

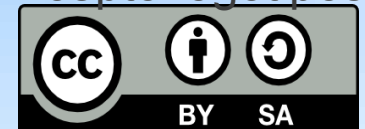


# Inteligencia artificial para videojuegos

## Una aproximación con sistemas expertos basados en reglas



(c) Manuel Palomo Duarte,  
Excepto logotipos



# Índice

- Introducción
- Sistemas Expertos
- Sistemas Expertos basados en reglas (RBES)
  - CLIPS
- Aplicación a un juego sencillo: Gades Siege
- Conclusiones
- Referencias

# Introducción

- La inteligencia artificial (IA) en los videojuegos suele usarse para manejar los personajes no jugables (NPC)
- Hay que mantener compromisos:
  - Que sea real pero superable por el jugador
  - Que sea un reto (superable aprendiendo)
  - Que no consuma demasiados recursos
  - Que no requiera demasiado tiempo de desarrollo
  - ...

# Introducción

- Ramas en la IA según Wikipedia en español:
  - Inteligencia computacional: usa métodos automáticos para crear los comportamientos
    - Redes neuronales, computación evolutiva, enjambres, etc
  - Inteligencia simbólica: razona con representaciones simbólicas según comportamientos “definidos”
    - Aprendizaje automático, planificación automática, sistemas expertos, etc

