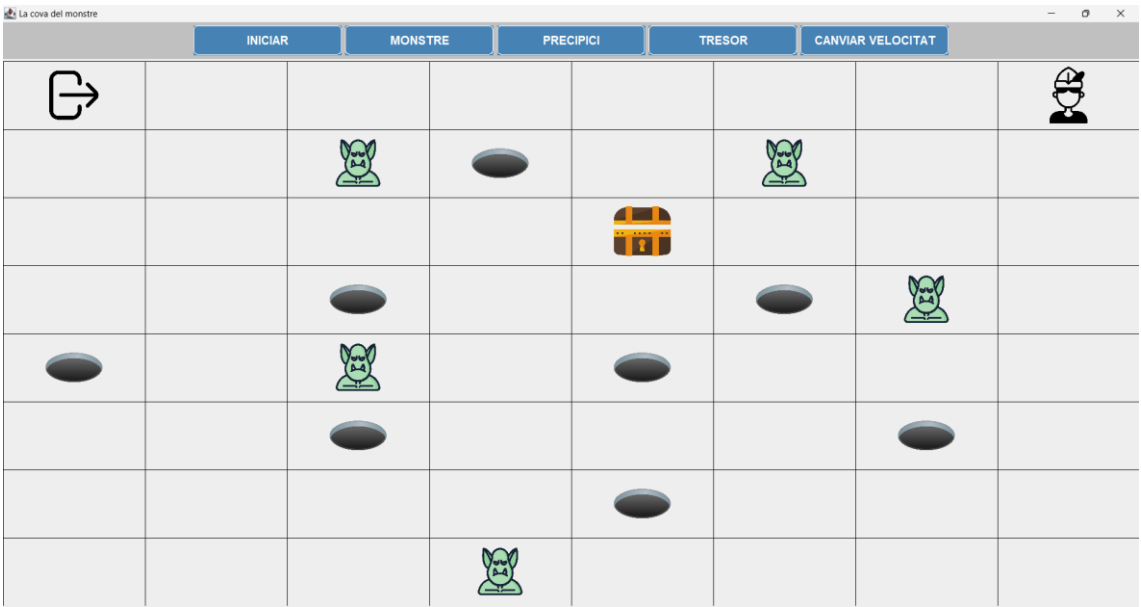


La cova del monstre

Curs 2024 - 2025



Marc Llobera Villalonga
DNI - 43461915B

Descripció de l'enunciat

Aquesta pràctica tracte de desenvolupar un programa que simula el problema de la cova del monstre. Es crea un tauler de caselles que simula la cova en si. La casella inicial [1,1] es l'entrada de l'agent, aquest explorarà la cova cercant un tresor, però a la cova poden haver diversos monstres i diversos precipicis, aquests son obstacles per els quals l'agent no pot passar. L'agent té la capacitat de recordar les caselles que son obstacles i les caselles que ja han estat visitades. Quan ha trobat el tresor tornarà a la casella inicial per sortir de la cova i finalitzar l'execució del programa.

La pràctica ha de contenir una interfície gràfica d'usuari on es representin, mitjançant una quadrícula que il·lustra l'entorn, els elements de l'entorn del problema i on s'il·lustri el recorregut de l'agent a través de l'ambient.

- L'usuari serà qui situï els precipicis, el monstre i el/s tresor/s inicialment a l'entorn.
- El recorregut de l'agent s'ha de fer a una velocitat adequada que permeti l'observació del funcionament del mateix. S'aconsella fer servir dues o més velocitats de l'agent en el recorregut, una de les quals ha de ser manual, perquè l'usuari comprovi el bon funcionament de l'agent.

