

```
;
; Modelling the Wumpus World in PDDL: 3rd time's a charm...
; by: Patrik Haslum
; Source web page:
;   http://users.cecs.anu.edu.au/~patrik/pddlman/wumpus.html
;

(define (domain wumpus-c)
  (:requirements :strips)
  (:predicates
    (at ?what ?square)
    (adj ?square-1 ?square-2)
    (pit ?square)
    (wumpus-in ?square)
    ;; <-> (exists ?x (and (is-wumpus ?x) (at ?x ?square) (not (dead ?x))
    (have ?who ?what)
    (is-agent ?who)
    (is-wumpus ?who)
    (is-gold ?what)
    (is-arrow ?what)
    (dead ?who))

  (:action move-agent
    :parameters (?who ?from ?to)
    :precondition (and (is-agent ?who)
                       (at ?who ?from)
                       (adj ?from ?to)
                       (not (pit ?to))
                       (not (wumpus-in ?to)))
    :effect (and (not (at ?who ?from))
                 (at ?who ?to))
  )

  (:action take
    :parameters (?who ?what ?where)
    :precondition (and (is-agent ?who)
                       (at ?who ?where)
                       (at ?what ?where))
    :effect (and (have ?who ?what)
                 (not (at ?what ?where)))
  )

  (:action shoot
    :parameters (?who ?where ?with-what ?victim ?where-victim)
    :precondition (and (is-agent ?who)
                       (have ?who ?with-what)
                       (is-arrow ?with-what)
                       (at ?who ?where)
                       (is-wumpus ?victim)
                       (at ?victim ?where-victim)
                       (adj ?where ?where-victim))
    :effect (and (dead ?victim)
                 (not (wumpus-in ?where-victim))
                 (not (have ?who ?with-what)))
  )
)
```

```
(:action move-wumpus
:parameters (?who ?from ?to)
:precondition (and (is-wumpus ?who)
                   (at ?who ?from)
                   (adj ?from ?to)
                   (not (pit ?to))
                   (not (wumpus-in ?to)))
:effect (and (not (at ?who ?from))
             (at ?who ?to)
             (not (wumpus-in ?from))
             (wumpus-in ?to))
)

)

(define (problem wumpus-c-1)
  (:domain wumpus-c)
  (:objects sq-1-1 sq-1-2 sq-1-3
            sq-2-1 sq-2-2 sq-2-3
            the-gold the-arrow
            agent wumpus)
  (:init (adj sq-1-1 sq-1-2) (adj sq-1-2 sq-1-1)
         (adj sq-1-2 sq-1-3) (adj sq-1-3 sq-1-2)
         (adj sq-2-1 sq-2-2) (adj sq-2-2 sq-2-1)
         (adj sq-2-2 sq-2-3) (adj sq-2-3 sq-2-2)
         (adj sq-1-1 sq-2-1) (adj sq-2-1 sq-1-1)
         (adj sq-1-2 sq-2-2) (adj sq-2-2 sq-1-2)
         (adj sq-1-3 sq-2-3) (adj sq-2-3 sq-1-3)
         (pit sq-1-2)
         (is-gold the-gold)
         (at the-gold sq-1-3)
         (is-agent agent)
         (at agent sq-1-1)
         (is-arrow the-arrow)
         (have agent the-arrow)
         (is-wumpus wumpus)
         (at wumpus sq-2-3)
         (wumpus-in sq-2-3))
  (:goal (and (have agent the-gold) (at agent sq-1-1)))
)
```

Resulting plan:

```
(MOVE-AGENT AGENT SQ-1-1 SQ-2-1)
(MOVE-AGENT AGENT SQ-2-1 SQ-2-2)
(SHOOT AGENT SQ-2-2 THE-ARROW WUMPUS SQ-2-3)
(MOVE-AGENT AGENT SQ-2-2 SQ-2-3)
(MOVE-AGENT AGENT SQ-2-3 SQ-1-3)
(TAKE AGENT THE-GOLD SQ-1-3)
(MOVE-AGENT AGENT SQ-1-3 SQ-2-3)
(MOVE-AGENT AGENT SQ-2-3 SQ-2-2)
(MOVE-AGENT AGENT SQ-2-2 SQ-2-1)
(MOVE-AGENT AGENT SQ-2-1 SQ-1-1)
```