



INTELIGENCIA ARTIFICIAL

Proyecto N° 1

**Desarrollo de Agentes Inteligentes
para el control de personajes en un Juego de Rol**

Segundo Cuatrimestre de 2009

Introducción y Objetivos

El presente proyecto se presenta en el contexto de un *juego de rol virtual* (*i.e.*, implementado en software), *multi-jugador* (*i.e.*, admite la participación de varios jugadores al mismo tiempo). Concretamente, el juego se desarrolla en un mundo virtual recreando una comarca de la época medieval, con llanuras, bosques, ríos y montañas. Los personajes jugadores son caballeros, y merodean por la comarca en búsqueda de tesoros que se hallan dispersos por esta última. El **propósito de los caballeros** es apoderarse de la mayor cantidad posible de tesoros, y podrán conseguirlo ya sea explorando la comarca para descubrirlos y levantarlos del suelo, o atacando a otros jugadores para saquearlos (robarles los tesoros que traen). La comarca cuenta también con edificaciones de la época. Principalmente se destacan las posadas, donde los personajes pueden descansar y reponerse de sus travesías, ya que si están demasiado débiles se encuentran más expuestos a ser saqueados por otros agentes, o incluso a desmayarse por falta de energía y perder así todo lo que traen.

En el marco de este juego, el **objetivo general** del presente proyecto es el diseño e implementación de un agente inteligente para el control de un personaje jugador.

Características generales del juego y los agentes

La comarca es una grilla de m filas y n columnas, donde cada celda o *posición* tiene asociado un tipo de terreno (llanura, montaña, bosque o agua), y puede albergar uno o más objetos y/o agentes (caballeros) en un dado instante del juego.

Cada agente cuenta con cierta energía (*stamina*), que va consumiendo al realizar acciones o como resultado de sufrir ataques de otros agentes, y que puede recargar hospedándose en las posadas (*hostels*). Los agentes tiene además una cierta capacidad o límite máximo de energía, indicado por el atributo *max_stamina*. Adicionalmente, los agentes cuentan con una determinada habilidad de pelea (*fight_skill*), donde un mayor valor para este atributo se reflejará en una mayor chance de ganar un combate con otro agente.

Ciertas características o habilidades de los agentes pueden mejorar como resultado de sus experiencias. Concretamente, si un agente vence en un ataque, su habilidad de pelea (*fight_skill*) se incrementa en 1, y luego de un cierto número m (parámetro del sistema) de movimientos de avance o giro, el agente incrementará su capacidad de energía (*max_stamina*) en 1.

El tiempo en el juego se encuentra discretizado en turnos, y éstos tienen una duración aproximada de 1 segundo. Los agentes pueden solicitar una percepción o (intentar) efectuar una acción en cualquier momento, pero obtendrán a lo sumo una percepción por turno (al comienzo del turno) y podrán efectuar a lo sumo una acción por turno. Idealmente, un agente obtendrá una percepción al comienzo de cada turno, decidirá qué hacer en menos de 1 segundo,

y “solicitará” la ejecución de una acción, pidiendo inmediatamente la siguiente percepción para así obtenerla al comienzo del turno próximo, y así siguiendo. De esta forma, el agente consigue percibir y actuar en todos los turnos. Si el agente tarda más tiempo en decidir y, por ejemplo, comunica la acción a realizar luego de finalizado el turno, entonces ésta se ejecutará recién al final del turno siguiente, es decir, el agente pierde un turno.

Percepciones de los agentes

El agente es capaz de percibir el número de turno actual del juego, una imagen o concepción visual de la comarca, los valores de sus atributos y los objetos (tesoros) que carga actualmente.

El rango de visión del agente es retringido, y trata de simular las características de la visión en los seres humanos. Los agentes cuentan con un ángulo de visión de 180° (respecto a la dirección hacia la que mira) y una profundidad de visión limitada a d celdas de distancia (parámetro del juego). La figura 1 muestra el conjunto de celdas comprendidas dentro del rango de visión del agente para $d = 3$. Para cada posición dentro de su rango de visión, el agente ve el tipo de terreno de dicha posición, así como las “cosas” (objetos, agentes y edificaciones) que se encuentran en ella.

