

File created: 15-Jul-88 16:02:58 {POGO:AINORTH:XEROX}<LOOPSCORE>TRUCKIN>PLAYERS>HIGHROLLE  
R.;2

changes to: (CLASSES HighRoller)  
(METHODS HighRoller.TakeTurn HighRoller.SellGoods HighRoller.GoToStoppingPlace  
HighRoller.FindStoppingPlace HighRoller.BuyGoods)  
(VARS HIGHROLLERCOMS)

previous date: 15-Sep-87 19:27:51 {POGO:AINORTH:XEROX}<LOOPSCORE>TRUCKIN>PLAYERS>HIGHROLLER.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1987, 1988 by Xerox Corporation. All rights reserved.

```
(RPAQQ HIGHROLLERCOMS ((DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT HIGHROLLER))
  (INSTANCES)
  (CLASSES HighRoller)
  (FNS HighRoller.BuyGoods HighRoller.FindStoppingPlace HighRoller.GoToStoppingPlace
    HighRoller.SellGoods HighRoller.TakeTurn)
  (METHODS HighRoller.BuyGoods HighRoller.FindStoppingPlace HighRoller.GoToStoppingPlace
    HighRoller.SellGoods HighRoller.TakeTurn)))

(DECLARE%: DONTCOPY

(PUTPROPS HIGHROLLER MAKEFILE-ENVIRONMENT (:PACKAGE "IL" :READTABLE "INTERLISP" :BASE 10))
)

(DEFCLASSES HighRoller)

(DEFCLASS HighRoller (MetaClass PlayerMeta doc "HighRoller is a player who specializes in LuxuryGoods. It has a
  simple endgame strategy which usually works. This is a slightly modified version of
  the winner of the knowledge competition in the first Loops course in January 1983
  (Toyota by Dick Lyon)" Edited%: (* mjs%:"18-FEB-83 19:54"))
  (Supers Traveler)
  (InstanceVariables (destination NIL doc "Current destination.")
    (direction NIL doc "Forward or Backward direction of travel on the highway.")
    (sitCount 0 doc "Number of turns sitting at Alices Restaurant during an EndGame.")
    (goal NIL doc "Buy, Sell, SitTight, or GoToAlices.)))

(DEFINEQ

(HighRoller.BuyGoods
  [LAMBDA (self)
    (CL:MACROLET [[ArgsOfMethodBeingCompiled NIL ''(self]
      (ClassNameOfMethodOwner NIL ''HighRoller)
      (SelectorOfMethodBeingCompiled NIL ''BuyGoods)
      (SelfOfMethodBeingCompiled NIL 'self)
      [_Super
        (&REST Send-Super-Args)
        (COND
          ((NULL Send-Super-Args)
            ' (DOAPPLY* (FindSuperMethod self 'BuyGoods (LOADTIMECONSTANT (OldClass HighRoller)))
              self))
          ((NEQ 'BuyGoods (CADR Send-Super-Args))
            (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
          ((NEQ 'self (CAR Send-Super-Args))
            (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
          (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'BuyGoods (LOADTIMECONSTANT (OldClass
                                                                                      HighRoller)
                                                                                      self)
              (CDDR Send-Super-Args]
            [_Super?
              (&REST Send-Super-Args)
              (COND
                ((NULL Send-Super-Args)
                  ' (DOAPPLY* (FindSuperMethod self 'BuyGoods (LOADTIMECONSTANT (OldClass HighRoller)
                    (FUNCTION NIL))
                    self))
                ((NEQ 'BuyGoods (CADR Send-Super-Args))
                  (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
                ((NEQ 'self (CAR Send-Super-Args))
                  (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
                (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'BuyGoods (LOADTIMECONSTANT (OldClass
                                                                                      HighRoller)
                                                                                      self)
                    (FUNCTION NIL))
                    self)
                  (CDDR Send-Super-Args]
              [_SuperFringe
                (&REST Send-Super-Args)
```

```

(COND
