



Nombre del Alumno:



Examen de Sistemas Inteligentes (Prueba de concepto)

21 de mayo de 2024

Instrucciones generales para realizar el examen:

- **Pon tu nombre en la cabecera**
 - **Contesta a las preguntas digitalmente usando Word.** Puedes apoyarte en tantas hojas en sucio adicionales como quieras, pero las respuestas deben ir en el examen de forma digital.
 - **El examen se entrega en formato PDF para evitar modificaciones.** Para guardar el formato pdf ir a Archivo>exportar>crear pdf.
 - El tiempo para la realización del examen es de **3h**. Consta de dos partes, una teórica con problemas que se contestan en el Word de 5 puntos divididos en dos preguntas de 2.5 puntos. Y una práctica que se contesta usando Júpiter Notebook. La entrega debe contener un zip con lo siguiente en la raíz del zip: El pdf con el enunciado y las respuestas de la parte teórica, el notebook con la implementación de la parte práctica y los datos. La parte práctica debe ejecutar, así que cuidado con las rutas (hacedlas relativas).
-

Problema 1 (2,5 p)

En una ciudad existen 5 puntos negros (zonas con alta tasa de accidentes) situados en coordenadas X e Y en una cuadrícula de tamaño 10x10.

Estos puntos negros implican para su subsanación la movilización de ciertos recursos.

- Grúa: El ayuntamiento dispone de 3.
- Operarios: El ayuntamiento dispone de 10 operarios.
- Hormigonera: el Ayuntamiento dispone de 2.

La grúa y las hormigoneras deben estar accionadas mediante 1 operario.

Los 5 puntos negros son numerados con letras y requieren los siguientes recursos para poder realizarse en 1 día. El objetivo es asignar eficientemente los recursos para resolver el mayor número de puntos negros en un día.



A: 4 operarios sin hormigonera y 2 con hormigonera.

B: 1 operario y 2 grúas o 6 operarios sin grúas.

C: 1 operario, 1 grúa y 1 hormigonera o bien 3 operarios y 1 hormigonera o 8 operarios sin hormigonera ni grúa.

D: 5 operarios y 2 hormigoneras o 9 operarios

E: 1 grúa (indispensable) y 4 operarios o bien 2 grúas y 2 operarios.

Se pide:

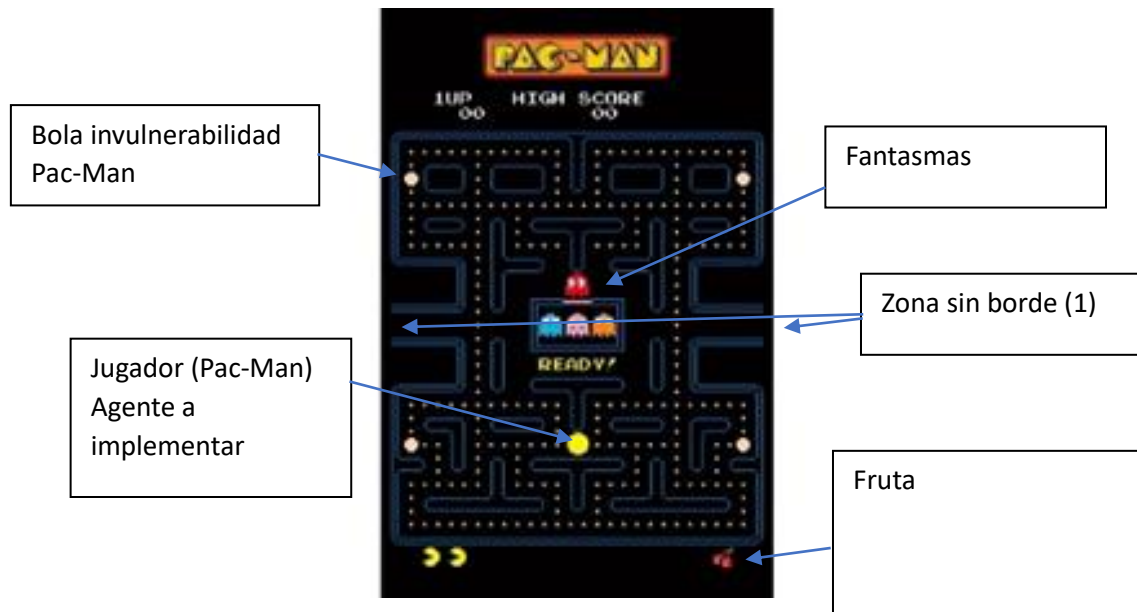
Modelar el problema con un algoritmo que permita resolver el problema planteado. Indica cuál es el que eliges y por qué crees que es el más recomendado. (2,5p)