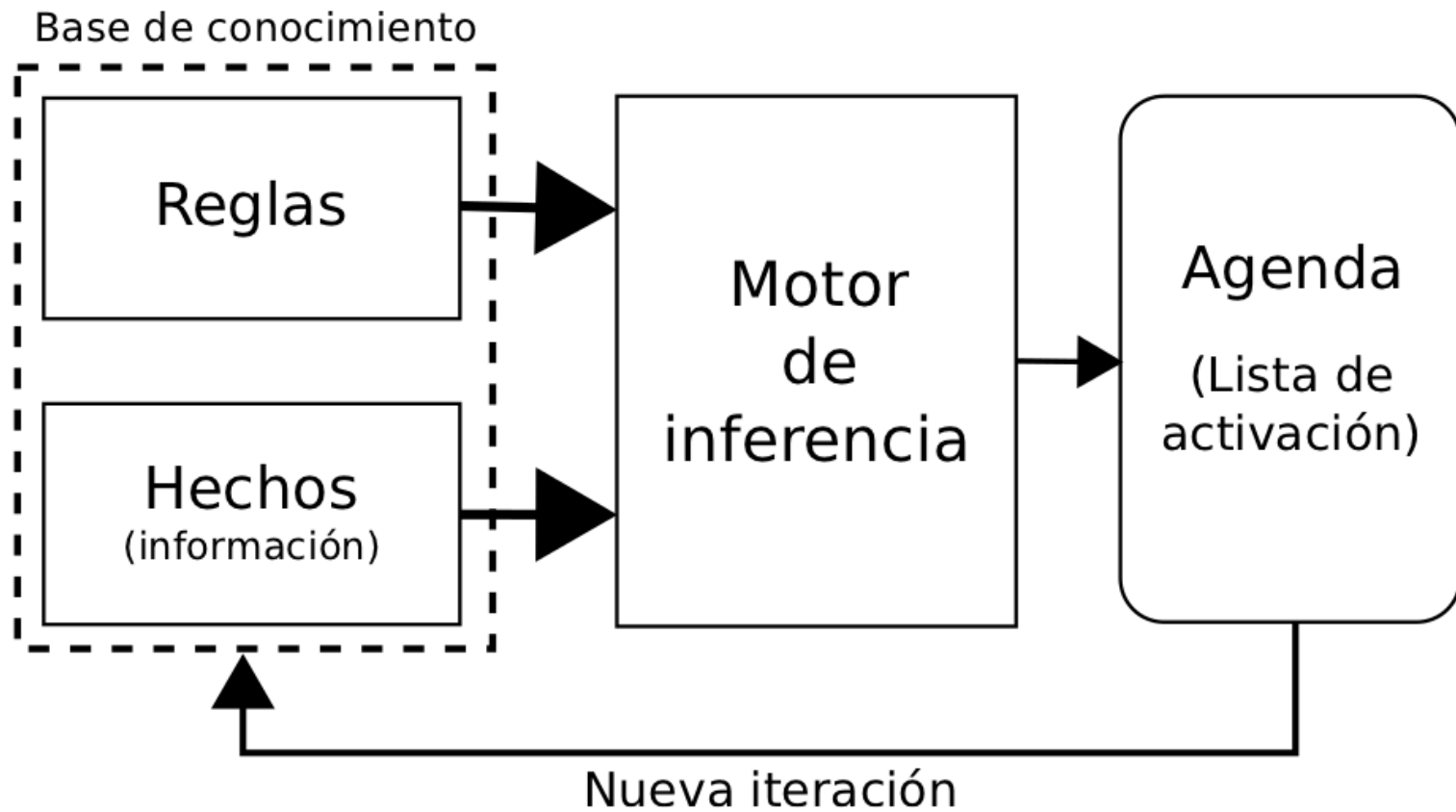
# Sistemas expertos

- Son programas que simula el comportamiento de un human experto en una materia concreta
- Se usan desde hace más de 30 años en muchas ramas de la ciencia
  - De los primeros “casos de éxitos” de IA aplicada
- Existen diversos tipos:
  - Basados en casos, árboles de decisión, basados en reglas, etc

# SEBR

- Componentes:
  - Hechos: información sobre el entorno
  - Reglas: mecanismos para inferir nuevo conocimiento
  - Motor de inferencia: gestiona las reglas
    - *Agenda*: lista de reglas “activables/disparables”
    - Estrategia de resolución de conflictos: algoritmo de decisión de qué regla activar en caso de que haya varias posibles (FIFO, random, less triggered, etc)

