

Proyecto: Etapa 3

Inteligencia Artificial

Germán Gómez - Tomás Miguel

22/11/2018

Índice

Implementación	2
Deseos	2
Altas Prioridades	3
Selección de intención	6
Planificación	7
Achieved	9
Actions representation and projection	10
Consideraciones	11

Implementación

El ciclo del agente se basa en la arquitectura BDI.

Deseos

Representan el estado de motivación del agente, esto es, los objetivos o situaciones que el agente quisiera lograr o alcanzar (donde la prioridad se determina por el orden de declaración). El agente considera como deseos lo siguiente:

- **get(Oro):** El agente chequea su base de creencias, y en caso de conocer un oro, es agregado a la lista de deseos, el deseo de tomar ese oro.
- **get(Potion):** El agente chequea su base de creencias, y en caso de conocer una poción, es agregado a la lista de deseos, el deseo de tomar esa poción.
- **get(Tumba):** El agente chequea su base de creencias, y en caso de conocer una tumba con al menos un oro dentro, y si además el agente posee tres pociones, entonces el deseo de abrir dicha tumba es agregado a la lista de deseos.
- **depositar(Posada,MisOros):** Si el agente posee oros, agrega el deseo de ir a su propia base y depositar los oros. Para esto, accede a su base de creencias para obtener el home del agente. De esta manera el deseo de ir al home a dejar oros es agregado a la lista de deseos. Es importante aclarar en este punto, que el agente agrega el deseo, si tiene uno o más oros, ya que se contempla el caso en que le quede solo un oro, y este sea el último que pueda tomar en el contexto de la partida.
- **saquear(Home):** el agente chequea su base de creencias y verifica:
 - conocer el home enemigo,
 - que el home enemigo posea oros.

Si se cumplen estas condiciones, el agente agrega a su lista de deseos, saquear el home rival.

- **rest:** Si el agente tiene menos de 100 puntos de vida, el deseo de ir a descansar a una posada es agregado a la lista de deseos.
- **hechizar(Enemigo):** el agente chequea su base de creencias y verifica:
 - tener una poción,
 - conocer un enemigo,
 - que el enemigo tenga una poción,
 - que el enemigo tenga vida.

Si se cumplen estas condiciones, el agente agregar a su lista de deseos, hechizar a un rival. Es importante aclarar que este tipo de hechizo es por precaución a que el enemigo me hechice a mi.

- **explorar:** Si el agente llega a esta instancia, agrega a la lista de deseos el objetivo de explorar nuevos nodos del mapa, aun no conocidos.
- **asesinar(Enemigo):** el agente agrega el deseo de perseguir un enemigo hasta matarlo si el enemigo:
 - es conocido en mi base de creencias,
 - tiene vida,
 - tiene 3 o más oros,
 - está en rango de ataque.

Si se cumplen estas condiciones, el agente agregar a su lista de deseos, asesinar a un rival.

- **move_at_random:** Si el agente llega a esta instancia, agrega a la lista de deseos el objetivo de moverse random por el mapa, de manera de siempre estar en movimiento.

Altas Prioridades

Determina si existe una acción prioritaria que debe realizar el agente interrumpiendo la intención actual. El orden en que se presentan a

continuación determina la prioridad de las acciones. Por ejemplo, si se activa el hechizar, solamente se puede activar la alta prioridad de rest, la cual está por encima de hechizar.

- **rest:** si el agente tiene menos de 70 de vida y conoce al menos una posada, va a descansar.
- **hechizar(Enemigo):** el agente chequea su base de creencias y verifica:
 - tener una poción,
 - conocer un enemigo con una poción,
 - que el enemigo tenga vida,
 - que el enemigo esté en posición de ataque.

Si se cumplen estas condiciones, el agente determina que es urgente hechizar al rival. Es importante aclarar que este tipo de hechizo es por precaución a que el enemigo me hechice a mí.

- **atacar(Enemigo):** el agente chequea su base de creencias y verifica:
 - conocer un enemigo,
 - tener más vida que el enemigo,
 - tener más experiencia que el enemigo,
 - que esté en posición de ataque.

Si se cumplen estas condiciones, el agente determina que es urgente atacar al rival. Es importante aclarar que este tipo de ataque se realiza hasta que deja de estar en rango de ataque.

