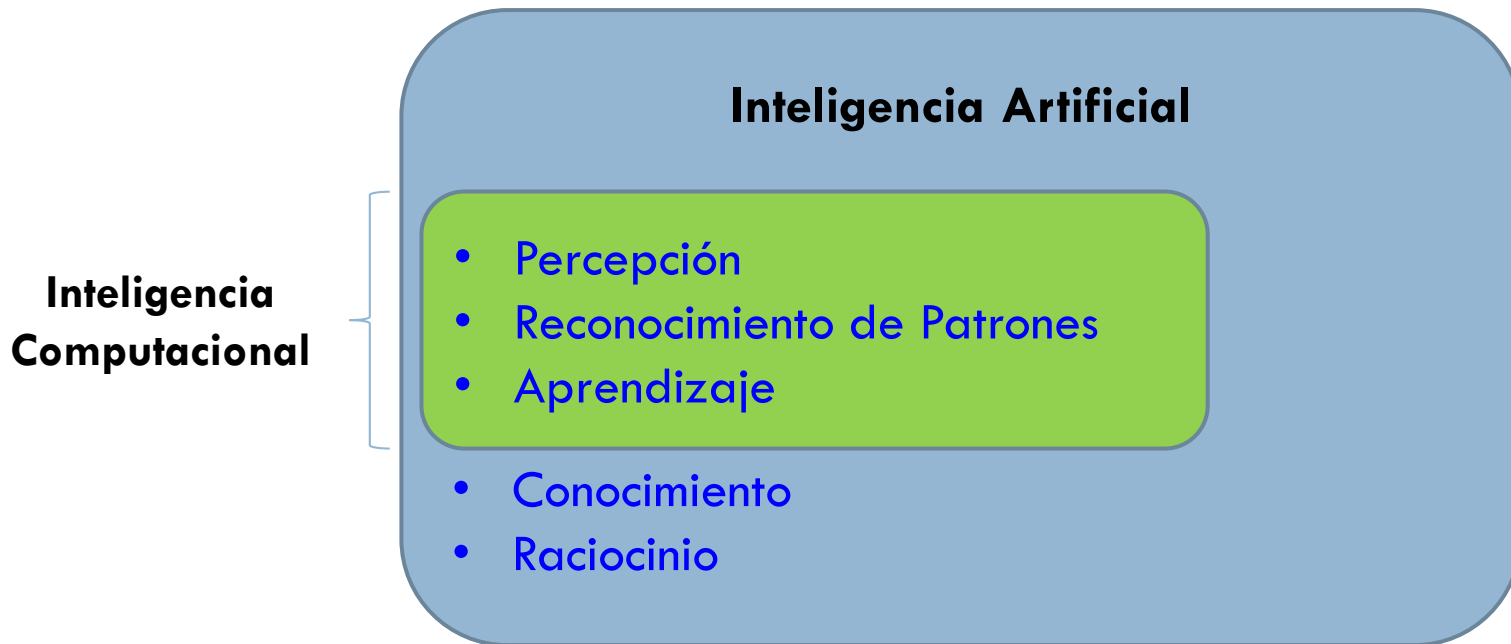
- Stuart Russell & Peter Norvig “[Artificial Intelligence: A modern Approach](#)”, Prentice Hall, Third Edition, 2010
- Georgios N. Yannakakis and Julian Togelius: [Artificial Intelligence and Games](#): Springer, 2018,

Definición de IA

- Área de las ciencias de la computación que estudia la automatización del comportamiento inteligente
- A diferencia de la filosofía, psicología y neurociencia, que intentan comprender el cerebro y el comportamiento inteligente, la IA busca construir entidades inteligentes

Definición de IA

Capacidades de una entidad inteligente



Enfoques de IA

Enfoques de estudio de IA:

	Imitación al humano	Racionalidad
Pensamiento	Sistemas que piensan como humanos	Sistemas que piensan racionalmente
Comportamiento	Sistemas que actúan como humanos	Sistemas que actúan racionalmente

Visión "test de Turing"

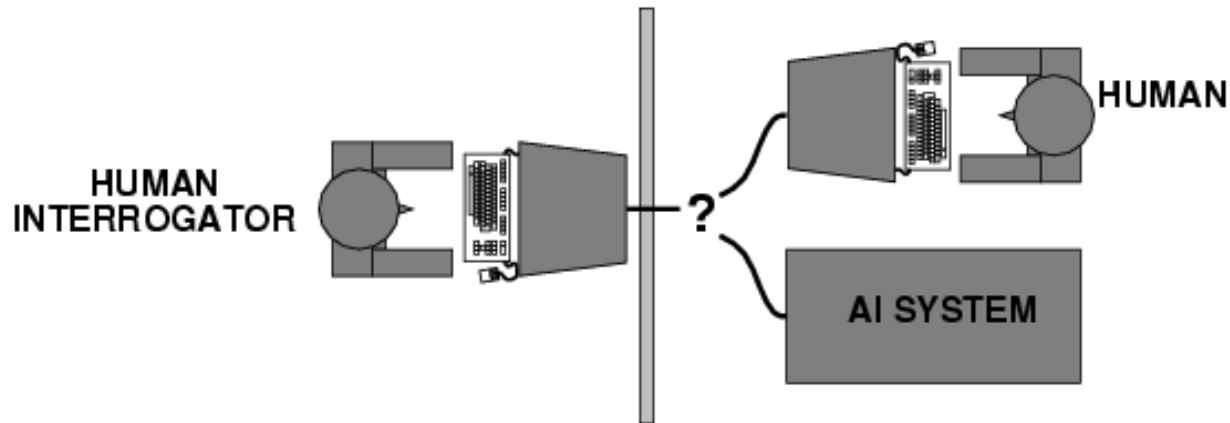
Visión del agente inteligente
(vision del curso)

Enfoques de IA

Sistemas que actúan como humanos: “Test de Turing”

- En 1950 Turing propuso el famoso “Test de Turing” como forma de definir inteligencia (artículo “Computing machinery and intelligence”):

Un sistema es inteligente si un interrogador humano, después de hacer algunas preguntas por escrito, no es capaz de distinguirlo de otro humano.



Enfoques de IA

Sistemas que actúan como humanos: “Test de Turing”

- Para pasar el “Test de Turing” se requiere las siguientes capacidades:
 - Procesamiento de lenguaje natural
 - Representación del conocimiento
 - Raciocinio automatizado
 - Aprendizaje de máquina
- La crítica del Test de Turing es que no está basada en principios básicos, sino en la imitación.
 - Sería similar a intentar construir un avión imitando el vuelo de un pájaro

Enfoques de IA

Sistemas que actúan racionalmente: “El agente racional”

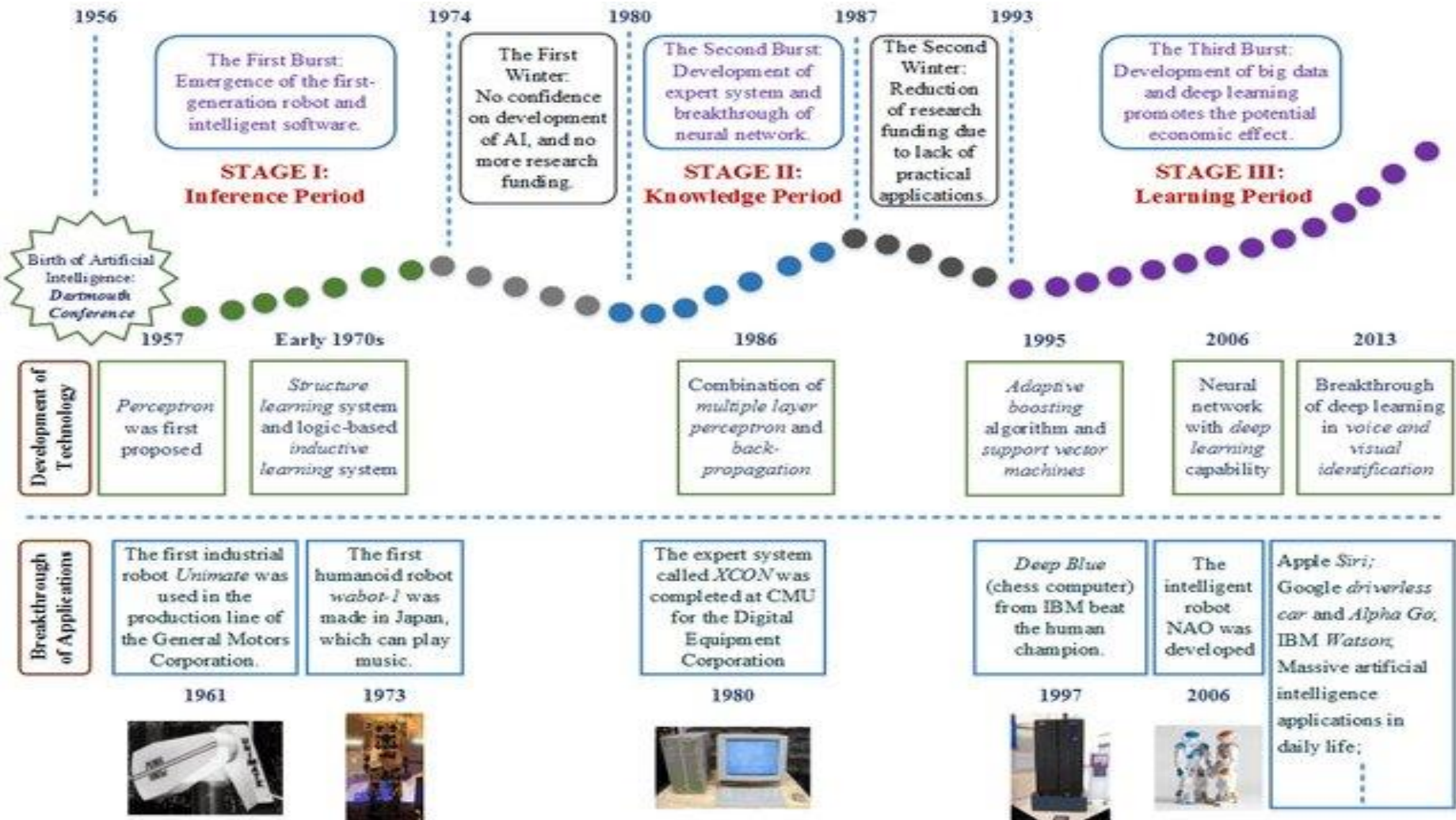
❑ **Comportamiento racional** = hacer lo que es esperado en el momento preciso para alcanzar los objetivos, dada la información disponible

❑ No necesariamente involucran raciocinios lógicos

❑ En este enfoque, una maquina inteligente ideal es un **agente racional**:

entidad computacional que percibe su entorno y actúa maximizando sus posibilidades de éxito en algún objetivo o tarea

Evolución histórica de la IA



Estado actual de la IA

Áreas de IA con investigación acelerada

- ☐ Deep learning
- ☐ Aprendizaje reforzado profundo
- ☐ Redes adversariales generativas
- ☐ Modelos Híbridos
- ☐ Aprendizaje de Maquina automático
- ☐ IA explicable

Estado actual de la IA

Etapa Actual: *Deep Reinforcement Learning*

❑ Volodymir Mnih *et al.*, Nature 02/2015

- Algoritmo Deep Q-learning utilizado con 49 juegos del Atari 2600
- Entrada: 84x84 píxeles y score del juego
- Nivel comparable al de un testador profesional de juegos ([mismo algoritmo, arquitectura y hiper-parámetros](#))

Estado actual de la IA

Etapa Actual: IBM Watson

- ❑ Sistema capaz de entender lenguaje natural y responder preguntas de una determinada área
 - Ganador del juego Jeopardy en 2011 (tuvo acceso a 200 millones de paginas de contenido estructurado y no estructurado, incluyendo Wikipedia)



- Desde el 2014 esta comercialmente disponible como sistema consejero.

