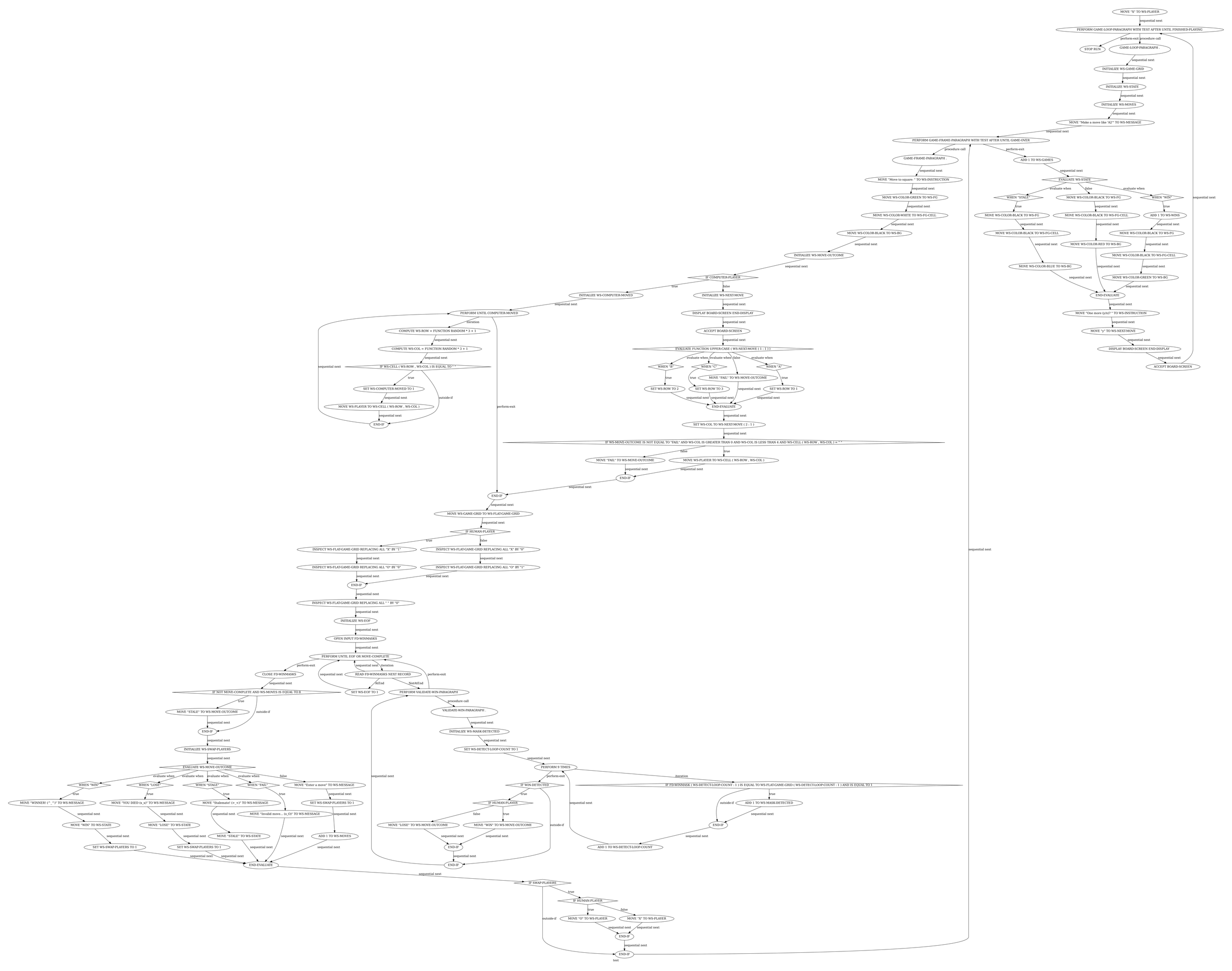
MOVE "X" TO WS-PLAYER
sequential next
PERFORM GAME-LOOP-PARAGRAPH WITH TEST AFTER UNTIL FINISHED-PLAYING
perform-exit
procedure call
STOP RUN
GAME-LOOP-PARAGRAPH .
sequential next
INITIALIZE WS-GAME-GRID
INITIALIZE WS-STATE
INITIALIZE WS-MOVES
MOVE "Make a move like 'A2'" TO WS-MESSAGE
PERFORM GAME-FRAME-PARAGRAPH WITH TEST AFTER UNTIL GAME-OVER
GAME-FRAME-PARAGRAPH .