- **huir:** si el agente no está en una posada, tiene menos de 110 de vida, y hay un enemigo cerca, que tiene mayor vida y que está en rango de ataque, entonces nuestro agente huye a la posada más cercana para evitar ser atacado y a la vez recomponer su salud o vida.
- **asesinar(Enemigo):** el agente chequea su base de creencias y verifica si el enemigo:

- es conocido,
- tiene vida,
- tiene 2 o más oros,
- que esté en posición de ataque.

Si se cumplen estas condiciones, el agente determina que es urgente atacar al rival. Es importante aclarar que este tipo de ataque se realiza hasta que el agente rival muera.

● **asesinar(Enemigo):** el agente chequea su base de creencias y verifica:

- que el tiempo sea mayor a 2200,
- conocer un enemigo,
- que el enemigo tenga vida,
- que el enemigo tenga 1 o más oros o pociones,
- que el enemigo esté en posición de ataque.

Si se cumplen estas condiciones, el agente determina que es urgente atacar al rival. Es importante aclarar que este tipo de ataque se realiza hasta que el agente rival muera.

Selección de intención

Representan el estado deliberativo del agente, esto es, lo que el agente ha elegido hacer. Toda intención verifica que se encuentre el deseo en la lista de deseos. Selecciona una intención con el siguiente orden de prioridad:

1. **rest:** el agente tiene menos de 90 de vida, por lo que decide ir a descansar antes de encarar un nuevo deseo.
2. **depositar(Home,ListaOros):** Si la cantidad de oros es mayor al umbral establecido, entonces el agente adopta la intención de ir al propio home a dejar los oros. Nuevamente chequea que en su base de creencias se conozca la posición del home local. El umbral establecido para dejar oros en el home es válido al tener más oros que oros que deseo recoger en el mapa; por ejemplo, si sé que hay 4 oros en mis deseos, pero llevo 5 oros conmigo, es mejor depositar esos 5 oros.
3. **saquear(Home):** si conozco la posición del home enemigo y tiene oros, busco si existe un plan hacia ella, siempre y cuando mi agente tenga al menos una poción.
4. **hechizar(Enemigo):** si tengo 2 pociones, un enemigo tiene una poción, mi enemigo tiene más de 0 de vida, y está en posición de ataque, entonces mi nueva intención es hechizar a mi enemigo.
5. **get([potion,Nombre]):** el agente adopta la intención de ir a recoger la poción más cercana que percibe o conoce a su alrededor, pero con la condición de que se encuentre como máximo a 40 de distancia. A este tipo de poción la denominaremos poción cercana, y a las que están a más de 40 de distancia se la llamará poción lejana.
6. **get(Obj):** El agente adopta la intención de ir a recoger/abrir el objeto más cercano entre los distintos oros, pociones, o tumbas que percibe o conoce a su alrededor. Aclaración: el agente ve con la misma importancia un oro, una tumba o una poción lejana y al momento de planificar, genera un plan acorde a tomar un objeto o abrir un objeto.
7. **depositar(Home,ListaOros):** el agente adopta la intención de dejar los oros en el home si conoce su home y tiene oros.

8. **explorar:** El agente adopta la intención de seguir explorando el entorno.
9. **move at random:** El agente adopta la intención de moverse random por el mapa.

Planificación

Los planes son secuencias de acciones que un agente puede realizar para alcanzar su intención actual. Los planes pueden incluir otros planes, por ejemplo: un plan para ir a un nodo, y otro para tomar el oro de dicho nodo.

De esta manera se genera el plan de cada intención:

- **hechizar(Enemigo):** se verifica que el agente tenga una poción P y esté en rango de ataque, luego se genera un plan con la acción *'cast_spell(sleep(Enemigo,[potion,P]))'*.
- **atacar(Enemigo):** realiza el ataque al Enemigo y luego llama recursivamente a atacar(Enemigo). El caso base sería el achieved relacionado a atacar(Enemigo).
- **asesinar(Enemigo):** realiza dos ataques al Enemigo, se dirige a la última posición conocida del enemigo y luego llama recursivamente a asesinar(Enemigo). El caso base sería el achieved relacionado a asesinar(Enemigo).
- **saquear(Home):** se verifica que el agente tenga una poción P y conocer la posición del Home. Luego genero un plan que consiste de un camino hacia el home y la acción *'cast_spell(open(Home,[potion,P]))'*.
- **Get(Tumba):** busco la posición de la tumba, verifiqué que tenga oros y verifico tener una poción P. Luego genero un plan hacia el nodo de la tumba y lo concateno a la acción de abrir la tumba *'cast_spell(open([grave,NombreTumba],[potion,P]))'*.