Estado actual de la IA

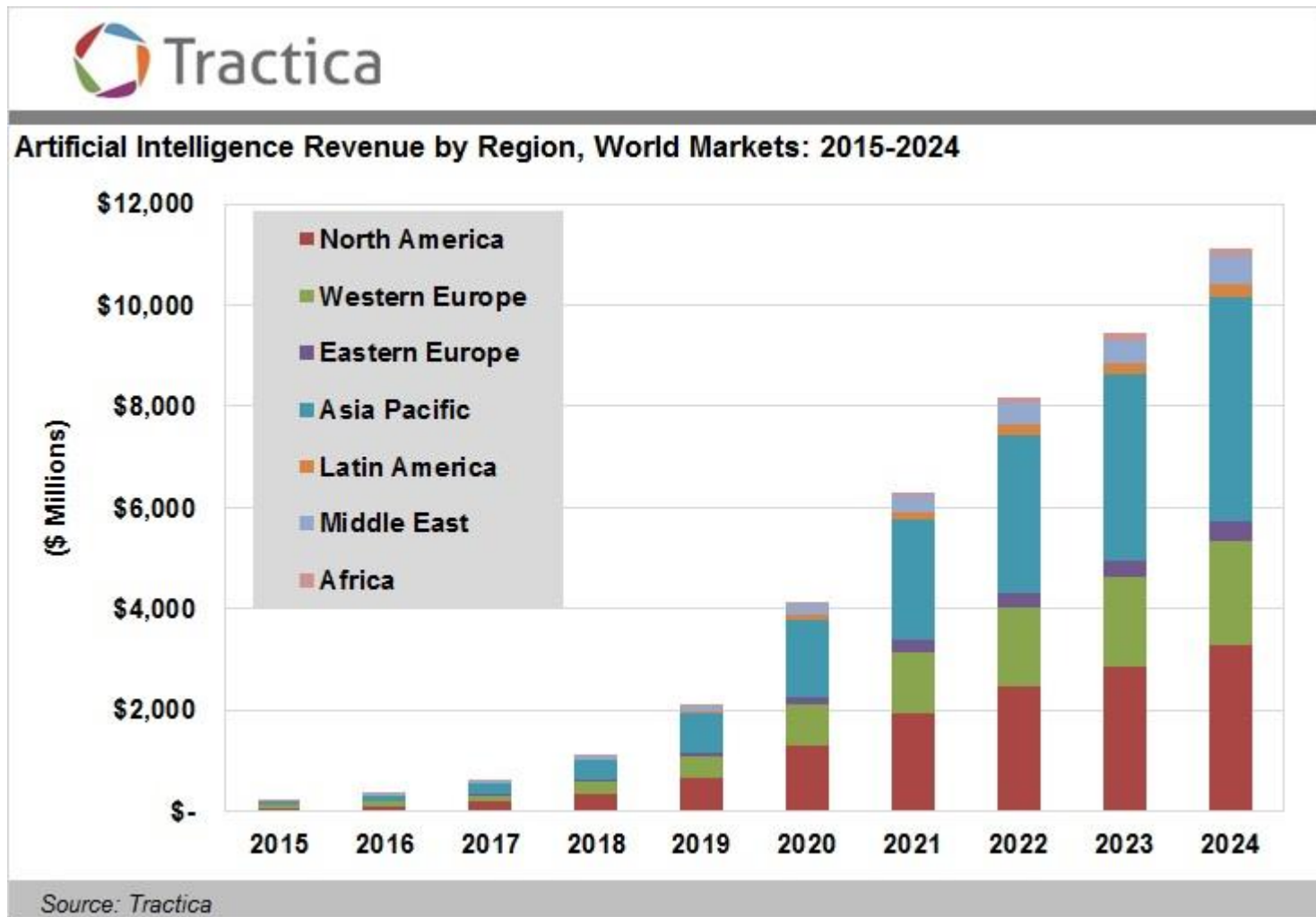
Etapas Actual: Pasamos el test de Turing?

- ❑ Algunos sistemas claman que ya pasamos
 - ❑ IBM Ask Watson
 - ❑ Chatbot 'Mitsuku', Cleverbot, etc



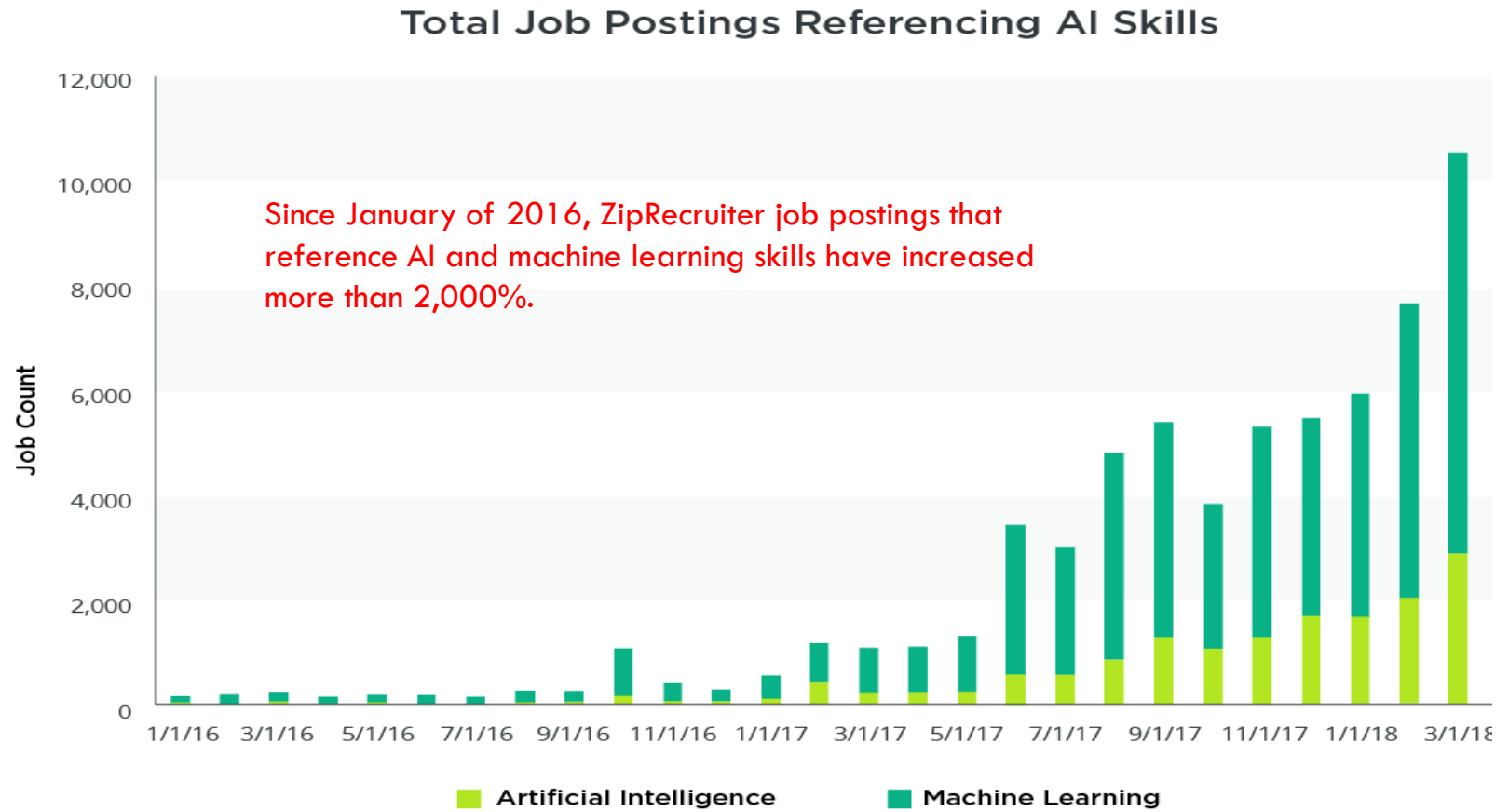
Futuro de la IA

Proyección de ganancias del mercado de IA en el mundo



Futuro de la IA

Proyección del mercado de trabajo en IA (USA, UK)



IA EN JUEGOS (GAME IA)



¿Por qué estudiar IA para juegos?

- ☐ Aprender IA usando entornos de juegos es estimulante y divertido
- ☐ Juegos son excelentes laboratorios para probar las capacidades de técnicas de IA
- ☐ Los grandes desafíos de IA son comúnmente encontrados en juegos
- ☐ La industria de Juegos necesita de la IA y viceversa

¿Que hacen los humanos con juegos?

- ☐ Jugarlos
- ☐ Desarrollarlos
- ☐ Crear contenido
- ☐ Aprender
- ☐ Distraerse
- ☐ Hacer dinero!



¿Por qué son relevantes los juegos?

- ❑ Entornos interesantes y desafiantes
- ❑ Enormes espacios de estados
- ❑ Son en tiempo real
- ❑ Juegos siempre han motivado grandes avances en IA (*Deep Blue, Kinect , Jeopardy, Chinook-Checkers, AlphaGo, ...*)



¿Por qué son relevantes los juegos?