  [(NULL Send-Super-Args)
   (LIST 'for 'cls 'in (LIST 'fetch 'localSupers 'of (LIST 'LOADTIMECONSTANT
                                                         (LIST 'OldClass class-name)))
   'do
    (LIST* 'DOAPPLY* [LIST* 'OR (LIST 'FetchMethod 'cls (LIST 'QUOTE selector))
                          '((FUNCTION NIL)]
    (NCONC args]
  ((NEQ 'BuyGoods (CADR Send-Super-Args))
   (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
  ((NEQ 'self (CAR Send-Super-Args))
   (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
  (T (LIST* 'for 'cls 'in '(fetch localSupers of (LOADTIMECONSTANT (OldClass HighRoller)))
   'bind
    [LIST 'argList '_ (LIST* 'MAPCAR [LIST 'QUOTE (CONS (CAR Send-Super-Args)
                                                         (NCONC (CDDR Send-Super-Args)
                                                         '((FUNCTION EVAL]
    ' (do (APPLY (OR (FetchMethod cls 'BuyGoods)
                    (FUNCTION NIL))
            argList]
  (PROG (^auditRecord ^value commodity volumeCapacity weightCapacity volumeLimit weightLimit cashLimit
        producerLimit numBuy)
    ^firstRules
    [SETQ ^value (SETQ commodity (GetClassValue (GetValue (GetValue self 'truck)
                                                         'location)
                                                         'Commodity)]
    [COND
      ((NOT (SUBCLASS commodity ($ LuxuryGoods)))
       (SETQ ^value (PROGN
                    (RuleSetStop NIL 'NoBuy 'NotLuxury)
                    (GO QUIT]
        (* ^value set by RuleSetStop)
      [SETQ ^value (PROGN [SETQ volumeCapacity (DIFFERENCE (GetClassValue (GetValue self 'truck)
                                                         'MaxVolume)
                                                         (GetValue (GetValue self 'truck)
                                                         'volume]
        [SETQ weightCapacity (DIFFERENCE (GetClassValue (GetValue self 'truck)
                                                         'MaxWeight)
                                                         (GetValue (GetValue self 'truck)
                                                         'weight]
        [SETQ volumeLimit (QUOTIENT volumeCapacity (GetClassValue commodity
                                                         'VolPerQty]
        [SETQ weightLimit (QUOTIENT weightCapacity (GetClassValue commodity
                                                         'WtPerQty]
        [SETQ cashLimit (QUOTIENT (GetValue (GetValue self 'truck)
                                                         'cashBox)
                                   (PricePerUnit (GetValue (GetValue self 'truck)
                                                         'location)
        (SETQ producerLimit (GetValue (GetValue (GetValue self 'truck)
                                                         'location)
        'qty))
        (SETQ numBuy (MIN volumeLimit weightLimit cashLimit producerLimit]
    [COND
      ((GEQ numBuy 1)
       (SETQ ^value (PROGN [PROGN
                           (* Make an audit record for this rule and set its audit values.)
                           (SETQ ^auditRecord (_ ($ StandardAuditRecord)
                                                  New))
                           (PutValue ^auditRecord 'rule (GetObjFromUID '(LOX0.%.F5.o28.D_
                                                                      . 589]
                           (_ gameMaster Buy self numBuy)
                           (PutAuditRec self 'goal 'Sell ^auditRecord)
                           (PutAuditRec self 'destination (NearestRoadStop (Buyers commodity))
                           ^auditRecord]
       QUIT
      ^lastRules
      (RETURN ^value])