- **Get(Oro):** busco la posición del oro. Luego genero un plan hacia el nodo del oro y lo concateno a la acción de tomar el oro.
- **Get(Poción):** busco la posición de la poción. Luego genero un plan hacia el nodo de la poción y lo concateno a la acción de tomar la poción.
- **goto(Pos):** el agente busca un plan de desplazamiento a la posición Pos.
- **rest:** genero una lista de posadas permitidas, busco un plan hacia una de ellas. Luego concateno el plan a la posada con la acción de stay, para que en el destino el agente cargue su vida.
- **depositar(Home,ListaOros):** genero un plan que consiste en ir a la posición del home, dejar un oro con la acción drop y luego llamar recursivamente a depositar con la lista de oros restantes. El achieved relacionado a esta acción es el caso base, el cual consiste en tener todos los oros que se tenían que depositar, en el home propio.
- **explorar:** genero una frontera de nodos que no conozco y busco un plan hacia uno de ellos.
- **stay:** genero un plan compuesto por no hacer nada y stay nuevamente.
- **move_at_random:** busco un nodo que conozco, y genero un plan hacia el.
- **huir:** Planificación similar a **rest**, donde genero una lista de posadas a las que puedo acceder y busco un plan hacia una de ellas. Luego concateno el plan a la posada con la acción de stay, para que en el destino el agente cargue su vida.

Achieved

Determina si la intención fue alcanzada. Esto es, si se verifica de acuerdo a las creencias del agente.

- **rest:** si mi vida es mayor a la vida total que puedo tener menos 10, entonces tengo que dejar de descansar.
- **depositar(Home, Oros):** si todos los oros que tenía, ahora los tiene la home, entonces ya termine de depositar.
- **get([grave,Grave]):** si la tumba no tiene más oros, entonces ya la abrí.
- **get(Obj):** si el objeto lo tiene mi agente, entonces ya lo agarré. Este achieved sirve para cuando quiera juntar un oro o una poción.
- **saquear(Home):** si el home enemigo no tiene más oros, entonces ya lo ataqué.
- **hechizar(Enemigo):** si el enemigo no tiene oros, no tiene pociones y no tiene vida, entonces ya fue hechizado.
- **goto(Pos):** si mi agente esté en esa posición, entonces ya llegué.
- **huir:** Similar a un descanso normal en la posada, pues verifico que mi vida sea mayor a la vida total que puedo tener menos 10.
- **atacar(Enemigo):** si el agente enemigo ya no está en rango o no tiene vida, hay que dejar de atacarlo.
- **asesinar(Enemigo):** si el agente enemigo no tiene vida, hay dejar de atacarlo.

Actions representation and projection

Determina si una acción puede llevarse a cabo y qué creencias se agregan o eliminan de mi base de conocimientos. Las siguientes dos fueron agregadas al juego:

- **drop(Obj):**
 - Precondiciones: tener el objeto y que el agente esté en un nodo.
 - Agregar: el objeto en el nodo donde estaba el agente.
 - Eliminar: el agente tiene el objeto .
- **cast_spell(open(Grave,Potion)):**
 - Precondiciones:
 - Que la tumba este en un nodo N.
 - Que el agente esté en un nodo N.
 - Que el agente tenga una poción.
 - Agregar:
 - Agregar para cada Oro de la tumba, que está en un nodo.
 - Eliminar:
 - La tumba tiene los oros
 - El agente tiene la poción

Consideraciones

Se corrigió de la etapa anterior:

- Valores para la selección de la intención rest mal configurados.
- No se especificó en STRIPS la acción drop.

El nombre del agente es 'cátulo'.

