```
;
; Modelling the Wumpus World in PDDL: 2nd try...
; by: Patrik Haslum
; Source web page:
;   http://users.cecs.anu.edu.au/˜patrik/pddlman/wumpus.html
;

(define (domain wumpus-b)
  (:requirements :strips)
  (:predicates
   (adj ?square-1 ?square-2)
   (pit ?square)

   (at ?what ?square)
   (have ?who ?what)

   (takeable ?what)
   (is-gold ?what)
   (is-arrow ?what)

   (alive ?who)
   (dead ?who))

  (:action move
    :parameters (?who ?from ?to)
    :precondition (and (alive ?who)
                       (at ?who ?from)
                       (adj ?from ?to)
                       (not (pit ?to)))
    :effect (and (not (at ?who ?from))
                 (at ?who ?to))
    )

  (:action take
    :parameters (?who ?what ?where)
    :precondition (and (alive ?who)
                       (takeable ?what)
                       (at ?who ?where)
                       (at ?what ?where))
    :effect (and (have ?who ?what)
                 (not (at ?what ?where)))
    )

  (:action shoot
    :parameters (?who ?where ?arrow ?victim ?where-victim)
    :precondition (and (alive ?who)
                       (have ?who ?arrow)
                       (is-arrow ?arrow)
                       (at ?who ?where)
                       (alive ?victim)
                       (at ?victim ?where-victim)
                       (adj ?where ?where-victim))
    :effect (and (dead ?victim)
                 (not (alive ?victim))
                 (not (at ?victim ?where-victim))
                 (not (have ?who ?arrow)))
    )
)
```

```
(define (problem wumpus-b-1)
  (:domain wumpus-b)
  (: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)
         (at the-gold sq-1-3)
         (is-gold the-gold)
         (takeable the-gold)
         (at agent sq-1-1)
         (alive agent)
         (have agent the-arrow)
         (is-arrow the-arrow)
         (takeable the-arrow)
         (at wumpus sq-2-3)
         (alive wumpus))
  (:goal (and (have agent the-gold)
              (at agent sq-1-1)
          ))
  )

Resulting plan:

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