# SEBR - Arquitectura



# CLIPS



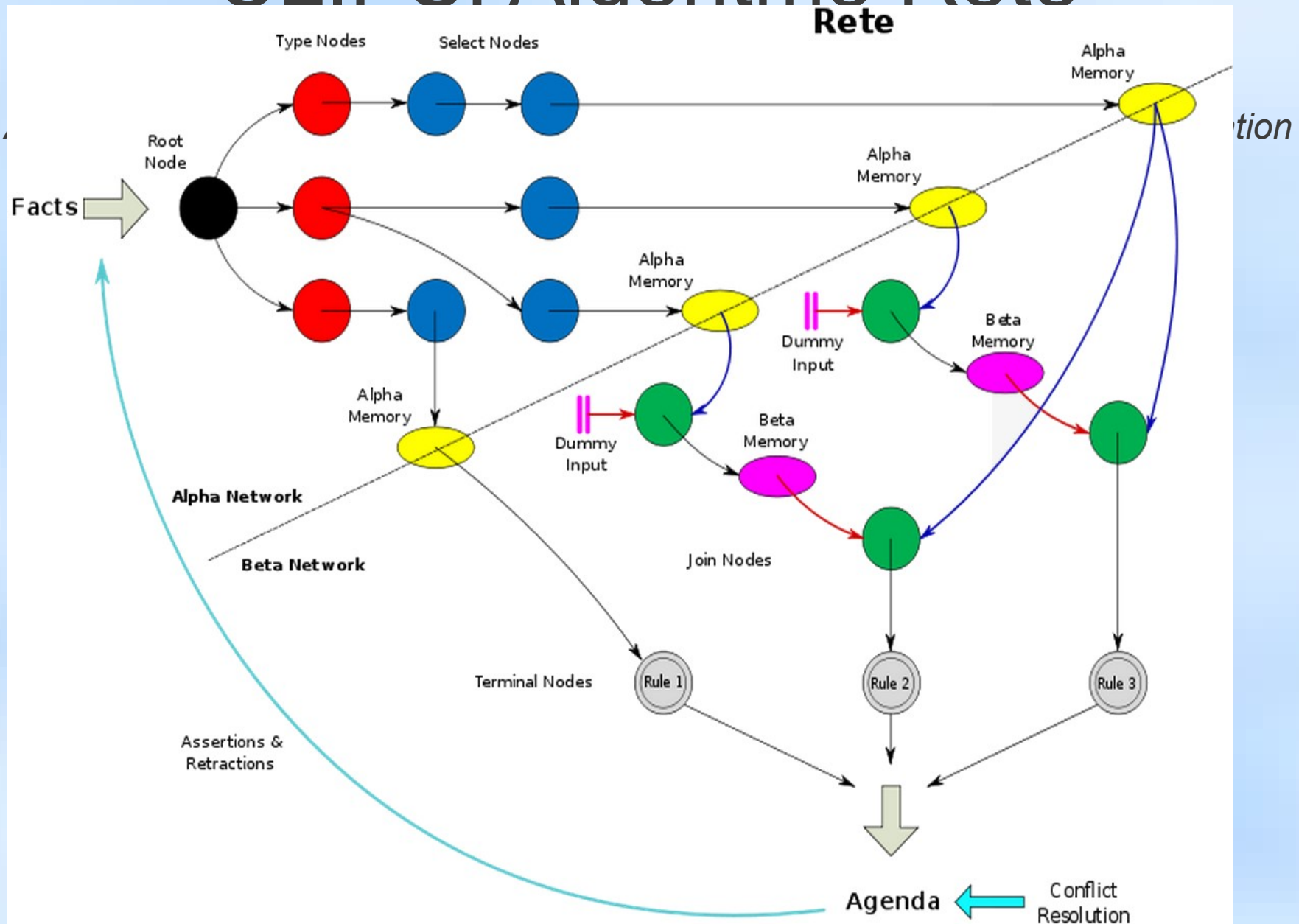
- Es el entorno de referencia de SEBR
- Desarrollado inicialmente por la NASA
  - Mantenido por uno de sus creadores
  - Disponible bajo licencia libre
- Sabores: pyCLIPS, fuzzyCLIPS, JESS, etc.
- Ojo, no confundir con Common LISP (CLISP)
  - Otro lenguaje, funcional. Con sintaxis también basada en paréntesis



# CLIPS

- Elementos propios:
  - Plantillas para estructurar información
    - Ej: (persona (edad 66) (peso 80))
  - Reglas agrupables en módulos
    - Se pasan el control del sistema entre sí bajo demanda
  - Reglas con prioridades asociadas
    - Sólo se comprueban reglas de baja prioridad si ninguna de prioridad superior se puede activar
  - Permite definir funciones
    - Interfaz con lenguaje C

# CLIPS: Algoritmo Rete



# Aplicación a un juego sencillo

- Válido para cualquier tipo de juego que requiera IA
  - Aplicación ha de ser “dirigida” por un experto
- Varias reglas de igual prioridad dan aleatoriedad
  - No es un árbol de decisión
- Lo veremos aplicado a un “*serious game*”:
  - Asignatura “Diseño de Videojuegos”
    - Optativa 3º ITIS en la Universidad de Cádiz
  - ABP, grupos de 3. Llegaban justos al tema de IA
    - Algunos alumnos no tenían formación en IA
    - 2 sesiones teóricas y 2 de laboratorio (2 horas cada)