```

**(HighRoller.FindStoppingPlace**

```

[LAMBDA (self)
  (CL:MACROLET [[ArgsOfMethodBeingCompiled NIL ''(self]
                (ClassNameOfMethodOwner NIL ''HighRoller)
                (SelectorOfMethodBeingCompiled NIL ''FindStoppingPlace)
                (SelfOfMethodBeingCompiled NIL 'self)
  [_Super
   (&REST Send-Super-Args)
   (COND
     ((NULL Send-Super-Args)
      ' (DOAPPLY* (FindSuperMethod self 'FindStoppingPlace (LOADTIMECONSTANT (OldClass
                                                                               HighRoller))
                  self))
     ((NEQ 'FindStoppingPlace (CADR Send-Super-Args))
      (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
     ((NEQ 'self (CAR Send-Super-Args))
      (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
     (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'FindStoppingPlace (LOADTIMECONSTANT

```

```

(OldClass HighRoller)))
      self)
      (CDDR Send-Super-Args]
[_Super?
  (&REST Send-Super-Args)
  (COND
    ((NULL Send-Super-Args)
      ' (DOAPPLY* (FindSuperMethod self 'FindStoppingPlace (LOADTIMECONSTANT (OldClass
                                                                    HighRoller)))
      (FUNCTION NIL))
    (self))
    ((NEQ 'FindStoppingPlace (CADR Send-Super-Args))
      (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
    ((NEQ 'self (CAR Send-Super-Args))
      (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
    (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'FindStoppingPlace (LOADTIMECONSTANT
                                                                    (OldClass HighRoller)))
      (FUNCTION NIL))
      self)
      (CDDR Send-Super-Args]
[_SuperFringe
  (&REST Send-Super-Args)
  (COND
    [(NULL Send-Super-Args)
      (LIST 'for 'cls 'in (LIST 'fetch 'localSupers 'of (LIST 'LOADTIMECONSTANT
                                                                    (LIST 'OldClass class-name)))
      'do
        (LIST* 'DOAPPLY* [LIST* 'OR (LIST 'FetchMethod 'cls (LIST 'QUOTE selector))
          ' ((FUNCTION NIL)]
          (NCONC args]
        ((NEQ 'FindStoppingPlace (CADR Send-Super-Args))
          (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
        ((NEQ 'self (CAR Send-Super-Args))
          (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
        (T (LIST* 'for 'cls 'in ' (fetch localSupers of (LOADTIMECONSTANT (OldClass HighRoller)))
          'bind
            [LIST 'argList '_ (LIST* 'MAPCAR [LIST 'QUOTE (CONS (CAR Send-Super-Args)
                                                                    (NCONC (CDDR Send-Super-Args)
                                                                    ]
                                                                    ' ((FUNCTION EVAL]
          ' (do (APPLY (OR (FetchMethod cls 'FindStoppingPlace)
            (FUNCTION NIL))
            argList]
      (PROG (^auditRecord ^value wStation gasStation)
        ^firstRules
        [COND
          [(EQ (GetValue (GetValue self 'truck)
            'fuel)
            0)
            (SETQ ^value (PROGN [PROGN
              (* Make an audit record for this rule and set its audit values.)
              (SETQ ^auditRecord (_ ($ StandardAuditRecord)
                New))
              (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
                . 591]
              (PutAuditRec self 'stoppingPlace (AnyRoadStop ($ UnionHall))
                ^auditRecord]
              [[AND [LESSP (GetValue (GetValue self 'truck)
                'fuel)
                (TIMES 0.25 (GetClassValue (GetValue self 'truck)
                  'MaxFuel]
                (GREATERP (GetValue (GetValue self 'truck)
                  'cashBox)
                  0)
                (SETQ gasStation (NearestRoadStop (RoadStops ($ GasStation)
                  (_ self Rangel)
                  NIL
                  'Room]
                (SETQ ^value (PROGN [PROGN
                  (* Make an audit record for this rule and set its audit values.)
                  (SETQ ^auditRecord (_ ($ StandardAuditRecord)
                    New))
                  (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
                    . 592]
                  (PutAuditRec self 'stoppingPlace gasStation ^auditRecord]
                  [[AND [SETQ wStation (NearestRoadStop (RoadStops ($ WeighStation)
                    (_ self Rangel)
                    (GetValue self 'direction)
                    'Room]
                    (LESSP (Distance wStation)
                      (Distance (GetValue self 'destination)
                      (SETQ ^value (PROGN [PROGN
                        (* Make an audit record for this rule and set its audit values.)
                        (SETQ ^auditRecord (_ ($ StandardAuditRecord)
                          New))
                        (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
                          . 593]
                        (PutAuditRec self 'stoppingPlace wStation ^auditRecord]
                        [[AND (LEQ (Distance (GetValue self 'destination)
                          (_ self Rangel))