Se identificó un problema en el entorno brindado por la cátedra. Al momento de abrir una tumba, los oros liberados quedan en el mismo vector. Cuando el agente se encuentra en el mismo nodo que los oros e intenta tomar uno de ellos, las precondiciones fallan, por lo cual agarra el otro oro que se encontraba en ese vector. Luego de esto, en vez de querer tomar el oro en el cual las precondiciones habían fallado, busca uno más lejano, lo toma y recién ahí busca el oro que había fallado en primera instancia. Por ejemplo los oros 21, 22, 23 que estaban en la tumba:

```
Agent: agentBDI      time: 1762      life: 401 / 450      skill: 120      Pos: 1404

I remember:
[potion,p6] at 4 (vector(7.407768,2.534575,1.533943)). Descr: [[weight,2]].
[gold,gold19] at 94 (vector(5.0.2857494,5)). Descr: [[weight,2]].
[inn,inn1] at 143 (vector(9.7,-0.5,6.7)). Descr: [[forbidden_entry,[]]].
[gold,gold28] at 148 (vector(20.70085,0.7.194546)). Descr: [[weight,2]].
[potion,p27] at 421 (vector(13.7413,2.534575,18.68477)). Descr: [[weight,2]].
[grave,grave0] at 1151 (vector(2.831895,0.2620876,51.3806)). Descr: [].
[gold,gold20] at 1337 (vector(5.6968,3.11844,58.60737)). Descr: [[weight,2]].
[potion,p25] at 429 (vector(30.6061,2.534575,18.26314)). Descr: [[weight,2]].
[inn,inn2] at 790 (vector(16.2977,-0.5,35.04412)). Descr: [[forbidden_entry,[[age]]]].
[home,homeRed] at 1112 (vector(16.36307,-0.8545628,49.09412)). Descr: []. Has: [gold,gold21,gold,gold22,gold,gold23].
[agent,me] at 1404 (vector(48.79533,0.7,61.06507)). Descr: [[life,401],[lifeTotal,450],[skill,120],[lastAction,[move(1404),1750]],home,homeRed].
[gold,gold1] at 1230 (vector(68.00223,0.03,53.07312)). Descr: [[weight,2]].
[gold,gold21] at 1404 (vector(46.76585,2.762088,60.32346)). Descr: [[weight,2]].
[gold,gold22] at 1404 (vector(46.76585,2.762088,60.32346)). Descr: [[weight,2]].
[gold,gold23] at 1404 (vector(46.76585,2.762088,60.32346)). Descr: [[weight,2]].
[gold,gold7] at 1223 (vector(54.14717,0.03,53.60359)). Descr: [[weight,2]].
[gold,gold8] at 1390 (vector(15.05559,0.52,24075)). Descr: [[weight,2]].
[grave,grave3] at 1403 (vector(46.76585,0.2620876,60.32346)). Descr: [].
[potion,p0] at 903 (vector(57.65839,2.534575,38.28077)). Descr: [[weight,2]].
```

Oros 2 y 3:

```

Agent: agentBDI      time: 368      life: 408 / 450      skill: 110      Pos: 1152

I remember:
[gold,gold16] at 282 (vector(11.34207,3.11844,13.56117)). Descr: [[weight,2]].
[potion,p25] at 429 (vector(30.6061,2.534575,18.26314)). Descr: [[weight,2]].
[potion,p27] at 421 (vector(13.7413,2.534575,18.68477)). Descr: [[weight,2]].
[agent,me] at 1152 (vector(4.850712,0.9029515,50.97746)).
  Descr: [[life,408],[lifeTotal,450],[skill,110],[lastAction,[move(1152),358]],[home,homeRed]].
[gold,gold8] at 1390 (vector(15.05559,0.52.24075)). Descr: [[weight,2]].
[gold,gold10] at 934 (vector(27.69813,0.03,41.66537)). Descr: [[weight,2]].
[gold,gold9] at 1339 (vector(10.96229,0.03,59.14618)). Descr: [[weight,2]].
[gold,gold20] at 1337 (vector(5.6968,3.11844,58.60737)). Descr: [[weight,2]].
[gold,gold2] at 1198 (vector(2.831895,2.762088,51.3806)). Descr: [[weight,2]].
[gold,gold3] at 1198 (vector(2.831895,2.762088,51.3806)). Descr: [[weight,2]].
[inn,inn2] at 790 (vector(16.2977,-0.5,35.04412)). Descr: [[forbidden_entry,[]]].
[grave,grave0] at 1151 (vector(2.831895,0.2620876,51.3806)). Descr: [].
[home,homeRed] at 1112 (vector(16.36307,-0.8545628,49.09412)). Descr: [].
[potion,p24] at 1212 (vector(32.86953,2.534575,52.19806)). Descr: [[weight,2]].

Plan no factible. Fallarán las PRE de [pickup([gold,gold2]),[get([gold,gold2])]]
Current plan became infeasible.

Desires: [get([gold,gold16]),get([gold,gold8]),get([gold,gold10]),get([gold,gold9]),get([gold,gold20],p27]),get([potion,p24]),explorar,move_at_random]

New Intention: get([gold,gold2]), since es el objeto más cercano de los que deseo obtener

Following plan: [get([gold,gold2])]
    -- expanded into --
[goto(1198),pickup([gold,gold2]),[get([gold,gold2])]]
    -- expanded into --
[move(1198),[goto(1198)],pickup([gold,gold2]),[get([gold,gold2])]]

Next action: move(1198)