# Gades Siege

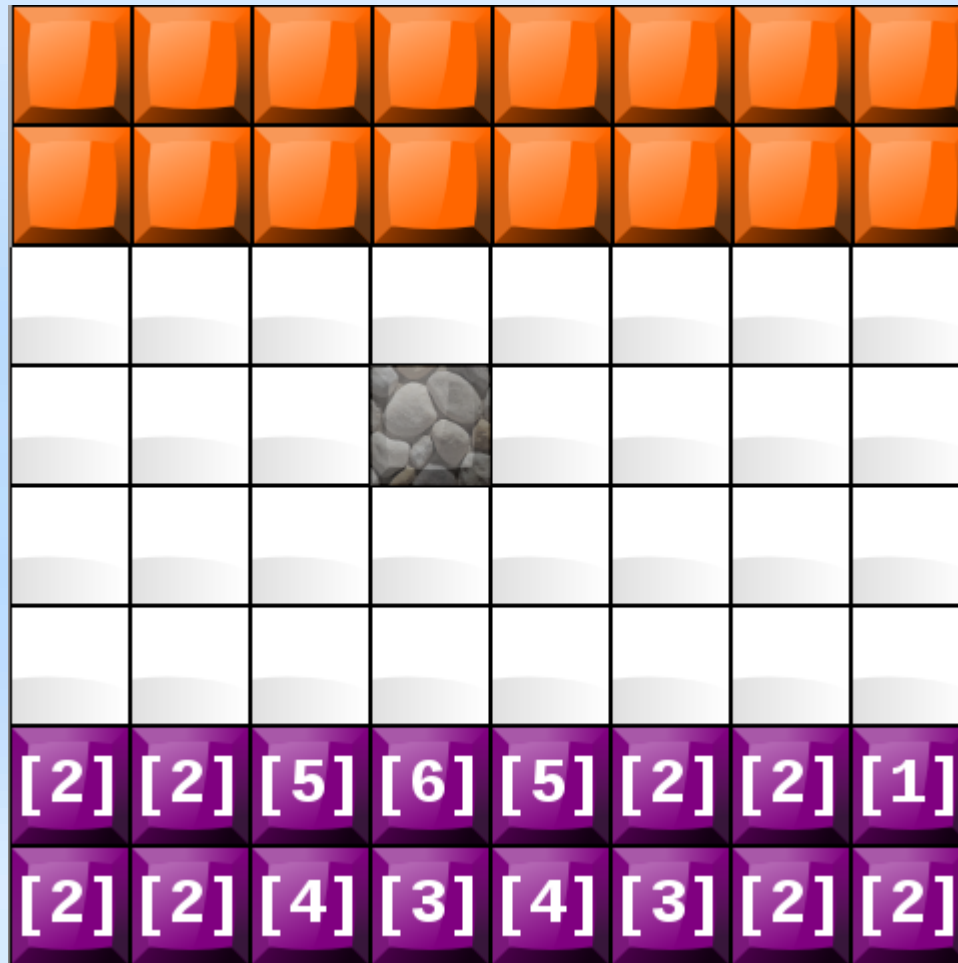


- Simplificación del juego de mesa Stratego
- Enfrentamiento entre dos ejércitos
  - Tablero de 8x8
  - Cada ficha tiene un valor asociado
    - Inicialmente permanece oculto al contrario (conocimiento parcial del entorno)
  - Las fichas se mueven por turnos
    - Cuando colisionan fichas de distinto ejército se descubren sus valores y la más débil muere
- Objetivo: capturar al rey del contrario (ficha 1 punto)