```

```

(RoomToParkP (GetValue self 'destination]
(SETQ ^value (PROGN [PROGN (* Make an audit record for this rule and set its audit values.)
                           (SETQ ^auditRecord (_ ($ StandardAuditRecord
                                                    New))
                           (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
                                                                    . 594]
                           (PutAuditRec self 'stoppingPlace (GetValue self 'destination)
                           ^auditRecord]
[[SETQ gasStation (FurthestRoadStop (RoadStops ($ GasStation)
                                             (_ self Rangel)
                                             (GetValue self 'direction)
                                             'Room]
(SETQ ^value (PROGN [PROGN (* Make an audit record for this rule and set its audit values.)
                           (SETQ ^auditRecord (_ ($ StandardAuditRecord
                                                    New))
                           (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
                                                                    . 595]
                           (PutAuditRec self 'stoppingPlace gasStation ^auditRecord]
(T (SETQ ^value (PROGN [PROGN (* Make an audit record for this rule and set its audit values.)
                              (SETQ ^auditRecord (_ ($ StandardAuditRecord
                                                       New))
                              (PutValue ^auditRecord 'rule (GetObjFromUID
                                                             ' (LOX0.%.F5.o28.D_; . 596]
                              (PutAuditRec self 'stoppingPlace (NthRoadStop (GetValue self
                                                                              'maxMove)
                                                                              (GetValue self 'direction)
                                                                              NIL
                                                                              'Room)
                              ^auditRecord]
QUIT
^lastRules
(RETURN ^value])

```

**(HighRoller.GoToStoppingPlace**

```

[LAMBDA (self)
  (CL:MACROLET [[ArgsOfMethodBeingCompiled NIL ''(self]
                (ClassNameOfMethodOwner NIL ''HighRoller)
                (SelectorOfMethodBeingCompiled NIL ''GoToStoppingPlace)
                (SelfOfMethodBeingCompiled NIL 'self)
                [_Super
                 (&REST Send-Super-Args)
                 (COND
                  ((NULL Send-Super-Args)
                   ' (DOAPPLY* (FindSuperMethod self 'GoToStoppingPlace (LOADTIMECONSTANT (OldClass
                                                                                               HighRoller))
                               self))
                  ((NEQ 'GoToStoppingPlace (CADR Send-Super-Args))
                   (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
                  ((NEQ 'self (CAR Send-Super-Args))
                   (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
                  (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'GoToStoppingPlace (LOADTIMECONSTANT
                                                                                               (OldClass HighRoller))))
                           self)
                   (CDDR Send-Super-Args]
                [_Super?
                 (&REST Send-Super-Args)
                 (COND
                  ((NULL Send-Super-Args)
                   ' (DOAPPLY* (FindSuperMethod self 'GoToStoppingPlace (LOADTIMECONSTANT (OldClass
                                                                                               HighRoller))
                               self))
                  ((FUNCTION NIL))
                  ((NEQ 'GoToStoppingPlace (CADR Send-Super-Args))
                   (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
                  ((NEQ 'self (CAR Send-Super-Args))
                   (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
                  (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'GoToStoppingPlace (LOADTIMECONSTANT
                                                                                               (OldClass HighRoller))
                               self)
                           self)
                   (CDDR Send-Super-Args]
                [_SuperFringe
                 (&REST Send-Super-Args)
                 (COND
                  ((NULL Send-Super-Args)
                   (LIST 'for 'cls 'in (LIST 'fetch 'localSupers 'of (LIST 'LOADTIMECONSTANT
                                                                           (LIST 'OldClass class-name))))
                   'do
                   (LIST* 'DOAPPLY* [LIST* 'OR (LIST 'FetchMethod 'cls (LIST 'QUOTE selector))
                                   ' ((FUNCTION NIL))
                   (NCONC args]
                  ((NEQ 'GoToStoppingPlace (CADR Send-Super-Args))
                   (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
                  ((NEQ 'self (CAR Send-Super-Args))
                   (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))