- ☐ Son populares
- ☐ Pueden usarse para fines educativos, salud, capacitación, etc.
- ☐ Para entender las habilidades cognitivas y emocionales humanas
- ☐ Ricos en contenido audiovisual



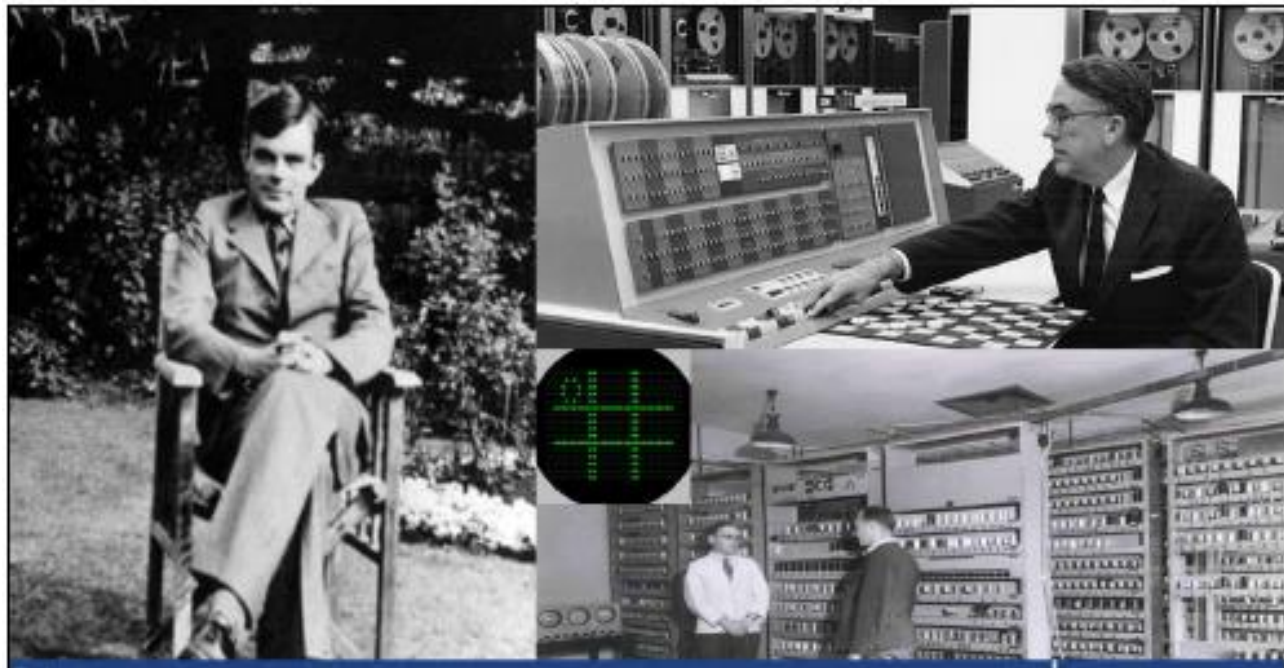
Áreas de IA en Juegos

- ☐ Procesamiento de señales
- ☐ Percepción (*speech recognition, computer vision*)
- ☐ Búsqueda y planificación
- ☐ Representación de conocimiento
- ☐ Razonamiento
- ☐ Procesamiento de lenguaje natural
- ☐ Machine Learning, etc

Historia de IA en juegos

Inicios:

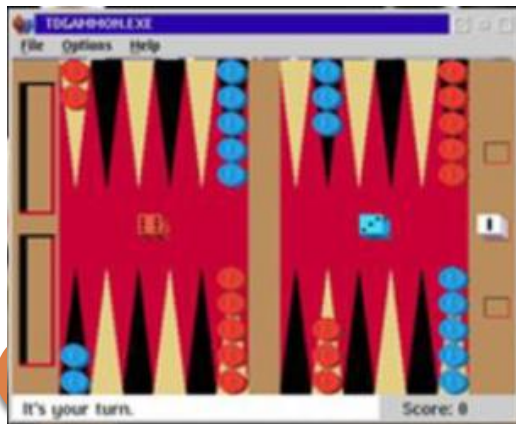
- ❑ 1951: Alan Turing crea el algoritmo Minimax y lo usó para jugar Chess
- ❑ 1952: Alexander Douglas desarrolla el primer software que domina el juego Tic-Tac-Toe
- ❑ 1959: Arthur Samuel ideó el primer algoritmo basado en aprendizaje por refuerzo que aprende a jugar Damas



Historia de IA en juegos

Hitos de IA en juegos de mesa:

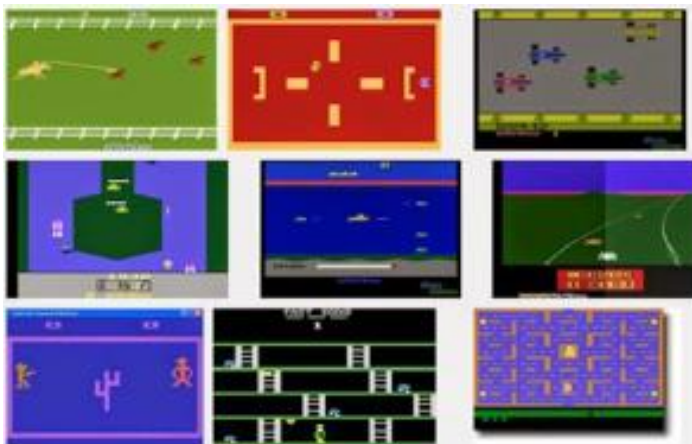
- ❑ **1992:** Se propone TD-Gammon para jugar backgammon, una red neuronal entrenada con diferencias temporales
- ❑ **1994:** Chinook, un método basado en búsqueda en árboles pudo ganar al campeón mundial de damas
- ❑ **1997:** El sistema Deep Blue de IBM (con algoritmo Minimax) le gana al campeón mundial de ajedrez Kasparov
- ❑ **2017:** AlphaGo (deep RL) gana al campeón mundial de Go



Historia de IA en juegos

Hitos de IA en videojuegos:

- ❑ **2012:** El Test de Turing de juegos fue pasado por dos bots de IA en el “Unreal Tournament 2004”. Se usó neuroevolución
- ❑ **2014:** Google DeepMind propone un algoritmo (DQN) que aprendió a jugar + de 40 juegos de Atari 2600 con desempeño super-humano solo desde la entrada de píxeles de la pantalla del juego
- ❑ **2017:** Ms Pac-Man is practically solved por el equipo Microsoft Maluuba usando una arquitectura híbrida de RL



Game-AI en la academia

Hitos de Game AI en la academia:

- ❑ **2001**: Fue el nacimiento del área Game-AI con el artículo de Laird & Lent “Human-level AI’s killer application: Interactive Computer games”
- ❑ **2005**: Nacen las conferencias IEEE CIG y AIIDE
- ❑ **2009**: IEEE Transactions on Computational Intelligence and AI Games
- ❑ **2018**: IEEE Transactions on Games



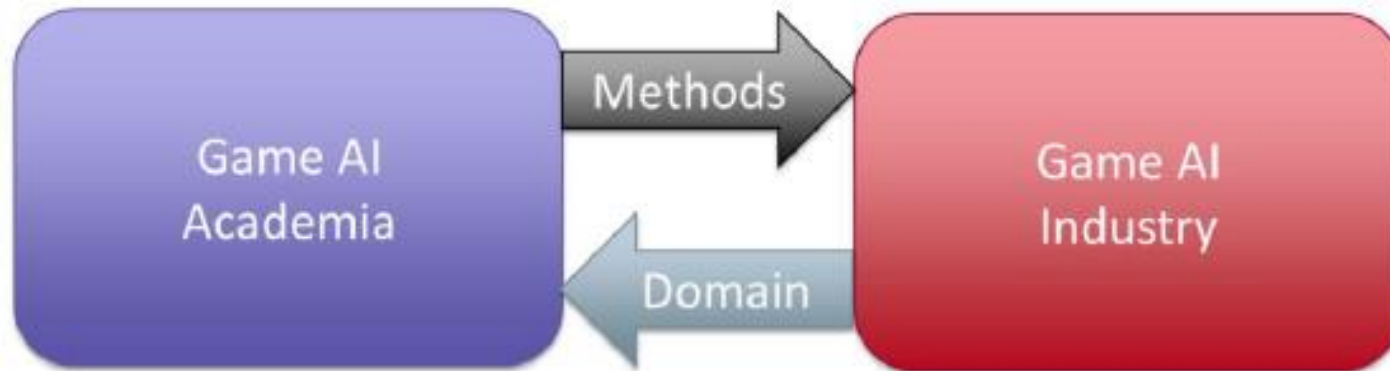
Game-AI en la academia

Dos principales área de investigación en Game IA:

- ❑ **Juegos como entornos para AI:** entre 2005 a 2012 el 54% de papers han sido en esta área relativos a agentes para controlar, rutear, tomar decisiones, etc. de NPCs
- ❑ **IA para mejorar juegos:** El 46 % de trabajos han sido en esta area para mejorar el diseño del juego o la experiencia del usuario



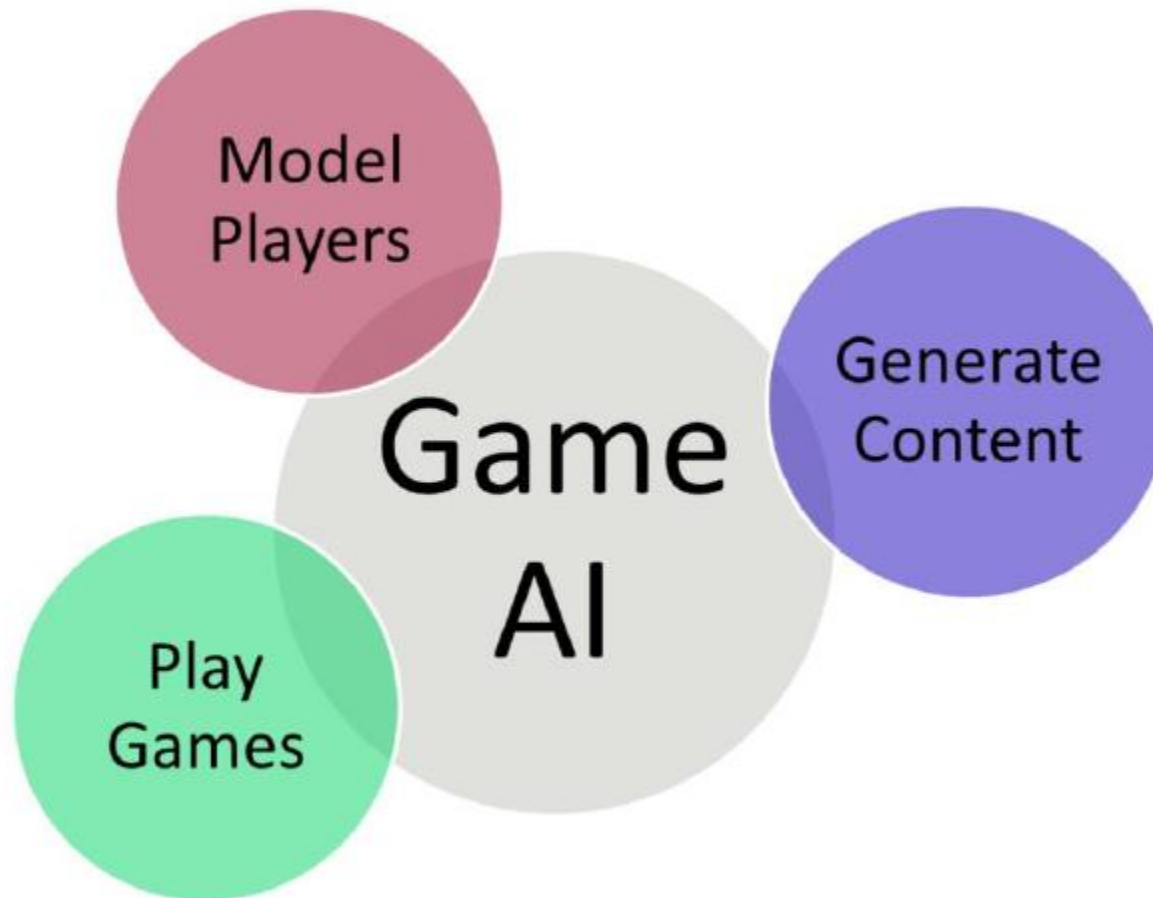
Interacción Academia-Industria en Game-AI



