**COMP 521 Assignment 3 (Zhiguo Zhang 260550226)**

Player is controlled by “WASD” for movement and “space” for teleport trap.

HTN illustration:

(1) World state: The world state is created at the start of game. It is updated based on the sensor information and the action performed by AI. Information includes:
    (a) IsGameOver
    (b) Score of each agent
    (c) Coins left with their position
    (d) Enemy position
    (e) NavMesh for possible path
(2) Sensor: The sensors of AI can detect
    (a) the position of 2 enemies
    (b) the position of player
    (c) number of coins left
    (d) current coin positions
    (e) current player and AI score
(3) Primitive tasks:
    (a) MoveTo(alcove): move to an alcove
        (i) Condition: the path to the alcove is complete, i.e. no obstacles or enemy field of view in between the AI and alcove
        (ii) Effect: Move AI to the alcove position
        (iii) Operator: pathfinding using navmesh to the alcove location
    (b) HideInAlcove(alcove): hide in an alcove
        (i) Condition: an enemy(s) field of view in between closest alcoves
        (ii) Effect: Move AI to the alcove position
        (iii) Operator: pathfinding using navmesh to the alcove location
    (c) CollectCoin: collect the coin in the current alcove
        (i) Condition: current alcove has a coin
        (ii) Effect:
            1) Move AI to the coin position
            2) AI score += 1
        (iii) Operator: pathfinding using navmesh to the coin location
    (d) UseTrap: use the teleport trap
        (i) Condition: Number of traps possessed > 0
        (ii) Effect:
            1) If closer to an enemy, de-and-respawn the enemy
            2) If closer to the player, teleport the player to a random alcove
        (iii) Operator: teleport is done by game manager
    (e) Finish: denote no further coins to collect
        (i) Condition: number of coins left = 0
        (ii) Effect: stop the AI from moving
        (iii) Operator: set “game over” flag by the game manager
(4) Compound tasks:
    (a) WinGame: win the game

(i) Method1
1) Condition: Coin left in game
2) Sub-tasks: CollectNearestCoin, CollectNearestCoin, …, Finish

(ii) Method2
1) Condition: No coin left in game
2) Sub-tasks: Finish

(b) CollectNearestCoin: collect the nearest coin

(i) Method1
1) Condition: no enemy in between AI and the nearest alcove with a coin
2) Sub-tasks: MoveTo(alcove with nearest coin), CollectCoin

(ii) Method2
1) Condition:
a) Number of traps > 0
b) a enemy in between AI and the nearest alcove with a coin
c) AI is closer to the enemy than to another agent
2) Sub-tasks: UseTrap, MoveTo(alcove with nearest coin), CollectCoin

(iii) Method3
1) Condition:
a) Number of traps > 0
b) a player is between AI and the nearest alcove with a coin (assuming the player is going to the alcove)
c) AI is closer to the player than to any enemy
d) Coin left < 3 (More likely to teleport the player to an alcove without coin)
2) Sub-tasks: UseTrap, MoveTo(alcove with nearest coin), CollectCoin

(iv) Method4
1) Condition: a enemy in between AI and the nearest alcove with a coin
2) Tasks: HideInAlcove(nearest alcove), MoveTo(alcove with nearest coin), CollectCoin

(5) Priority: The priority of each compound taks is given by the order of compound tasks. In this case, WinGame has higher priority than CollectNearestCoin. For different methods of a given compound task, method with smaller index has higher priority. For instance, Method4 of CollectNearestCoin has the lowest prioirty as it simply lets the AI wait in an alcove for the enemy to pass.

(6) For the planner, the WinGame is the goal task. It will decompose the WinGame tasks until all tasks are primitive tasks. For example, WinGame will become a sequence of CollectNearestCoin. Then, each CollectNearestCoin is decomposed to a sequence of primitive tasks. The AI will need a plan when (1) the current plan is finished or failed (2) it has no plan currently(e.g. Start of the game) (3) the world state changes which makes the current plan is no longer valid.