```

## (HighRoller.SellGoods

```
[LAMBDA (self)
  (CL:MACROLET
    [ [ArgsOfMethodBeingCompiled NIL '' (self)
      (ClassNameOfMethodOwner NIL '' HighRoller)
      (SelectorOfMethodBeingCompiled NIL '' SellGoods)
      (SelfOfMethodBeingCompiled NIL 'self)
      [_Super
        (&REST Send-Super-Args)
        (COND
          ( (NULL Send-Super-Args)
            ' (DOAPPLY* (FindSuperMethod self 'SellGoods (LOADTIMECONSTANT (OldClass HighRoller)))
              self))
          ( (NEQ 'SellGoods (CADR Send-Super-Args))
            (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
          ( (NEQ 'self (CAR Send-Super-Args))
            (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
          (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'SellGoods (LOADTIMECONSTANT (OldClass
                                                                    HighRoller
                                                                    )))
                                self)
              (CDDR Send-Super-Args]
        [_Super?
          (&REST Send-Super-Args)
          (COND
            ( (NULL Send-Super-Args)
              ' (DOAPPLY* (FindSuperMethod self 'SellGoods (LOADTIMECONSTANT (OldClass HighRoller))
                (FUNCTION NIL))
                self))
            ( (NEQ 'SellGoods (CADR Send-Super-Args))
              (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
            ( (NEQ 'self (CAR Send-Super-Args))
              (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
            (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'SellGoods (LOADTIMECONSTANT (OldClass
                                                                    HighRoller
                                                                    )))
                                self)
                (FUNCTION NIL))
                self)
              (CDDR Send-Super-Args]
        [_SuperFringe
          (&REST Send-Super-Args)
```