Problema 2 (2,5 p)

Queremos modelar un agente autónomo que sustituya al jugador en el videojuego de Pac-Man. El jugador en este videojuego controla un personaje en forma de esfera amarilla con una abertura que simula la boca del personaje. Su movimiento es automático en la última dirección presionada por el jugador. Si llega a una pared el jugador se detiene. El objetivo del juego es comerse todas las bolitas amarillas que hay en el escenario para tener más puntuación, evitando que te toquen los fantasmas que son tus enemigos. Hay unas bolitas amarillas más grandes que otorgan invulnerabilidad al jugador durante cierto tiempo. Durante ese tiempo, el jugador si toca a alguno de los enemigos se lo puede comer. Una parte del escenario (Véase 1 en el dibujo) no tiene límite en el borde, lo que permite salirse del mismo y entrar por el lado contrario del escenario.

Los fantasmas van hacia el jugador siempre que el escenario se lo permita. El comportamiento del jugador que queremos modelar es el siguiente.

- El agente jugador intentará comerse todas las bolitas pequeñas.
- Si detecta que hay un fantasma cerca, pero está más cerca una bola grande, se comerá la bola grande.
- Si choca con una pared, cambiará de dirección de forma aleatoria.
- Si el jugador tiene invulnerabilidad (se ha comido una bola amarilla grande), se moverá en la dirección del fantasma más cercano.
- Si el fantasma se come al jugador, este desaparece durante unos segundos y reaparece en el centro del escenario.
- Si ve que hay fantasmas cerca, se mueve en la dirección contraria al fantasma
- Si hay una fruta a menos de 2 cuadros de distancia, da igual el comportamiento que estuviera ejecutando que intenta comerse la fruta. Después volverá a comerse las bolitas pequeñas.



Se pide:

1. ¿Qué tipo de agente es? simple, basado en modelo, basado en objetivo, etc. Razona la respuesta. (0.5 p)
2. Formaliza el agente usando REAS (0,5p)
3. Modela el agente con el formalismo que consideres más adecuado (1,5 p)

Ejercicio práctico: (5 puntos)

En el laboratorio, se entrega un Jupiter Notebook con las respuestas. El dataset está subido al campus.

Disponemos de un dataset sobre demencia con los siguientes campos:

- Subject ID: identificación del paciente.
- MRI ID: identificación de la visita al médico.
- Group: Grupo al que pertenece, que puede ser:
 - Demented: Ha sido diagnosticado con la enfermedad desde el principio.
 - Nondemented: No ha sido diagnosticada la enfermedad en ningún momento.
 - Converted: A 14 pacientes se le diagnosticó mal inicialmente y acabaron teniendo demencia.



- Visit: número de visita.
- MR Delay: Deley desde la última consulta.
- M/F: Género.
- Hand: Si son diestros o no.
- Age: Edad

Y los siguientes valores que son pruebas médicas realizados a los pacientes:

SES,MMSE,CDR,eTIV,nWBV,ASF

En el dataset hay varios pacientes, cada uno de ellos han visitado el médico al menos 2 veces. Queremos construir un modelo que prediga el estado del paciente entre Demented, Nondemented y Converted. Con esto podremos identificar a tiempo sobre todo aquellos pacientes a los que las pruebas diagnósticas iniciales indicaba que no eran dementes, pero que acabaron siéndolo.

Ejercicio 1 (1 punto): Limpia el dataset de errores y elementos que no sean relevantes. Justifica en una celda de markdown las decisiones que has tomado.

Ejercicio 2 (1.5 puntos): Usa el Perceptrón multicapa de SKLearn. El resultado mínimo que debéis conseguir de precisión es de un 65%. Dejad bien claro cuál es el modelo con el que finalmente os quedáis.

Ejercicio 4 (1.5 puntos): Usa el algoritmo Decision Tree de Sklearn para construir un segundo modelo. Calcula accuracy y la matriz de confusión. El resultado mínimo que debéis conseguir de precisión es de un 85% (random_state=42)

Ejercicio 4 (1 Punto) compara los resultados de ambos y justifica con cual te quedarías.