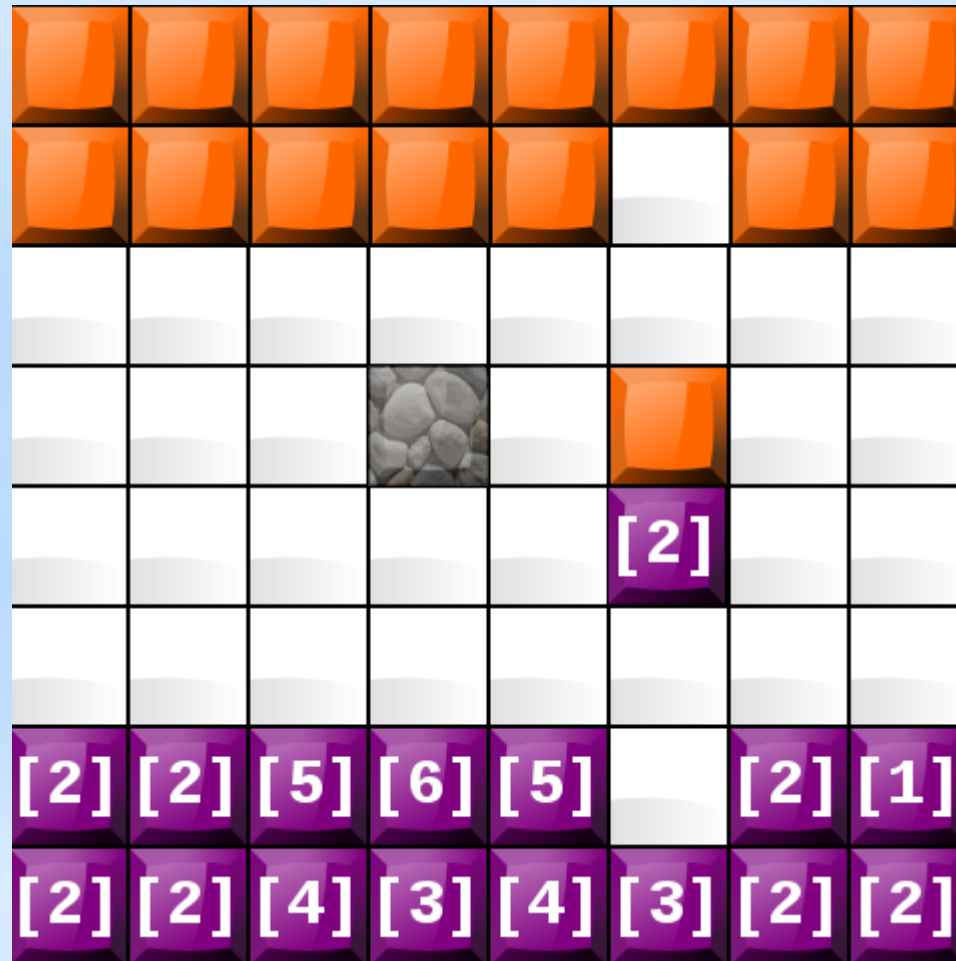
# Gades Siege

- Cada jugador pone las fichas en el orden inicial que desee en sus dos filas más próximas
- Si no se da una orden de mover ficha, se mueve una al azar hacia el centro del tablero
- Se pueden poner casillas no alcanzables (obstáculos)
- Si tras un tiempo máximo nadie mata al rey contrario hay empate
- El juego es intencionadamente sencillo