## (HighRoller.TakeTurn

[illegible]

```

(CDDR Send-Super-Args]
[_Super?
(&REST Send-Super-Args)
(COND
  ((NULL Send-Super-Args)
   ' (DOAPPLY* (FindSuperMethod self 'TakeTurn (LOADTIMECONSTANT (OldClass HighRoller))
               (FUNCTION NIL))
   self))
  ((NEQ 'TakeTurn (CADR Send-Super-Args))
   (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
  ((NEQ 'self (CAR Send-Super-Args))
   (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
  (T (APPEND ' (DOAPPLY* (FindSuperMethod self 'TakeTurn (LOADTIMECONSTANT (OldClass
                                                                    HighRoller))
                                                                    )
              (FUNCTION NIL))
      self)
      (CDDR Send-Super-Args]
_SuperFringe
(&REST Send-Super-Args)
(COND
  [(NULL Send-Super-Args)
   (LIST 'for 'cls 'in (LIST 'fetch 'localSupers 'of (LIST 'LOADTIMECONSTANT
                                                         (LIST 'OldClass class-name)))
   'do
    (LIST* 'DOAPPLY* [LIST* 'OR (LIST 'FetchMethod 'cls (LIST 'QUOTE selector))
                    ' ((FUNCTION NIL)
                      (NCONC args)
                      ((NEQ 'TakeTurn (CADR Send-Super-Args))
                       (ERROR "Selector to _Super does not match method selector" (CADR Send-Super-Args)))
                      ((NEQ 'self (CAR Send-Super-Args))
                       (ERROR "Can't _Super to other then first arg of method" (CDR Send-Super-Args)))
                      (T (LIST* 'for 'cls 'in ' (fetch localSupers of (LOADTIMECONSTANT (OldClass HighRoller)))
                          'bind
                          [LIST 'argList ' _ (LIST* 'MAPCAR [LIST 'QUOTE (CONS (CAR Send-Super-Args)
                                                                                  (NCONC (CDDR Send-Super-Args)
                                                                                      ]
                                                                                  ' ((FUNCTION EVAL)
                                                                                    ' (do (APPLY (OR (FetchMethod cls 'TakeTurn)
                                                                (FUNCTION NIL))
                                                                argList]
                                                                (PROG (^auditRecord ^value alice)
^firstRules
[COND
  ((AND (NOT (GetValue self 'goal))
        (NOT (GetValue self 'destination)
        (SETQ ^value (PROGN [PROGN
(* Make an audit record for this rule and set its audit values.)
(SETQ ^auditRecord ( _ ($ StandardAuditRecord
New))
(PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.:%F5.o28.D_;
. 603]
(PutAuditRec self 'goal 'Buy ^auditRecord]
[COND
  ((EQ (GetValue self 'goal)
'Buy)
(SETQ ^value (PROGN [PROGN
(* Make an audit record for this rule and set its audit values.)
(SETQ ^auditRecord ( _ ($ StandardAuditRecord
New))
(PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.:%F5.o28.D_;
. 604]
(PutAuditRec self 'destination (NearestRoadStop (Sellers ($ LuxuryGoods
)))
^auditRecord]
[COND
  ((OR (NOT (GetValue self 'destination))
        (ISA (GetValue (GetValue self 'truck)
'location)
($ UnionHall)))
(SETQ ^value (PROGN [PROGN
(* Make an audit record for this rule and set its audit values.)
(SETQ ^auditRecord ( _ ($ StandardAuditRecord
New))
(PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.:%F5.o28.D_;
. 605]
(PutAuditRec self 'destination (NearestRoadStop (Sellers ($ LuxuryGoods
)))
^auditRecord)
(PutAuditRec self 'goal 'Buy ^auditRecord]
[COND
  ((AND (NOT (GetValue self 'destination))
        (NOT (Sellers ($ LuxuryGoods)))
        (EQ (GetValue self 'goal)
'Buy)
(SETQ ^value (PROGN [PROGN
(* Make an audit record for this rule and set its audit values.)
(SETQ ^auditRecord ( _ ($ StandardAuditRecord
New))
(PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.:%F5.o28.D_;
. 606]

```

```

(PutAuditRec self 'goal 'GetClose ^auditRecord)
[SETQ alice (NearestRoadStop (RoadStops ($ AlicesRestaurant]
(PutAuditRec self 'destination (GetValue alice 'prev)
  ^auditRecord]
[COND
  ((AND (LESSP (GetValue Simulator 'timeLeft)
    50)
    (NEQ (GetValue self 'goal)
      'SitTight)
    (NEQ (GetValue self 'goal)
      'GoToAlices))
    (SETQ ^value (PROGN [PROGN
      (* Make an audit record for this rule and set its audit values.)
      (SETQ ^auditRecord (_ ($ StandardAuditRecord
        New))
      (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
        . 607]
      [SETQ alice (NearestRoadStop (RoadStops ($ AlicesRestaurant]
      (PutAuditRec self 'destination (GetValue alice 'prev)
        ^auditRecord)
      (PutAuditRec self 'goal 'GetClose ^auditRecord]
[COND
  ((AND (LESSP (GetValue Simulator 'timeLeft)
    20)
    (NOT (ISA (GetValue (GetValue self 'truck)
      'location)
      ($ AlicesRestaurant]
    (SETQ ^value (PROGN [PROGN
      (* Make an audit record for this rule and set its audit values.)
      (SETQ ^auditRecord (_ ($ StandardAuditRecord
        New))
      (PutValue ^auditRecord 'rule (GetObjFromUID ' (LOX0.%.F5.o28.D_;
        . 608]
      (PutAuditRec self 'destination (NearestRoadStop (RoadStops ($
        AlicesRestaurant
        )))
      ^auditRecord)
      (PutAuditRec self 'goal 'GoToAlices ^auditRecord]
    (SETQ ^value (_ self GoToStoppingPlace))
  QUIT
  ^lastRules
  (RETURN ^value])
)
(\BatchMethodDefs)

```