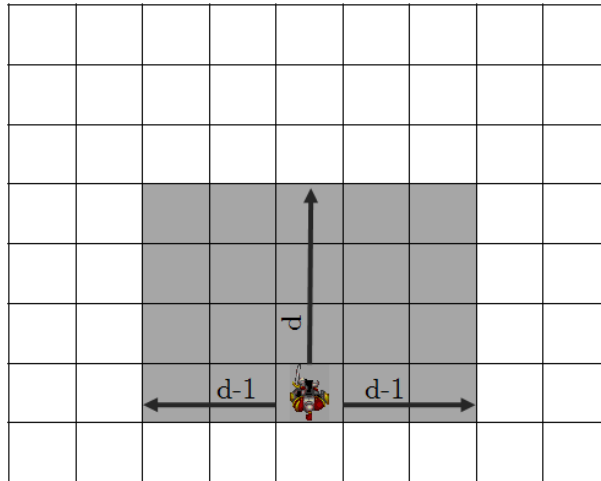


Figura 1: Rango de visión ($d = 3$)

Representación en Prolog

La percepción que el agente obtiene del entorno se representa mediante la siguiente estructura (lista) PROLOG $[Turn, Vision, Attrs, Inventory]$ donde:

- $Turn$ es un entero positivo representando el turno actual del juego.
- $Vision$ es una estructura codificando lo que el agente ve a través de sus ojos. Concretamente $Vision$ es una lista de la forma $[[Pos_1, Land_1, Things_1], \dots [Pos_n, Land_n, Things_n]]$ tal que Pos_i es una posición dentro del rango de visión del agente, $Land_i$ es el tipo de terreno asociado a la posición Pos_i , y $Things_i$ es una lista con todas las “cosas” (objetos, edificaciones y agentes) que se encuentran en Pos_i .
- $Attrs$ es una estructura codificando los valores de distintos atributos del agente. Concretamente, $Attrs$ tiene la forma $[[Attr_1, Val_1], [Attr_2, Val_2], [Attr_n, Val_n]]$,

donde cada $Attr_i$ es el nombre de un atributo del agente y Val_i es su valor asociado. La siguiente tabla describe los atributos del agente.

Nombre del atributo	Valor
<code>pos</code>	posición actual del agente
<code>dir</code>	n (north), e (east), s (south) o w (west), indicando la dirección actual del agente
<code>stamina</code>	entero positivo indicando el nivel de energía actual del agente
<code>max_stamina</code>	entero positivo indicando el nivel máximo de energía del agente
<code>fight_skill</code>	entero positivo indicando la habilidad de pelea del agente

- *Inventory* es una lista con los objetos que lleva el agente.
- Los objetos, edificaciones y agentes (“cosas”) se codifican mediante estructuras de la forma $[ThingType, ThingName, Description]$, tal que $ThingType \in \{agent, building, treasure\}$, $ThingName$ es una constante PROLOG, y $Description$ es una lista de la forma $[[Attr_1, Val_1], [Attr_2, Val_2], [Attr_n, Val_n]]$ donde cada $Attr_i$ es el nombre de un atributo de la “cosa” y Val_i es su valor asociado.
- una posición de la grilla se representa mediante una lista $[F, C]$, donde F y C son enteros positivos indicando la fila y columna, respectivamente, de la posición.

Acciones de los agentes

El agente cuenta con acciones para desplazarse por la comarca, concretamente, *avanzar* y *girar* hacia una dada dirección cardinal (norte, sur, este u oeste), es capaz de *levantar* y *soltar* objetos, y puede *atacar* a otros agentes.

Precondiciones y Efectos de las acciones

Para que un dado agente pueda ejecutar con éxito una cierta acción (consiguiendo así sus efectos), las precondiciones de la acción deben satisfacerse en el estado actual del mundo. Si el agente intenta ejecutar una acción cuyas precondiciones no se encuentran satisfechas, entonces la acción no se ejecuta y el agente pierde la oportunidad de actuar en ese turno.

A continuación se describen las precondiciones y efectos de las acciones disponibles para los agentes. ACLARACIÓN: cuando digamos que la ejecución de una dada acción isume n turnos nos referimos a que la acción terminará de ejecutarse (causando los efectos correspondientes) recién luego de transcurridos n turnos a partir de que el agente solicitó su ejecución. Si inmediatamente luego de solicitar la ejecución de la acción el agente pide una nueva percepción, no la obtendrá hasta luego de transcurridos los n turnos que la acción tarda en ejecutarse. En particular, si $n = 1$ la acción se habrá terminado de ejecutar en el turno inmediato siguiente, y el agente podrá percibir y volver a actuar en dicho turno.

ACCIÓN: avanzar

PRECONDICIONES

La posición en frente del agente de acuerdo a su dirección (posición destino) debe ser de llanura o de montaña. Esto es, el agente no puede avanzar sobre celdas de bosque ni de agua.