Gap Histórico:

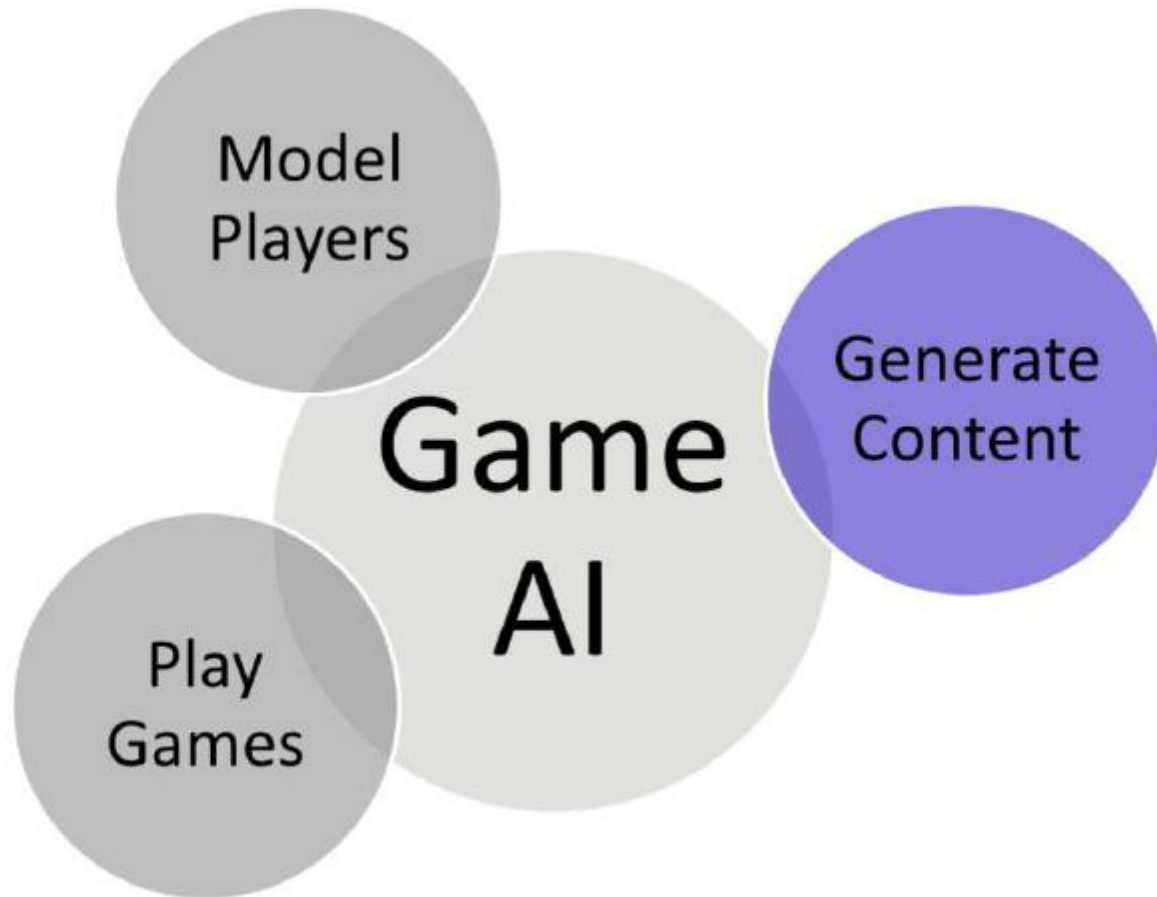
- ❑ Academia: “La industria no usa nuestras herramientas”
- ❑ Industry: “Academicos no tienen conocimiento del dominio e ignoran nuestros problemas”

Roles actuales de la IA en juegos



G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer Nature, 2018

IA para generación de contenido de juegos



G. N. Yannakakis and J. Togelius, **"Artificial Intelligence and Games,"** Springer Nature, 2018

IA para generación de contenido de juegos

- ❑ **Procedural Content Generation (PCG):** Producción automática o semiautomática de contenido de juego

