

TAD Tablero:

obs baldosa : dict<pos,mov>
obs final : int;

TAD Oca:

obs pos1 int;
obs pos2 int;
obs turno int;
obs tablero : Tablero
obs dadosHistorial : dict(\mathbb{Z} , seq< \mathbb{Z} >)
obs quienGano : int
obs casillaHistorial . dict(\mathbb{Z} ,sez< \mathbb{Z} >)

proc Inicializar (in t: Tablero) : Oca

 requiere {true}
 asegura {res.pos1 = 0 \wedge res.pos2 = 0 \wedge res.turno = 1}
 asegura {res.dadosHistorial[1] = [] \wedge res.dadosHistorial[2] = [] \wedge res.tablero = t}
 asegura {res.casillaHistorial[1] = [] \wedge res.casillaHistorial[2] = []}

proc Avanzar (in dado : \mathbb{Z} , inout oca : Oca)

 requiere { $o_0 = o \wedge$ quienGano \neq 1 \wedge quienGano \neq 2}
 asegura {oca.tablero = oca₀.tablero}
 asegura {oca₀.turno = 1 \rightarrow_L oca.dadosHistorial = SetKey(oca₀.dadosHistorial, 1, concat(oca₀.dadosHistorial[1], [dado])}
 asegura {oca₀.turno = 2 \rightarrow_L oca.dadosHistorial = SetKey(oca₀.dadosHistorial, 2, concat(oca₀.dadosHistorial[2], [dado])}
 asegura {(oca₀.turno = 1 \wedge_L (oca.pos1 = oca₀.pos1 + oca₀.tablero.baldosa(oca₀.pos1 + dado) \wedge oca.pos2 = oca₀.pos2 \wedge oca₀.turno = 2 \wedge (oca.pos1 \geq oca.tablero.final \rightarrow_L oca.quienGano = 1) \wedge_L oca.pos1 < oca.tablero.final \rightarrow_L oca.quienGano = oca₀.quienGano))
 \vee (oca.turno = 2 \wedge_L oca.pos2 = oca₀.pos2 + oca₀.tablero.baldosa(oca₀.pos2 + dado) \wedge oca.pos1 = oca₀.pos1 \wedge oca.turno = 1 (oca.pos2 \geq oca.tablero.final \rightarrow_L oca.quienGano = 2) \wedge_L (oca.pos2 < oca.tablero.final \rightarrow_L oca.quienGano = oca₀.quienGano))}
 asegura {oca₀.turno = 1 \rightarrow_L
 oca.casillaHistorial = SetKey(oca₀.casillaHistorial, 1, concat(oca₀.casillaHistorial[1], [oca.pos1])}
 asegura {oca₀.turno = 2 \rightarrow_L
 oca.casillaHistorial = SetKey(oca₀.casillaHistorial, 2, concat(oca₀.casillaHistorial[2], [oca.pos2])}

proc dadoEnJugada (in nro : \mathbb{Z} , in oca: Oca, in jugador : \mathbb{Z}) : \mathbb{Z}

 requiere {(jugador = 1 \vee jugador = 2) \wedge (0 \leq nro < |oca.dadosHistorial[1]| \vee 0 \leq nro < |oca.dadosHistorial[2]|)}
 asegura {(jugador = 1 \wedge res = oca.dadosHistorial[1][nro]) \vee (jugador = 2 \wedge res = oca.dadosHistorial[2][nro])}

proc CasillaFuePisada (in jugador : \mathbb{Z} , in casilla : \mathbb{Z} , in oca : Oca) : Bool

 requiere {(jugador = 1 \vee jugador = 2) \wedge casilla \leq oca.tablero.final}
 asegura {jugador = 1 \rightarrow_L (res = true \leftrightarrow casilla \in oca.casillaHistorial[1])}
 asegura {jugador = 2 \rightarrow_L (res = true \leftrightarrow casilla \in oca.casillaHistorial[2])}

proc Ganador (in o : Oca) : Bool

 requiere {oca.quienGano = 1 \vee oca.quienGano = 2}
 asegura {res = oca.quienGano}