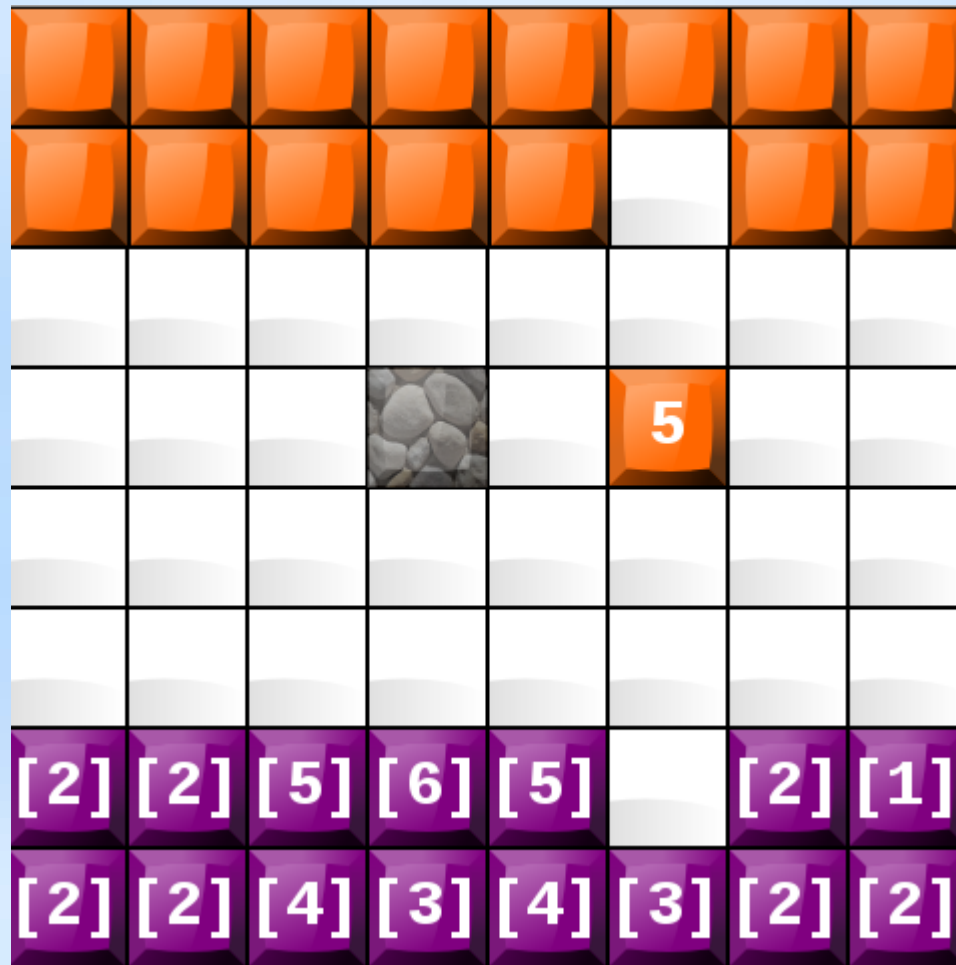
# Ejemplo de estados de una partida



# Ejemplo de estados de una partida



# Ejemplo de estados de una partida





# Ejemplo de estados de una partida

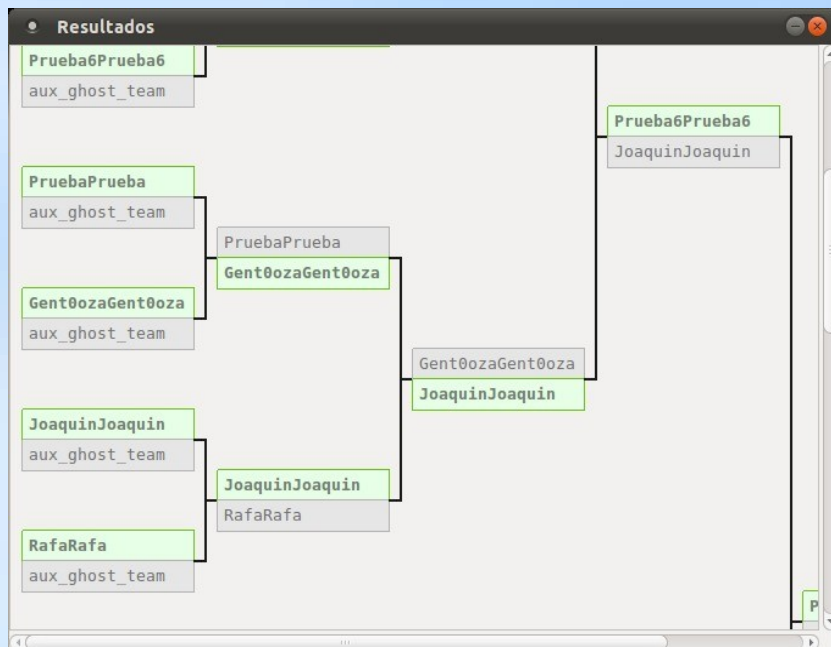
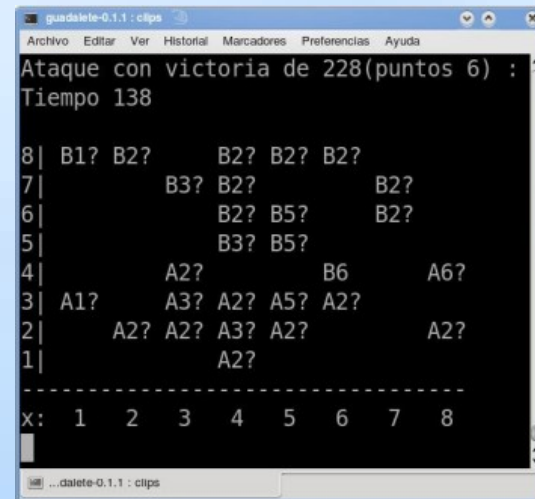
[2]	[2]	[2]	[1]	5	[2]	[2]	[2]
	[4]	5			[3]	[4]	
			[5]				
		[4]					
			[6]				5
				[4]			
			[3]		[2]	[2]	[1]
	6				[3]	[2]	[2]