- ❑ **Beneficios de PCG:**

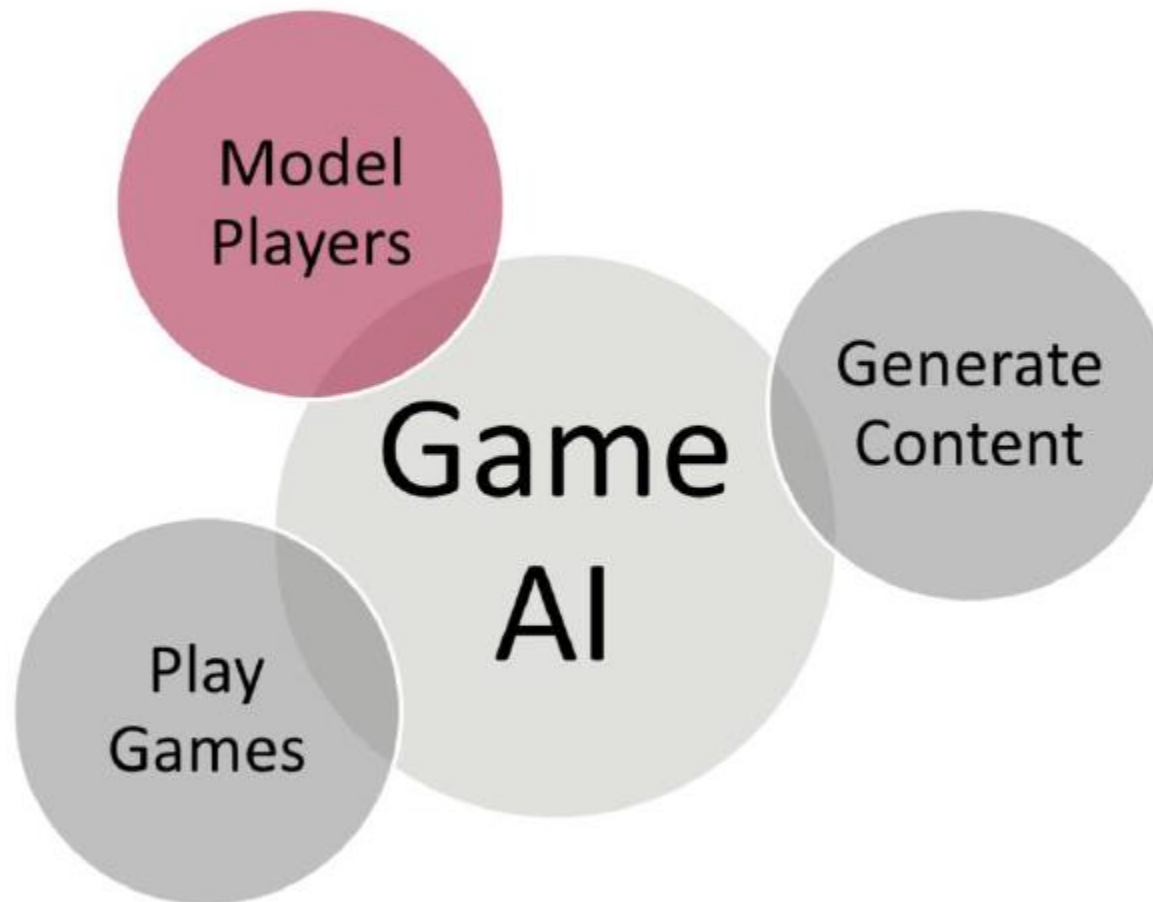
 - ❑ Disminuye costos de desarrollo

 - ❑ Posibilita juegos adaptables

 - ❑ Aumenta la re-jugabilidad

 - ❑ Abre camino a la creatividad mas alla del humano

IA para modelar jugadores

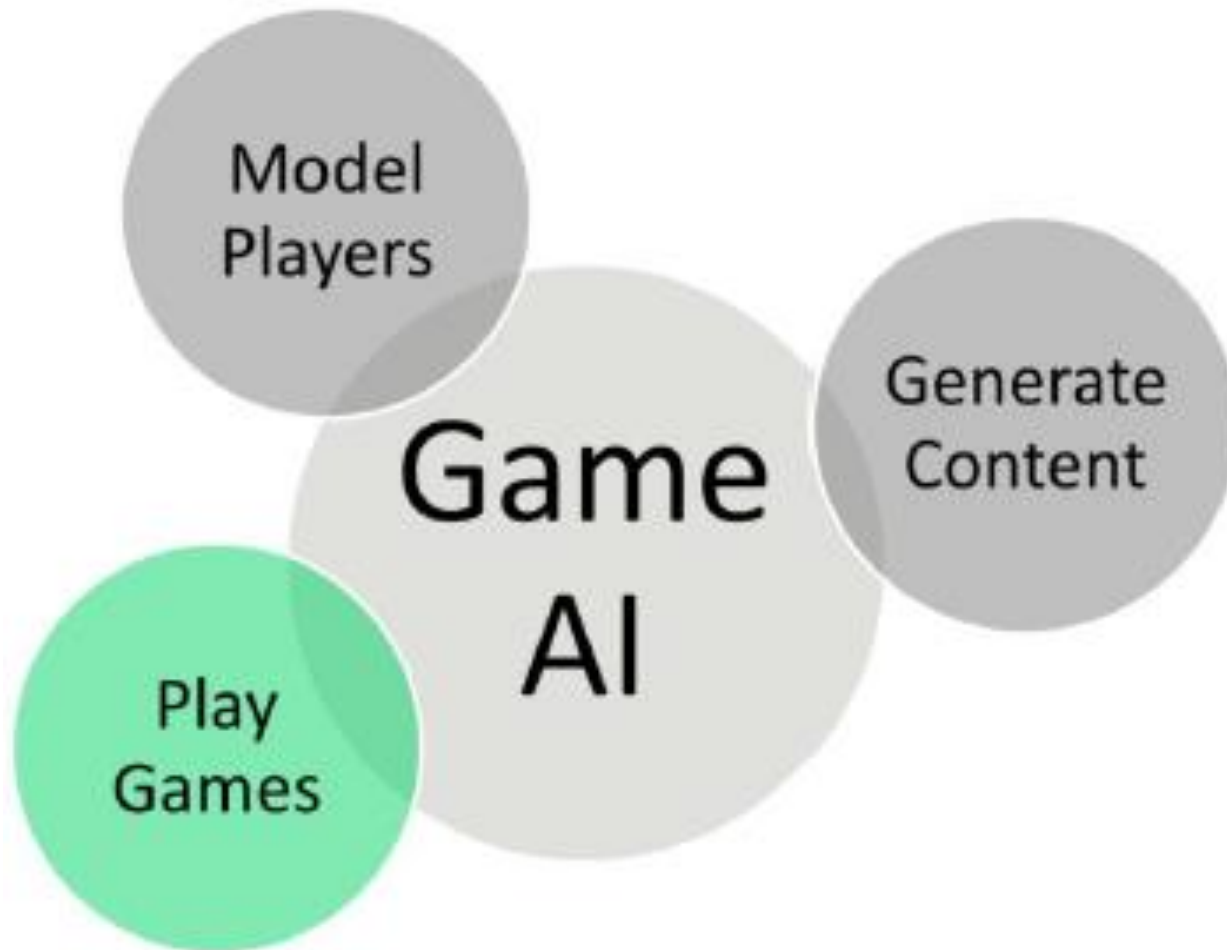


G. N. Yannakakis and J. Togelius, **"Artificial Intelligence and Games,"** Springer Nature, 2018

IA para modelar jugadores

- ❑ **“Games cannot be dissociated from emotion and learning...”. G.N. Yannakakis and A. Paiva “Emotion in Games”, in Oxford Handbook of Affective Computing, 2014**
- ❑ **¿Cuál es la sociación entre juegos, emociones y aprendizaje?**
 - ❑ Los juegos desafían nuestras capacidades cognitivas - aprendemos
 - ❑ Al jugar experimentamos emociones pos/neg, y de forma voluntaria!
 - ❑ Podemos controlar la experiencia (diferente de TV, películas, etc.)
- ❑ **¿Por que es importante modelar al jugador?**
 - ❑ Jugadores tienen comportamientos diferentes
 - ❑ Jugadores cambian con el tiempo
 - ❑ Tener un modelo del jugador es el **santo grail** para desarrollar juegos con gran potencial de re-jugabilidad

IA para Jugar (enfoque del curso)



¿Para qué crear agentes de IA que juegan?

❑ Dos dimensiones para responder:

- ❑ Se quiere un agente que juegue para ganar o para experimentar?
- ❑ Se quiere IA que controle al jugador (*player*) o a un NPC (non-player character)?

	Player	Non-Player
Win	<p>Motivation Games as AI testbeds, AI that challenges players, Simulation-based testing</p> <p>Examples Board Game AI (TD-Gammon, Chinook, Deep Blue, AlphaGo, Libratus), Jeopardy! (Watson), StarCraft</p>	<p>Motivation Playing roles that humans would not (want to) play, Game balancing</p> <p>Examples Rubber banding</p>
Experience	<p>Motivation Simulation-based testing, Game demonstrations</p> <p>Examples Game Turing Tests (2kBot Prize/Mario), Persona Modelling</p>	<p>Motivation Believable and human-like agents</p> <p>Examples AI that: acts as an adversary, provides assistance, is emotively expressive, tells a story, ...</p>

Qué tipos de IA se usan en agentes jugadores

- ☐ Random

- ☐ Comportamiento pre-especificado

- ☐ Búsqueda:

 - ☐ Sin informacion

 - ☐ Com informacion (A^*)

 - ☐ Adversarial

- ☐ Aprendizaje supervisado

- ☐ Aprendizaje por refuerzo

 - ☐ ADP

 - ☐ Diferencias temporales (Qlearning)

Se estudiaran en el curso

AGENTES INTELIGENTES Y ENTORNOS



Contenido

- Agentes Inteligentes
- Ambiente de Trabajo (PEAS)

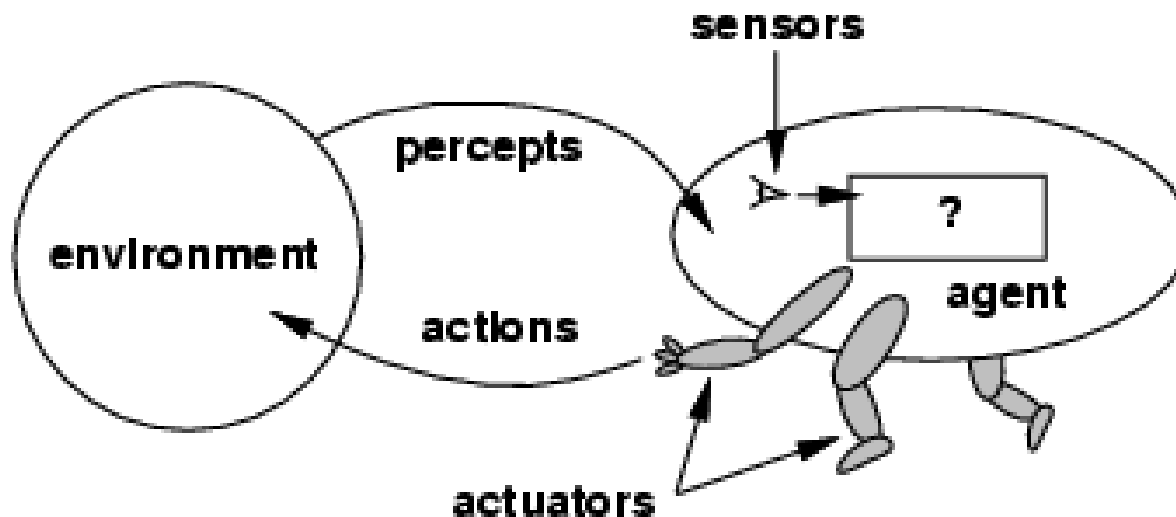
Bibliografía:

Capitulo 2.1 y 2.2 del libro:

Stuart Russell & Peter Norvig “[Artificial Intelligence: A modern Approach](#)”,
Prentice Hall, Third Edition, 2010

Agentes

- Un **agente** es algo capaz de percibir su **ambiente** por medio de **sensores** e de actuar sobre ese ambiente por medio de **actuadores**.



Agentes

Ejemplos:

□ Agente humano

- ▣ Sensores: Ojos, oídos y otros órganos.
- ▣ Actuadores: Manos, piernas, boca y otras partes del cuerpo.

□ Agente robótico

- ▣ Sensores: cámaras y detectores de infrarrojo.
- ▣ Actuadores: varios motores.

□ Agente de software

- ▣ Sensores: entrada por teclado, contenido de archivos y paquetes de red.
- ▣ Actuadores: pantalla, disco, envío de paquetes por la red.



Agentes

- El comportamiento del agente puede ser abstraído matemáticamente por la **función del agente**:

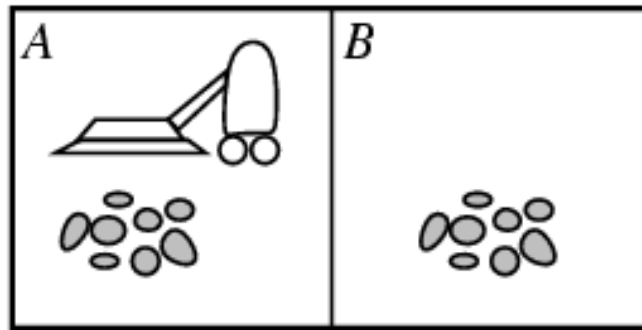
$$[f: P^* \rightarrow \mathcal{A}]$$

donde:

- P^* es una **secuencia de percepciones** (historia de todo lo que el agente percibió hasta el momento)
- \mathcal{A} es una acción.
- El **programa del agente** se ejecuta en una arquitectura física para producir f .
- Agente = arquitectura + programa.

Agentes

Ejemplo: Aspiradora en 2 posiciones



- Percepciones: ubicación y estado
 - ▣ Ejemplo: [A, sucio]
- Acciones: Izquierda, Derecha, Aspirar, No hacer nada

Agentes

Ejemplo: Aspiradora en 2 posiciones

Función de agente:

Secuencia de Percepciones	Acción
[A, Limpio]	Derecha
[A, Sucio]	Aspirar
[B, Limpio]	Izquierda
[B, Sucio]	Aspirar
[A, Limpio], [A, Limpio]	Derecha
[A, Limpio], [A, Sucio]	Aspirar
...	
[A, Limpio], [B, Limpio], [A, Limpio]	Derecha
[A, Limpio], [A, Limpio], [A, Sucio]	Aspirar
...	

Programa de agente:

Si la posición actual está sucia, entonces Aspirar, caso contrario moverse al otro lado.

Agentes Racionales

- Como llenar correctamente la tabla de acciones del agente para cada situación?
- Un **agente racional** toma la acción “conveniente” basado en aquello que él percibe que le dará éxito.
 - La noción de éxito es capturada por una **medida de desempeño**.
 - Ejemplos: cantidad de suciedad aspirada, gasto de energía, gasto de tiempo, cantidad de ruido generado, etc.
 - La medida de desempeño debe reflejar el resultado deseado.

Agentes Racionales

- **Agente racional:** es aquel que para cada secuencia de percepciones posible selecciona una acción que se espera venga a maximizar su medida de desempeño, dada la secuencia de percepciones y algún conocimiento interno incorporado al diseñar el agente.
- **Ejercicio:** para que medida de desempeño el agente aspirador es racional?