EFFECTOS

El agente avanza sobre la posición destino. Si esta es de llanura, la acción consume 1 nivel de stamina e insume 1 turno. Si la posición destino es de montaña, la acción consume 2 niveles de stamina e insume 2 turnos (es decir, el agente se encontrará en la posición destino recién en el turno siguiente al próximo).

ACCIÓN: girar (hacia un dado punto cardinal)

PRECONDICIONES

No tiene.

EFFECTOS

El agente queda mirando hacia la dirección cardinal del giro. Además, la acción consume 1 nivel de stamina e insume 1 turno.

ACCIÓN: levantar (un dado objeto)

PRECONDICIONES

El objeto y el agente deben estar en la misma posición, y no debe haber otro agente en estado consciente en dicha posición.

EFFECTOS

El objeto deja de estar en el suelo y se incorpora al inventario del agente. La ejecución de esta acción insume 1 turno.

ACCIÓN: soltar (un dado objeto)

PRECONDICIONES

El agente debe tener el objeto en su inventario.

EFFECTOS

El objeto deja de estar en el inventario del agente y pasa a estar en el suelo (más precisamente en la posición actual del agente). La ejecución de esta acción insume 1 turno.

ACCIÓN: atacar (a un dado agente víctima)

PRECONDICIONES

El agente víctima debe encontrarse dentro del *rango de ataque* del agente. La figura 2 muestra el rango de ataque de un agente que se encuentra “mirando” hacia el norte.

EFFECTOS

Existen dos posibles efectos para esta acción: el agente que ataca puede efectivamente conseguir hacer daño a su oponente, o el agente víctima logra resistir el ataque saliendo ileso. El efecto concreto de la acción se determina considerando las habilidades de pelea (*fight skill*) de los jugadores atacante y víctima, además de un componente de azar. Más precisamente, se efectúan dos tiradas de un dado de n caras (n es un parámetro del juego), una tirada en nombre del agente atacante y otra en nombre del agente víctima, y los

valores obtenidos en cada tirada se suman a la habilidad de pelea (*fight skill*) del jugador respectivo. Consideremos que a es el valor que suma el agente atacante y v es el valor que suma la víctima. Si $a \geq v$, entonces el atacante causa una disminución de stamina en la víctima (daño) de $a - v$. Si por el contrario $v \geq a$, entonces la víctima resiste el ataque y no recibe daño. En cualquiera de los dos casos, el agente que ataca disminuye su stamina en 1 debido al esfuerzo que involucra efectuar el ataque. La ejecución de esta acción insume 1 turno.

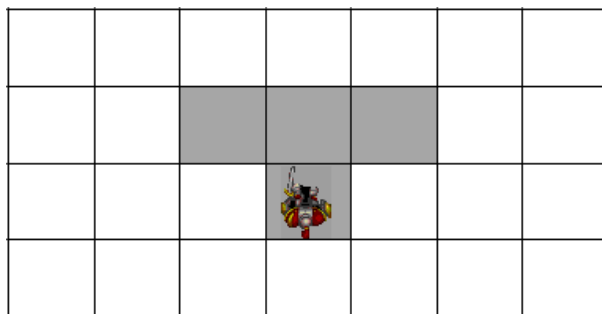


Figura 2: Rango de ataque

Algunas consideraciones adicionales

Si en un dado momento un agente se queda sin energía (ya sea como consecuencia de haber ejecutado una acción o de haber sufrido un ataque) entonces se desvanece, quedando imposibilitado de percibir y actuar, y permaneciendo en este estado por n turnos (parámetro del juego). Además, todos los objetos que cargaba caen al suelo, quedando en la posición donde el agente yace desmayado.

Los agentes pueden avanzar sobre una celda donde hay una posada, al igual que sobre cualquier celda de pasto. Cada turno que el agente permanezca en la posada su stamina se incrementará en k niveles (parámetro del juego). Para evitar que el agente permanezca indeterminadamente en la posada, al turno siguiente de haber llenado por completo su nivel de stamina, el agente es automáticamente expulsado de la posada. Además, luego de que el agente salió de una posada (ya sea voluntariamente o involuntariamente), no podrá volver a ingresar a la misma hasta luego de transcurridos t turnos (parámetro del juego).