# Gades Siege

- Evolución, una versión por curso
  - Versión 0.1: modo texto (totalmente funcional)
    - Versión 0.1.1: con visor gráfico de partidas ya jugadas
  - Versión 1.0: aplicación gráfica interactiva
    - Torneos. Aproximación errónea al control de CLIPS
  - Versión 2.0: reescritura.
    - Partidas por lotes, control humano, etc
  - Versión 3.0: ampliación.
    - Obstáculos en el tablero, mejor feedback, etc.

# Gades Siege

“Si mi madre me hubiera visto con 8 tíos más gritando y saltando delante de una pantalla negra con letritas, se moría del disgusto”



# Modelado del juego

- Plantilla Ficha:
  - Equipo: A (equipo propio), B (equipo contrario)
  - Num: identificador único
  - Puntos: valor de la ficha
  - Pos-x: posición en el eje X
  - Pos-y: posición en el eje Y
  - Descubierta: 0 si está oculto su valor al contrario, 1 en caso contrario

# Modelado del juego

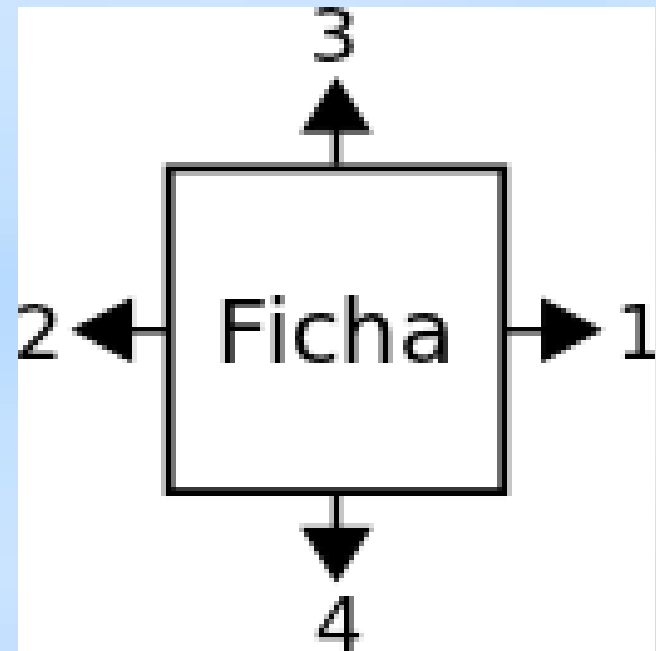
- Ejemplo de fichas
  - (ficha (equipo A) (num 234) (pos-x 1) (pos-y 2) (descubierta 0) (puntos 1))
  - (ficha (equipo B) (num 401) (pos-x 8) (pos-y 8) (descubierta 0))
  - (ficha (equipo B) (num 661) (pos-x 4) (pos-y 4) (descubierta 1) (puntos 3))

# Modelado del juego

- Hechos
  - Posiciones de las fichas
  - Obstáculo
  - Tiempo
  - Se pueden crear hechos internos en cada módulo
  - Se refleja la posición de las fichas para razonar siempre desde la casilla (0,0)

# Modelado del juego

- Plantilla Mueve:
  - Num: identificador de la ficha
  - Mov: movimiento (1-4)
  - Tiempo: paso de la partida
- Ejemplo:
  - (assert (mueve (num 137) (mov 3) (tiempo 23)))