1. Característiques de cada component del REAS del agent

- **Rendiment:** Trobar el tresor de manera eficient, evitant obstacles i minimitzant el nombre de moviments. Tornar a la casella inicial [1,1] després de trobar el tresor. Evitar caure en precipicis o ser atrapat pel monstre. Explorar de manera sistemàtica la cova, sense repetir innecessàriament caselles ja visitades. L'agent tindrà un millor rendiment com més ràpid aconsegueixi trobar el tresor i tornar a la sortida, minimitzant el risc i l'exploració redundant. El rendiment de l'agent es mesura per la seva capacitat per: Trobar el tresor a l'entorn. Evitar perills com monstres i precipicis. Tornar a la posició inicial un cop trobat el tresor. Minimitzar el nombre de moviments necessaris per completar la seva tasca. L'agent utilitza estratègies com: Mapeig de l'entorn (mapaPercepcions, mapaPosibles, mapaOk, mapaPerills). Registre de caselles visitades per evitar repeticions innecessàries. Algoritme A* per trobar el camí més curt de tornada a l'inici.
- **Entorn:** Una quadrícula que representa la cova. Caselles que poden contenir: Entrada/sortida [1,1], Espais buits transitables, Precipicis (obstacles), Monstre(s) (obstacles), Tresor(s). L'entorn és parcialment observable, ja que l'agent només pot percebre les caselles adjacents. És estàtic, ja que els elements no es mouen durant l'exploració de l'agent. És discret, amb moviments i accions definits per caselles.
- **Actuadors:** Moviment en quatre direccions: amunt, avall, esquerra, dreta. Capacitat de romandre en la mateixa posició. Habilitat per recollir el tresor quan el troba. Capacitat de marcar caselles com a visitades o obstacles en la seva memòria interna.
- **Sensors:** Detectar el contingut de les caselles adjacents (buida, precipici, monstre, tresor). Percebre la seva posició actual en relació amb la casella inicial [1,1]. Reconèixer quan ha trobat el tresor. Mantenir un registre de les caselles visitades i els obstacles trobats. Detectar quan ha tornat a la casella inicial després de trobar el tresor. Els sensors de l'agent li permeten percebre: Hedor (indica la presència propera d'un monstre) Brisa (indica la presència propera d'un precipici) Resplendor (indica la presència del tresor a la casella actual).

2. Etapa Deductiva

En aquesta etapa, l'agent actualitza la base de coneixements utilitzant la informació percebuda de l'entorn.

Variables emprades:

mapaPercepcions: Mapa que emmagatzema les percepcions (hedor, brisa, resplendor) per a cada posició.

mapaPosibles: Mapa que registra possibles perills (monstre, precipici) a cada posició.

mapaOk: Llista de posicions segures.

mapaPerills: Mapa que registra perills confirmats a cada posició.

Regles de producció (pseudocodi):

funció actualitzar_base_coneixements():

```
registrar_ok(fila_actual, columna_actual)

registrar_visita(fila_actual, columna_actual)

percepcions = obtenir_percepcions(fila_actual, columna_actual)

registrar_percepcions(fila_actual, columna_actual, percepcions)

actualitzar_possibles_perills(percepcions)

actualitzar_base_coneixements_completa()
```

funció actualitzar_possibles_perills(percepcions):

per a cada adreça adjacent:

nova_posició = calcular_nova_posició(direcció)

si nova_posició no és fora de l'escenari i no és segura i no és perill confirmat:

si no hi ha pudor i no hi ha brisa:

marcar_com_segura(nova_posició)

sinó:

registrar_posible_perill(nova_posició, percepcions)

funció actualitzar_base_coneixements_completa():

per a cada posició a l'escenari:

si posició no és perill confirmat:

si posició té percepcions:

si hi ha pudor:

registrar_perill(posició, MONSTRU)

si hi ha brisa:

registrar_perill(posició, PRECIPICI)

si posició té possibles perills:

verificar_i_actualitzar_perills(posició)

3. Etapa Reactiva

En aquesta etapa, l'agent determina les accions a prendre basant-se en la base de coneixements actualitzada.

Variables emprades:

teTressor: Booleà que indica si l'agent ha trobat el tresor.

camí: Pila que emmagatzema el camí recorregut per l'agent.

visitats: Mapa que registra el nombre de visites a cada posició.

Regles de producció (pseudocodi):

funció determinar_acció():

 si té_tresor:

 tornar_a_inici()

 sinó:

 si percep_resplandor:

 recollir_tresor()

 sinó:

 nova_posició = trobar_nova_posició_vàlida()

 si nova_posició no és nul·la:

 moure_a(nova_posició)

 sinó:

 retrocedir()

funció trobar_nova_posició_vàlida():

 posicions_segures = []

 posicions_risc = []

 per a cada adreça adjacent:

 nova_posició = calcular_nova_posició(direcció)

 si nova_posició és segura:

 afegir a posicions_segures

 sinó si nova_posició té possible perill:

 afegir a posicions_risc

 si posicions_segures no és buida:

 tornar posició_segura menys visitada

 sinó si posicions_risc no és buida:

 retornar primera posició_risc

 sinó:

 tornar nul

funció tornar_a_inici():

si posició_actual és inici:

finalitzar_recerca()

sinó:

camí = calcular_camí_volta()

si camí no és nul i no és buit:

moure_a(primer_posició_del_camí)