

75.68 SISTEMAS DE SOPORTE PARA CELDAS DE
PRODUCCIÓN FLEXIBLE

SISTEMA INTELIGENTE AUTÓNOMO QUE JUEGA
VIDEO JUEGOS

The Legend of Zelda (simplificado)



2do cuatrimestre, 2017

Autor: Ing. Juan Manuel Rodríguez

Contenido

Introducción y motivación	3
Objetivo del trabajo practico	3
Sobre el juego The Legend of Zelda y los niveles a resolver	3
Descarga y ejecución del <i>framework</i> de GVGAi.....	4
La clase ar.fi.uba.celdas.Agent	4
Entregas.....	5
Primera entrega	5
Entregables.....	5
Fecha de entrega.....	5
Segunda entrega	5
Entregables.....	5
Fecha de entrega.....	6
Informe.....	6

Introducción y motivación

El siguiente trabajo práctico está basado en la competencia: [The General Video Game AI Competition](#) (GVGAI). Dicha competencia busca promover y explorar el problema general que existe en crear un Agente Inteligente capaz de jugar diferentes vídeo juegos. Es decir, que al igual que un ser humano, se le podría presentar un vídeo juego nuevo, uno que nunca antes vio, y tendría que ser capaz de aprender sus reglas, jugarlo y ganarlo. Explora además de los juegos de un solo jugador, la variante de los juegos cooperativos y la creación automática de niveles para cualquier juego dado. Dicha competencia está patrocinada por Google DeepMind.

Objetivo del trabajo practico

El objetivo del trabajo práctico de la materia es construir un agente inteligente autónomo que juegue al video juego identificado como Zelda dentro del *framework* propuesto por GVGAI de forma cooperativa. Se trata no solo de desarrollar un sistema capaz de jugar correctamente a los niveles de Zelda propuestos, sino de que lo haga aprendiendo desde cero.

Sobre el juego The Legend of Zelda y los niveles a resolver

The Legend of Zelda es un video juego clásico originalmente para la consola Nintendo de 8 bits. En dicho juego, aparecen niveles en los cuales el personaje tiene que, por ejemplo, tomar una llave y llevarla hasta una puerta sorteando arañas u otros monstruos. Estos enemigos pueden ser muertos usando una espada. Estos niveles, son replicados, de forma sencilla dentro del framework GVGAI. Se muestra una captura a continuación:



Hay seis acciones posibles: no hacer nada, moverse hacia atrás, moverse hacia adelante, moverse hacia la izquierda, moverse a la derecha y atacar.

Descarga y ejecución del *framework* de GVGA

Pueden descargar una versión modificada, del *framework*, preparada para la materia en la siguiente dirección: <https://goo.gl/S4bXrA>

El proyecto pertenece a Eclipse, pero pueden importarlo usando la IDE que más les guste, editor de texto, etc. Eso corre por cuenta del alumno.

Para correr el juego, simplemente ejecuten la clase: “TestCeldas”. La cual está directamente en el directorio “src”, es decir en el paquete “default” de Java.

En dicha clase identifiquen las líneas 38 y 41.

La línea 38:

```
ArcadeMachine.playOneGame(game, level1, recordActionsFile, seed);
```

Permite que un ser humano juegue. En ese caso pueden utilizar el teclado para controlar al personaje. Utilicen las flechas o bien las letras: a,w,s,d para moverse y la barra espaciadora para atacar.

La línea 41:

```
ArcadeMachine.runOneGame(game, level1, visuals, celdasAgent, recordActionsFile, seed, 0);
```

Esta línea está originalmente comentada, descoméntenla y comenten la línea anterior para que el juego sea jugado por el agente inteligente, la clase “Agent” del paquete: “ar.fi.uba.celdas”

La clase ar.fi.uba.celdas.Agent

Esta clase representa a nuestro Agente Inteligente, por ahora no es muy inteligente ya que se mueve al azar. Identifiquen en dicha clase el método: “act” ese método es el que se ejecuta en cada iteración del juego, recibe una lista de observaciones, que se procesan como una “percepción”, el cual es un *array* que indica la posición de cada elemento en el nivel y devuelve una acción.

En esta clase tendrán que añadir la lógica necesaria para convertir a este Agente en un Agente Inteligente Autónomo. Según lo que se dictará en clases.

Entregas

Este trabajo tendrá al menos dos entregas.

Primera entrega

Esta entrega tiene como finalidad familiarizarse con el código y el problema a resolver. Su objetivo, es que el Agente utilice una serie de reglas escritas a mano, de tal forma que le permitan ganar por lo menos el nivel 1 del juego Zelda.

Las reglas deberán estar escritas en un archivo de texto con formato JSON, el agente deberá ser capaz de cargarlas en memoria y ejecutarlas en cada llamada al método “act”.

Las reglas pueden ser ejecutadas con un motor de reglas como Drools o bien su interpretación puede estar programada de forma manual con sentencias “IF”. Ya que de todos modos, estas reglas serán descartadas más adelante cuando el agente sea capaz de aprender por sí solo.

Entregables

- Código fuente
- Archivo JSON de reglas.

Fecha de entrega

Miércoles 27 de Septiembre

Segunda entrega

Para esta entrega se espera que el agente haya implementado el flujo de aprendizaje, basado en la utilización de Teorías, que implemente un planificador para escoger en cada posible curso de acción aquellas acciones que maximicen su utilidad y que implemente al menos una Heurística para generalizar el conocimiento adquirido.

Se espera que el agente pueda aprender solo las reglas necesarias para ganar cualquier nivel a partir de la experiencia y el aprendizaje por refuerzo. Dichas Reglas, que serán Teorías en nuestro modelo, se guardarán en el archivo JSON.

Una vez entregado el código fuente, se espera que el docente pueda repetir el proceso de entrenamiento y aprendizaje en su máquina. Partiendo de un archivo de conocimiento vacío.

Entregables

- Código fuente
- Archivo JSON de Teorías.

- Informe en formato con las características que indican más abajo

Fecha de entrega

Fin de cursada

Informe

Juntamente con las clases y el archivo JSON con el conocimiento del agente se deberá entregar un informe sencillo y conciso explicando la estrategia utilizada, las dificultades encontradas y la evolución del agente. Dicho informe deberá tener entre 10 y 15 páginas en tamaño A4 y deberá estar en el formato *lectures notes in computer science* (LNCS) que es un formato utilizado por Springer para la publicación de artículos científicos. Dejo a continuación *links* con ejemplos y plantillas tanto en Word como en Latex del formato: LNCS:

- LNCS Latex: <https://goo.gl/tKiYW5>
- LNCS Word: <https://goo.gl/xw5i8d>
- LNCS Word 2007: <https://goo.gl/aa4of4>

Si el trabajo está aprobado el alumno está en condiciones de rendir el coloquio, que será una defensa oral del trabajo.