Agentes Racionales

- Racionalidad es diferente de perfección.
 - ▣ La racionalidad maximiza el desempeño esperado, mientras la perfección maximiza el desempeño real.
 - ▣ Las acciones de un agente racional solo dependen de las percepciones obtenidas hasta el momento.
- Los agentes pueden (y deben!) ejecutar acciones para coleccionar informaciones
 - ▣ Un tipo importante de colecciona de informacion es la exploracion de un ambiente desconocido.

Agentes Racionales

- El agente también puede (y debe!) **aprender**, osea, modificar su comportamiento dependiendo de lo que él percibe a lo largo del tiempo.
 - ▣ En este caso un agente es llamado de **autónomo**.
 - ▣ Un agente que aprende se adapta a una variedad de ambientes.

Ambiente de Trabajo - PEAS

- Los agentes racionales actúan sobre ambientes de trabajo (problemas).
- Al crear un agente, la primera etapa debe ser siempre especificar el **ambiente de trabajo (PEAS)**.
 - **P**erformance = Medida de desempeño
 - **E**nvironment = Ambiente
 - **A**ctuators = Actuadores
 - **S**ensors = Sensores

Ambiente de Trabajo - PEAS

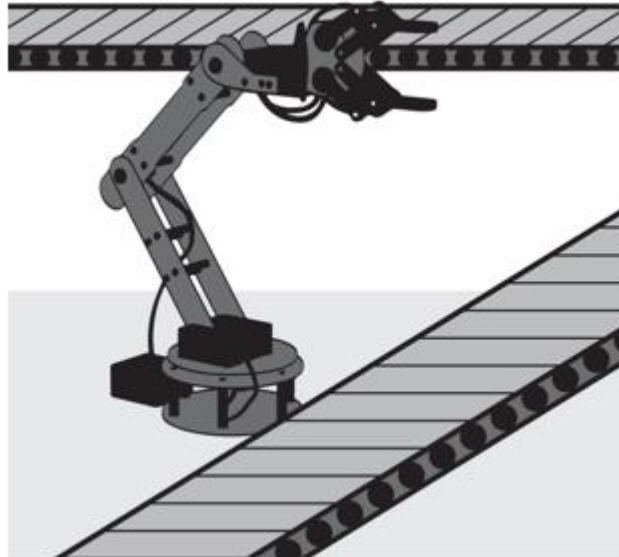
Ejemplo: Chofer de un taxi automatizado



- **Medida de desempeño:** viaje seguro, rápido, sin infracciones a las leyes de tránsito, confortable para los pasajeros, maximizando los lucros.
- **Ambiente:** avenidas, pasajes, otros vehículos, peatones, pasajeros.
- **Actuadores:** dirección, acelerador, freno, embriague, bocina, palanca de cambio, luces de señalización.
- **Sensores:** cámara, espejos, sonar, velocímetro, GPS, tacómetro, hodómetro, acelerómetro, sensores del motor (parámetros del motor).

Ambiente de Trabajo - PEAS

Ejemplo: Robot que selecciona piezas defectuosas



- ❑ **Medida de desempeño:** porcentaje de piezas defectuosas correctamente detectadas.
- ❑ **Ambiente:** correa transportadora con piezas; bandejas.
- ❑ **Actuadores:** brazo robótico.
- ❑ **Sensores:** cámara, sensores articulados.

Propiedades del Ambiente de Trabajo

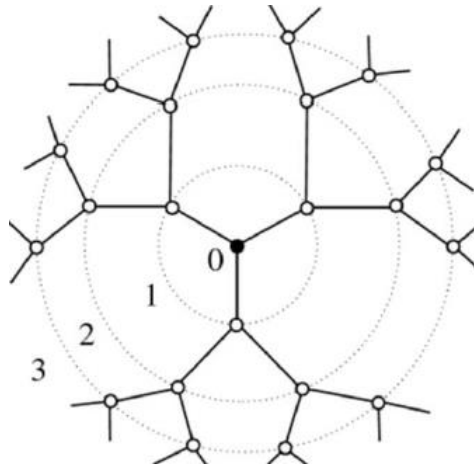
- **Completamente observable** (vs. **Parcialmente Observable**)
 - ▣ Los sensores del agente dan acceso al estado completo del ambiente en cada instante.
 - ▣ Todos los aspectos relevantes del ambiente son accesibles.
- **Determinístico** (vs. **Estocástico**)
 - ▣ El próximo estado del ambiente es completamente determinado por el estado actual y por la acción ejecutada por el agente.
- **Episódico** (vs. **Secuencial**)
 - ▣ La experiencia del agente puede ser dividida en episodios atómicos (percepción-ejecución de una única acción).
 - ▣ La elección de la acción en cada episodio solo depende del propio episodio.

Propiedades del Ambiente de Trabajo

- Estático (vs. dinâmico)
 - ▣ El ambiente no cambia mientras el agente delibera.
 - ▣ El ambiente es **semi-dinámico** si este no cambia con el paso del tiempo, pero el desempeño del agente si lo hace.
- Discreto (vs. contínuo)
 - ▣ Un número limitado y definido de percepciones y acciones.
- Agente único (vs. multi-agente)
 - ▣ Un único agente opera solo en el ambiente
 - ▣ En ambientes multi-agentes podemos tener
 - Multi-agentes cooperativos
 - Multi-agentes competitivos

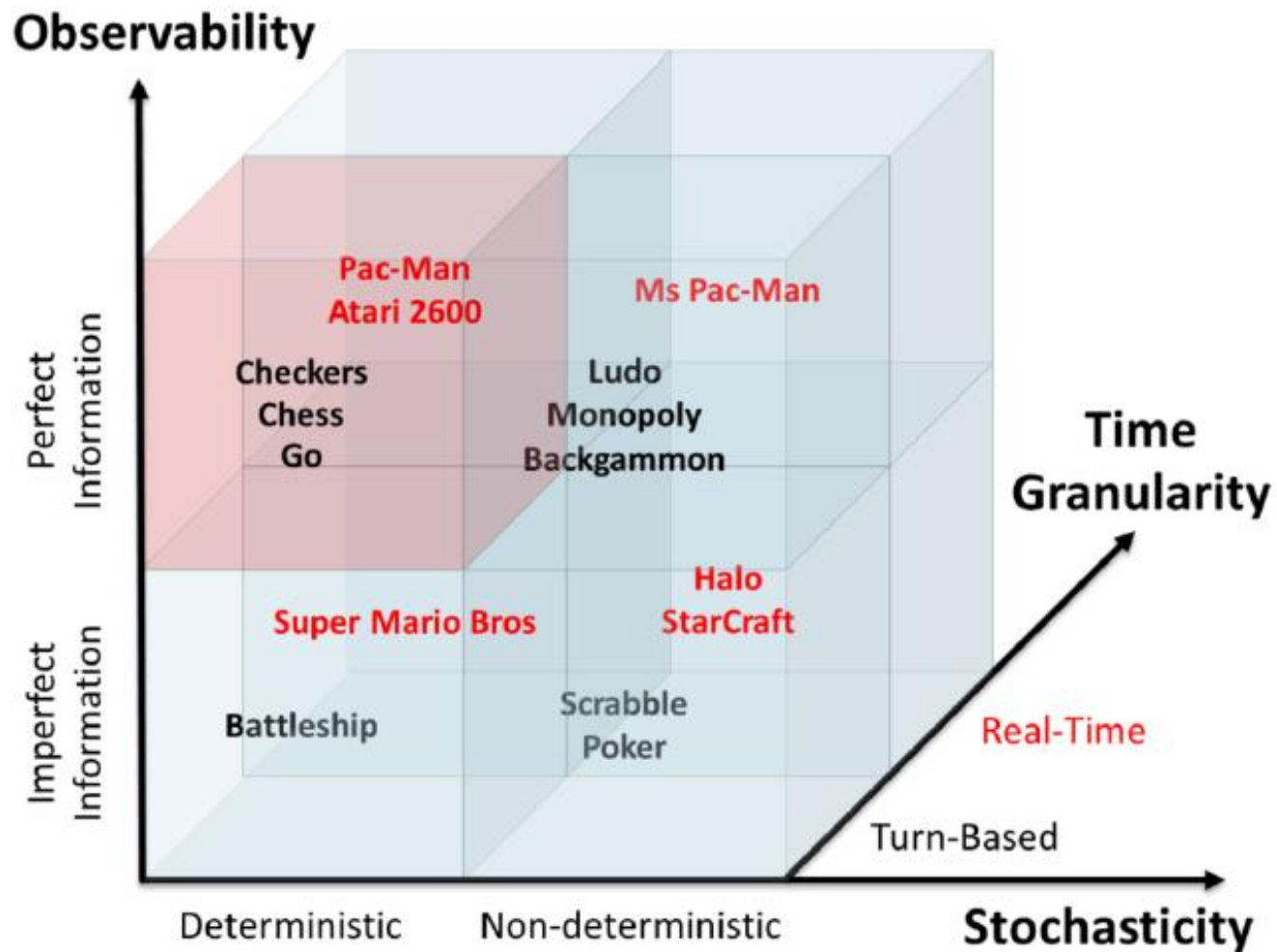
Propiedades Específicas de Ambiente de Juegos

- **Numero de jugadores:** 1 player / 2 players / multiplayer
- **Espacio de acciones y factor de ramificación**



- **Granularidad de tiempo:**
 - Por turnos (ply)
 - Tiempo real (ticks)

Propiedades Específicas de Ambiente de Juegos

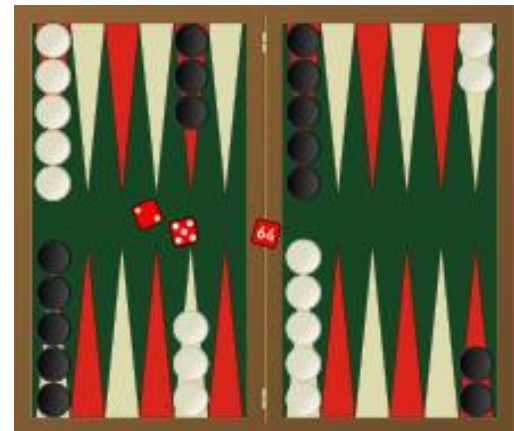


Ejemplos de Ambiente de Trabajo - Juegos

- **Ajedrez:** 2 jugadores, adversarial, determinístico, completamente observable, bf ~ 35 , ~ 70 turnos
- **Go:** 2 jugadores, adversarial, determinístico, completamente observable, bf ~ 350 , ~ 150 turnos
- **Backgammon:** 2 jugadores, adversarial, estocástico, completamente observable, bf ~ 250 , ~ 55 turnos