```

```

Agent: agentBDI      time: 372      life: 405 / 450      skill: 110      Pos: 1198

I remember:
[gold,gold16] at 282 (vector(11.34207,3.11844,13.56117)). Descr: [[weight,2]].
[potion,p25] at 429 (vector(30.6061,2.534575,18.26314)). Descr: [[weight,2]].
[potion,p27] at 421 (vector(13.7413,2.534575,18.68477)). Descr: [[weight,2]].
[agent,me] at 1198 (vector(4.31137,1.034687,52.11217)).
  Descr: [[life,405],[lifeTotal,450],[skill,110],[lastAction,[move(1198),370]],[home,homeRed]].
[gold,gold8] at 1390 (vector(15.05559,0.52.24075)). Descr: [[weight,2]].
[gold,gold10] at 934 (vector(27.69813,0.03,41.66537)). Descr: [[weight,2]].
[gold,gold9] at 1339 (vector(10.96229,0.03,59.14618)). Descr: [[weight,2]].
[gold,gold20] at 1337 (vector(5.6968,3.11844,58.60737)). Descr: [[weight,2]].
[gold,gold2] at 1198 (vector(2.831895,2.762088,51.3806)). Descr: [[weight,2]].
[gold,gold3] at 1198 (vector(2.831895,2.762088,51.3806)). Descr: [[weight,2]].
[inn,inn2] at 790 (vector(16.2977,-0.5,35.04412)). Descr: [[forbidden_entry,[]]].
[grave,grave0] at 1151 (vector(2.831895,0.2620876,51.3806)). Descr: [].
[home,homeRed] at 1112 (vector(16.36307,-0.8545628,49.09412)). Descr: [].
[potion,p24] at 1212 (vector(32.86953,2.534575,52.19806)). Descr: [[weight,2]].

Current Intention: get([gold,gold2])

Following plan: [pickup([gold,gold2]),[get([gold,gold2])]]

Next action: pickup([gold,gold2])

```

Ahora debería tomar el 3 que está en el lugar, pero:

```

Agent: agentBDI      time: 388      life: 405 / 450      skill: 110      Pos: 1198

I remember:
[gold,gold16] at 282 (vector(11.34207,3.11844,13.56117)). Descr: [[weight,2]].
[potion,p25] at 429 (vector(30.6061,2.534575,18.26314)). Descr: [[weight,2]].
[potion,p27] at 421 (vector(13.7413,2.534575,18.68477)). Descr: [[weight,2]].
[agent,me] at 1198 (vector(4.195977,0.8488343,52.47077)).
  Descr: [[life,405],[lifeTotal,450],[skill,110],[lastAction,[pickup(gold2),378]],[home,homeRed]].
  Has: [gold,gold2].
[gold,gold8] at 1390 (vector(15.05559,0.52.24075)). Descr: [[weight,2]].
[gold,gold10] at 934 (vector(27.69813,0.03,41.66537)). Descr: [[weight,2]].
[gold,gold9] at 1339 (vector(10.96229,0.03,59.14618)). Descr: [[weight,2]].
[gold,gold20] at 1337 (vector(5.6968,3.11844,58.60737)). Descr: [[weight,2]].
[gold,gold3] at 1198 (vector(2.831895,2.762088,51.3806)). Descr: [[weight,2]].
[inn,inn2] at 790 (vector(16.2977,-0.5,35.04412)). Descr: [[forbidden_entry,[]]].
[grave,grave0] at 1151 (vector(2.831895,0.2620876,51.3806)). Descr: [].
[home,homeRed] at 1112 (vector(16.36307,-0.8545628,49.09412)). Descr: [].
[potion,p24] at 1212 (vector(32.86953,2.534575,52.19806)). Descr: [[weight,2]].

Intention get([gold,gold2]) achieved.
Desires: [get([gold,gold16]),get([gold,gold8]),get([gold,gold10]),get([gold,gold9]),get([gold,gold20]),get([gold,gold3]),get([gold,gold24]),dejar([home,homeRed],[gold,gold2])]]

New Intention: get([gold,gold9]), since es el objeto más cercano de los que deseo obtener

Following plan: [get([gold,gold9])]
    -- expanded into -->
[goto(1339),pickup([gold,gold9]),[get([gold,gold9])]]
    -- expanded into -->
[move(1245),move(1292),move(1339),[goto(1339),pickup([gold,gold9]),[get([gold,gold9])]]]
Next action: move(1245)

```