Representación en Prolog

Las acciones serán representadas mediante las siguientes constantes y estructuras PROLOG:

- `move_fwd`.
- `turn(Dir)`, donde $Dir \in \{n, s, e, w\}$.
- `pickup(ObjName)`, donde *ObjName* es el nombre de un objeto.
- `drop(ObjName)`, donde *ObjName* es el nombre de un objeto.
- `attack(Victim)`, donde *Victim* es el nombre de un agente.
- `null_action`. Ejecutar esta acción en un dado turno es equivalente a no actuar en dicho turno. La acción `null_action` se incluyó para facilitar la programación de los agentes, y su uso es opcional.

Requerimientos de Diseño e Implementación

El diseño del agente debe respetar la arquitectura mostrada en la figura 3. Concretamente, el agente contará con un estado interno codificando, entre otras cosas, la concepción que tiene acerca del estado actual del mundo (por ejemplo, tipo de terreno de las celdas exploradas, ubicación de las posadas, ubicación de otros agentes y tesoros, etc.), así como cualquier otra información que le resulte útil “recordar” para decidir su accionar en el mundo.

Además, la implementación del agente debe modularizarse (mínimamente) en dos subprogramas (conjuntos de predicados). Por un lado, el subprograma implementando la actualización de la concepción del mundo, que a partir de la percepción corriente y la información acerca de la concepción actual del mundo almacenada en el estado interno, actualiza esta última para reflejar los cambios del mundo percibidos. Por otro lado, el subprograma implementando el mecanismo de decisión (razonamiento) del agente, que basándose en la información almacenada en el estado interno “decide” qué acción efectuar. Posiblemente, este subprograma también actualice cierta información del estado interno que resultará de utilidad para decisiones futuras.

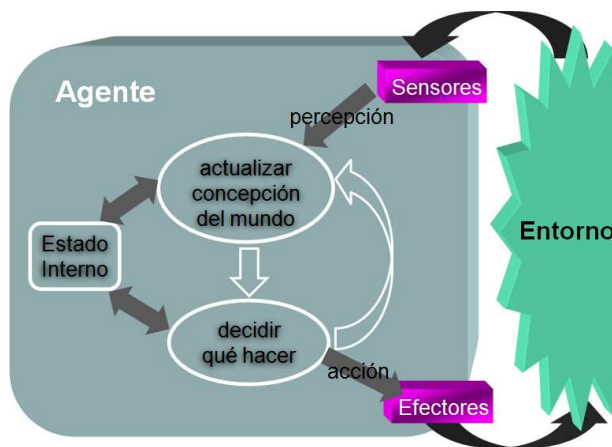


Figura 3: Arquitectura del agente

Condiciones de Entrega

Para el presente proyecto se establecen las siguientes condiciones de entrega:

1. Implementar un agente para el control de un personaje del juego, respetando los requerimientos establecidos en la sección anterior. **ACLARACIÓN:** en la evaluación del proyecto se dará importancia, principalmente, al uso *apropiado* de técnicas y conceptos vistos en la materia en la implementación del agente. Por supuesto, el fin último de aplicar dichas técnicas y conceptos es conseguir una buena performance del agente en el juego. Sin embargo, solo se valorará un buen desempeño del agente cuando dicho desempeño sea consecuencia de una implementación conceptualmente interesante.
2. Elaborar un informe de documento en forma completa el agente implementado. Principalmente, el informe deberá describir cómo se implementó cada uno de los componentes de la arquitectura presentada en la sección anterior, qué técnicas y conceptos de la materia se emplearon y cómo fueron empleadas, etc. Puede aprovechar el informe para sugerir ejemplos de corrida que ilustren distintas capacidades del agente desarrollado.

IMPORTANTE: El informe es el único instrumento con el que cuenta la comisión para describir la resolución (implementación) desarrollada, y por lo tanto constituye un elemento fundamental para la evaluación del proyecto por parte de la cátedra.

3. Las comisiones pueden estar conformadas por hasta 2 integrantes.
4. La fecha límite de entrega del presente proyecto es el día **lunes 16 de Noviembre de 2009** en el horario de clase. Los proyectos entregados fuera de término recibirán una penalización en su calificación, la cual será proporcional al retraso incurrido.
5. Deberá presentarse un folio plástico CERRADO (no entregar carpetas) conteniendo los siguientes elementos:
 - El informe impreso, que deberá estar encabezado por una carátula identificando claramente a los integrantes de la comisión.
 - Un disquete o cd conteniendo un archivo ‘agente.pl’ con el código del agente implementado. Puede incluirse también archivos ‘.pl’ codificando grillas que permitan ilustrar comportamientos interesantes del agente.

Agradecimientos

A Sebastián Gottifredi y Mariano Tucát, por su indispensable colaboración en la concreción de este proyecto.