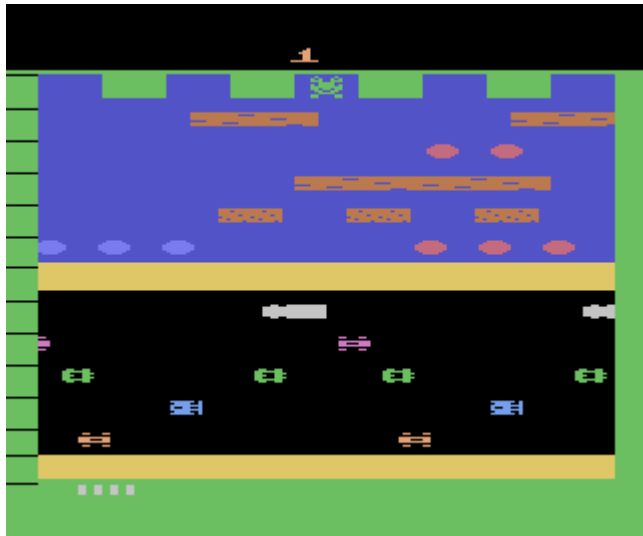
Tablero de Go



Tablero de Backgammon

Ejemplos de Ambiente de Trabajo - Juegos

- **Frogger** (Atari 2600): 1 jugador, determinístico, totalmente observable, $bf=6$
- **Montezuma's revenge** (Atari 2600): 1 jugador, determinístico, totalmente observable, $bf=6$



Pantalla de juego Frogger



Pantalla de juego Montezuma

Ejemplos de Ambiente de Trabajo - Juegos

- **StarCraft**: 2-4 jugadores, estocástico, parcialmente observable, bf>millon



Estructura de Agentes

- Un agente queda completamente especificado si se define la **función del agente** (mapeamiento de secuencias de percepciones en acciones)
- En la práctica es difícil especificar explícitamente la función del agente
- Al diseñar agentes se busca entonces encontrar una forma de representar concisamente la función racional del agente, esto es, se busca implementar un **programa de agente** para una arquitectura dada

Agente = arquitectura + programa

Estructura de Agentes

Agente Dirigido por Tabla

```
Function TABLE_DRIVEN_AGENT(percept) return action
```

Variables estáticas:

- *percepts*, una secuencia, inicialmente vacía
- *table*, tabla de acciones, indexada por secuencias de percepciones, de inicio completamente especificada

```
append percept to the end of percepts
```

```
action ← LOOKUP(percepts, table)
```

```
return action
```

- Desventajas:
 - Tabla gigante (ajedrez = 10^{150} entradas)
 - Mucho tiempo para construir la tabla
 - No tiene autonomía
 - Aunque use aprendizaje automático, tardaría mucho en aprender la tabla.

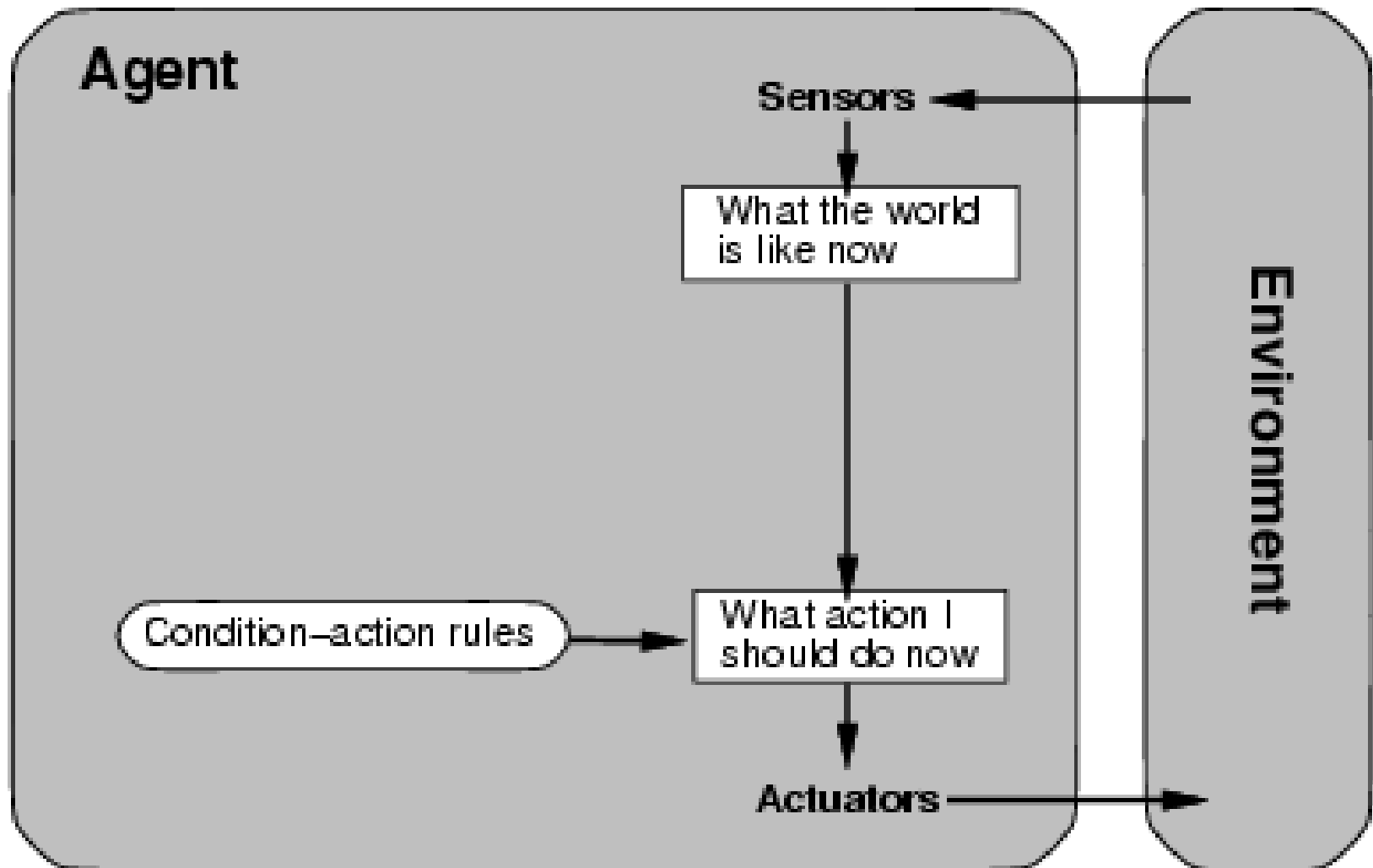
Estructura de Agentes

Tipos básicos de agentes

- Agentes reactivos simples
- Agentes reactivos basados en modelos
- Agentes basados en objetivos
- Agentes basados en utilidad

Estructura de Agentes

Agente Reactivo Simple



Estructura de Agentes

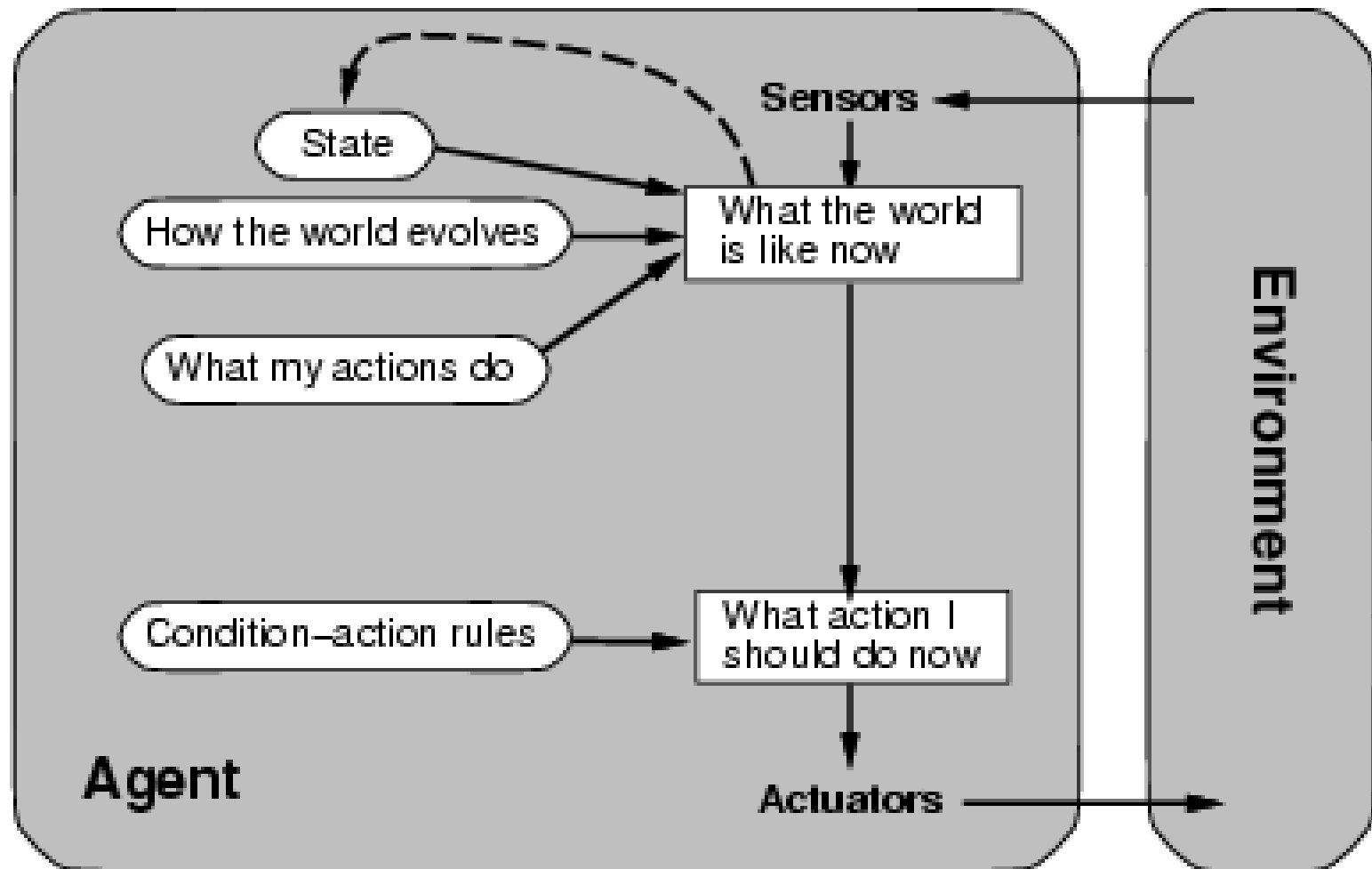
Programa del Agente Reactivo Simple

```
function SIMPLE-REFLEX-AGENT(percept) returns an action  
  persistent: rules, a set of condition–action rules  
  
  state  $\leftarrow$  INTERPRET-INPUT(percept)  
  rule  $\leftarrow$  RULE-MATCH(state, rules)  
  action  $\leftarrow$  rule.ACTION  
  return action
```

- El agente funciona apenas si el ambiente fuese completamente observable y la decisión correcta pudiese ser tomada basada apenas en la percepción actual.