# Ejemplos de reglas

```
(defrule EQUIPO-A::atacar
```

```
  (declare (salience 30))
```

```
  (ficha (equipo "A") (num ?n1) (pos-x ?x1) (pos-y ?y1) (puntos ?p1))
```

```
    (tiempo ?t)
```

```
=>
```

```
  (assert (mueve (num ?n1) (mov 2) (tiempo ?t)))
```

```
)
```



# Ejemplos de reglas

```
(defrule EQUIPO-A::atacar1
  (declare (salience 30))
  (ficha (equipo "A") (num ?n1) (pos-x ?x1) (pos-y ?y1) (puntos ?p1))
  (ficha (equipo "B") (num ?n2) (pos-x ?x2) (pos-y ?y2) (puntos ?p2)
    (descubierta 1))
  (test (and (> ?p1 ?p2) (= ?y1 ?y2) (> ?x1 ?x2)))
  (tiempo ?t)
=>
  (assert (mueve (num ?n1) (mov 2) (tiempo ?t)))
)
```

# Ejemplos de reglas

```
(defrule EQUIPO-A::atacar2
  (declare (salience 20))
  (ficha (equipo "A") (num ?n1) (pos-x ?x1) (puntos ?p1))
  (ficha (equipo "B") (num ?n2) (pos-x ?x2) (puntos ?p2) (descubierta 1))
  (test (and (> ?p1 ?p2) (> ?x1 ?x2)))
  (tiempo ?t)
=>
  (assert (mueve (num ?n1) (mov 2) (tiempo ?t)))
)
```

# Ejemplos de reglas

```
(defrule EQUIPO-A::huir1
  (declare (salience 20))
  (ficha (equipo "A") (num ?n1) (pos-x ?x1) (pos-y ?y1) (puntos 1))
  (ficha (equipo "B")          (pos-x ?x2) (pos-y ?y2) (puntos 5))
  (test (and (= ?x1 ?x2) (> ?y1 ?y2)))
  (tiempo ?t)
=>
  (assert (mueve (num ?n1) (mov 4) (tiempo ?t)))
)
```

# Ejemplos de reglas

```
(defrule EQUIPO-A::despistar
  (declare (salience 10))
  (ficha (equipo "A") (num 11) (pos-x ?x1) (pos-y ?y1))
  (test (< ?y1 4))
  (not (ficha (equipo "A") (pos-x ?x1) (pos-y (+ 1 ?y1)))))
  (tiempo ?t)
=>
  (assert (mueve (num 11) (mov 3) (tiempo ?t)))
)
```

# Estrategias

- Una buena estrategia tiene que equilibrar:
  - Reglas de ataque y de defensa/huida
  - Reglas para el principio de la partida y para el final
  - De acuerdo a una disposición inicial
- Se puede depurar el comportamiento viendo qué regla se disparó en qué momento
  - Y eliminarla o re-priorizarla

# Gades Siege

*¿Echamos unas partidillas?*

# Conclusiones

- Hemos mostrado el funcionamiento de SEBR
- Con un SEBR se puede implementar el comportamiento de un experto
  - De manera sencilla, extensible y mantenible
  - A partir de hechos, reglas, prioridades y módulos
- Aplicación a juego sencillo de mesa
  - Ejemplos de reglas sencillas ...
  - Combinadas en comportamientos complejos

# Referencias

- Desarrollo:
  - Joseph C. Giarratano, Gary D. Riley. Expert Systems: Principles and Programming, 4th Ed.
  - CLIPS: <http://clipsrules.sourceforge.net/>
- Aplicación a la enseñanza
  - Manuel Palomo-Duarte, Juan Manuel Doderó, Antonio García-Domínguez: Betting system for formative code review in educational competitions. Expert Systems with Applications 41(5): 2222-2230 (2014)



Proyecto “La competitividad y el análisis crítico en la evaluación” (CIE44) financiado por el programa de Innovación Educativa 2010 de la Universidad de Cádiz. Aportaciones de Roberto García Carvajal, Pablo Recio Quijano y José Tomás Tocino García

Gracias por vuestra atención  
*¿Preguntas?*

Presentación, entorno y equipos de ejemplo descargables en  
<http://code.google.com/p/gsiege>