Estructura de Agentes

Agente reactivo basado en modelo



Estructura de Agentes

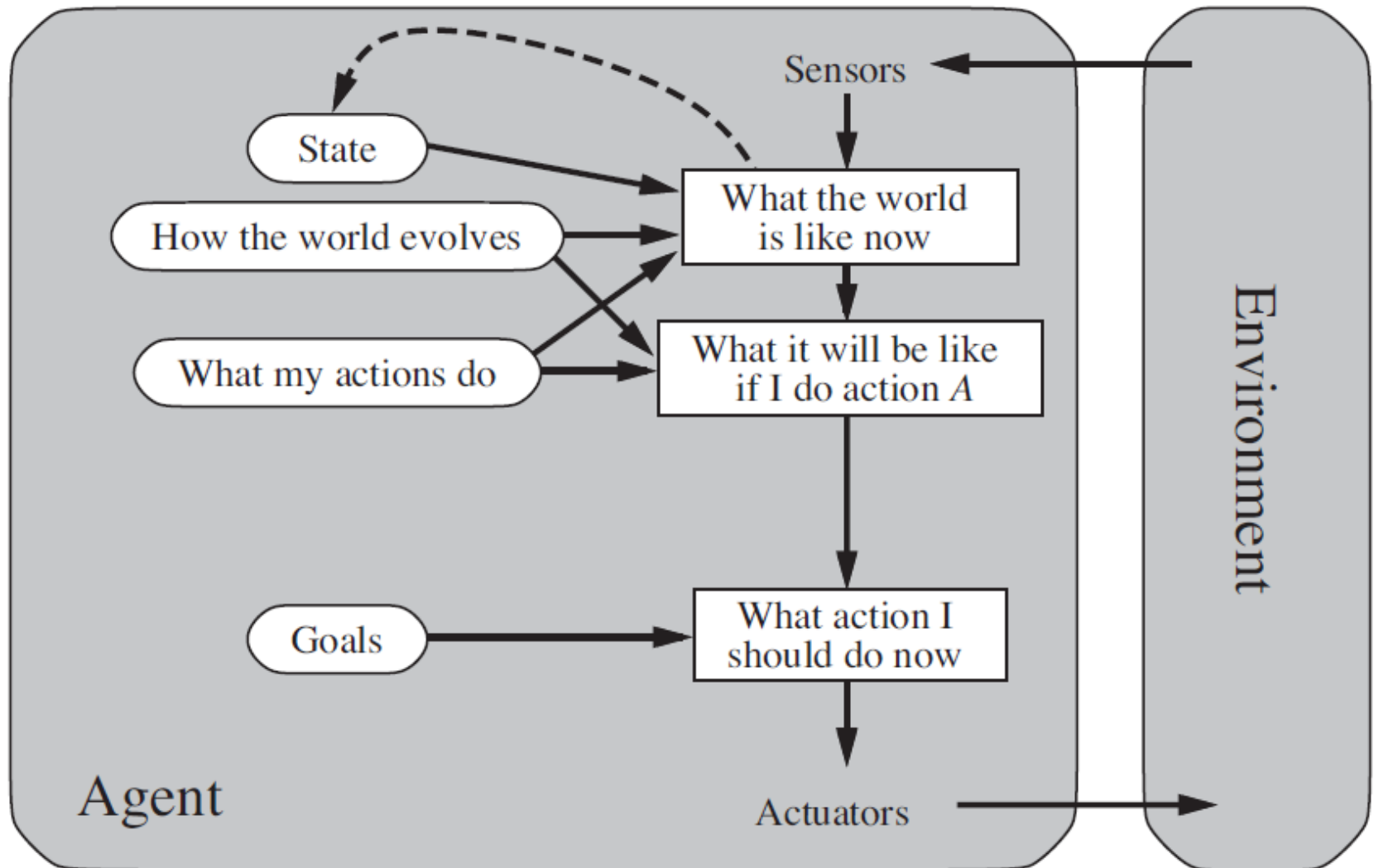
Programa del Agente Reactivo Basado en Modelo

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
               model, a description of how the next state depends on current state and action
               rules, a set of condition–action rules
               action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

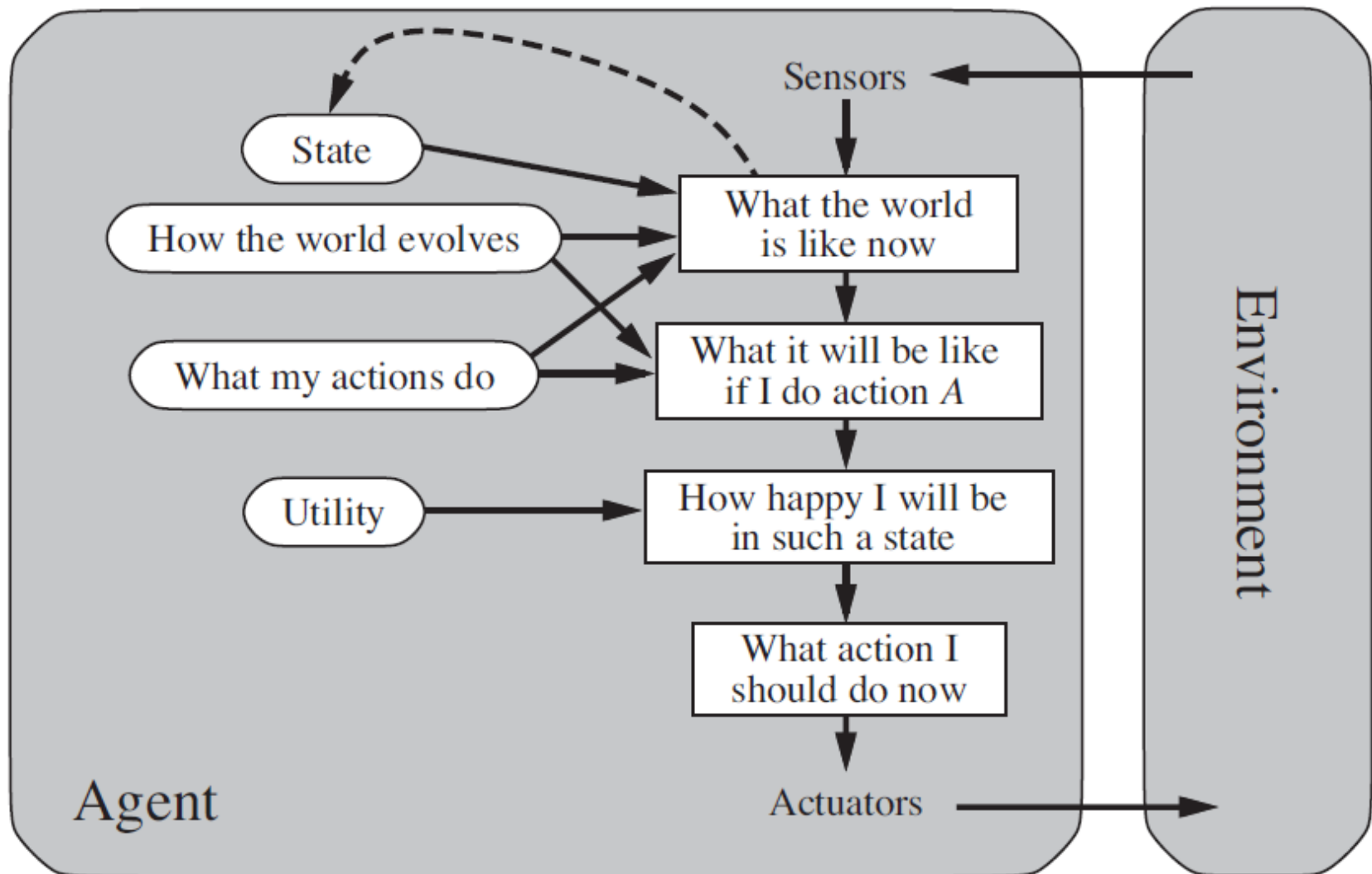
Estructura de Agentes

Agente basado en objetivo



Estructura de Agentes

Agente basado en utilidad

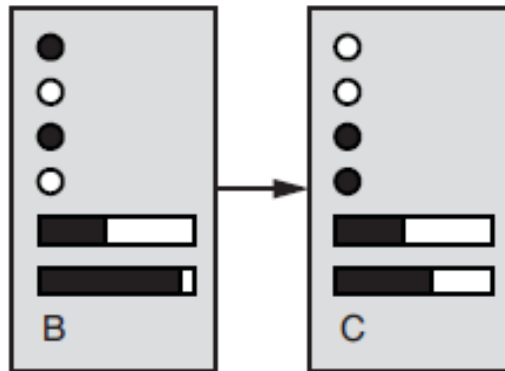


Representación de estados en agentes

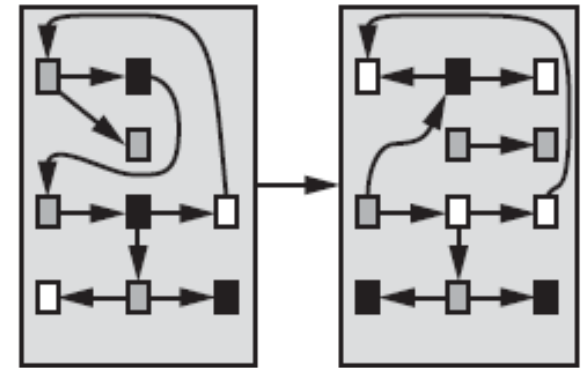
Tipos de Representación de Estados:



Atómica



Factorada



Estructurada

Representación de estados en agentes

Consideraciones para representacion de estados en juegos:

- Representación depende del tipo de salida del juego:
 - juegos textuales -> strings,
 - Juegos de tableros -> matriz de posiciones, etc)
- El mismo juego puede ser representado en varias formas. Por ejemplo, un juego de carreras de carros:
 - Lista de posiciones y velocidades de todos los carros (vista externa)
 - Lista de distancias y ángulos a todos los otros carros (vista interna)
- Representación es **crucial** para el resto del diseño del agente



Preguntas?