ADD 1 TO WS-GAMES
MOVE "Move to square: " TO WS-INSTRUCTION
EVALUATE WS-STATE
evaluate when
false
WHEN "STALE"
WHEN "WIN"
MOVE WS-COLOR-GREEN TO WS-FG
MOVE WS-COLOR-BLACK TO WS-FG
MOVE WS-COLOR-WHITE TO WS-FG-CELL
MOVE WS-COLOR-BLACK TO WS-FG-CELL
ADD 1 TO WS-WINS
MOVE WS-COLOR-BLACK TO WS-BG
MOVE WS-COLOR-BLACK TO WS-FG-CELL
MOVE WS-COLOR-BLACK TO WS-FG
INITIALIZE WS-MOVE-OUTCOME
MOVE WS-COLOR-RED TO WS-BG
MOVE WS-COLOR-BLACK TO WS-FG-CELL
IF COMPUTER-PLAYER
MOVE WS-COLOR-BLUE TO WS-BG
MOVE WS-COLOR-GREEN TO WS-BG
true
false
INITIALIZE WS-COMPUTER-MOVED
INITIALIZE WS-NEXT-MOVE
END-EVALUATE
MOVE "One more (y/n)? " TO WS-INSTRUCTION
MOVE "y" TO WS-NEXT-MOVE
DISPLAY BOARD-SCREEN END-DISPLAY
ACCEPT BOARD-SCREEN
PERFORM UNTIL COMPUTER-MOVED
iteration
COMPUTE WS-ROW = FUNCTION RANDOM * 3 + 1
COMPUTE WS-COL = FUNCTION RANDOM * 3 + 1
IF WS-CELL (WS-ROW , WS-COL) IS EQUAL TO " "
SET WS-COMPUTER-MOVED TO 1
MOVE WS-PLAYER TO WS-CELL (WS-ROW , WS-COL)
END-IF
outside-if
DISPLAY BOARD-SCREEN END-DISPLAY
ACCEPT BOARD-SCREEN
EVALUATE FUNCTION UPPER-CASE (WS-NEXT-MOVE (1 : 1))
WHEN "B"
WHEN "C"
WHEN "A"
MOVE "FAIL" TO WS-MOVE-OUTCOME
SET WS-ROW TO 2
SET WS-ROW TO 3
SET WS-ROW TO 1
SET WS-COL TO WS-NEXT-MOVE (2 : 1)
IF WS-MOVE-OUTCOME IS NOT EQUAL TO "FAIL" AND WS-COL IS GREATER THAN 0 AND WS-COL IS LESS THAN 4 AND WS-CELL (WS-ROW , WS-COL) = " "
MOVE "FAIL" TO WS-MOVE-OUTCOME
MOVE WS-PLAYER TO WS-CELL (WS-ROW , WS-COL)
MOVE WS-GAME-GRID TO WS-FLAT-GAME-GRID
IF HUMAN-PLAYER
INSPECT WS-FLAT-GAME-GRID REPLACING ALL "X" BY "1"
INSPECT WS-FLAT-GAME-GRID REPLACING ALL "X" BY "0"
INSPECT WS-FLAT-GAME-GRID REPLACING ALL "O" BY "0"
INSPECT WS-FLAT-GAME-GRID REPLACING ALL "O" BY "1"
INSPECT WS-FLAT-GAME-GRID REPLACING ALL " " BY "0"
INITIALIZE WS-EOF
OPEN INPUT FD-WINMASKS
PERFORM UNTIL EOF OR MOVE-COMPLETE
CLOSE FD-WINMASKS
READ FD-WINMASKS NEXT RECORD
AtEnd
NotAtEnd
SET WS-EOF TO 1
IF NOT MOVE-COMPLETE AND WS-MOVES IS EQUAL TO 8
MOVE "STALE" TO WS-MOVE-OUTCOME
PERFORM VALIDATE-WIN-PARAGRAPH
VALIDATE-WIN-PARAGRAPH .
INITIALIZE WS-MASK-DETECTED
SET WS-DETECT-LOOP-COUNT TO 1
PERFORM 9 TIMES
IF WIN-DETECTED
IF FD-WINMASK (WS-DETECT-LOOP-COUNT : 1) IS EQUAL TO WS-FLAT-GAME-GRID (WS-DETECT-LOOP-COUNT : 1) AND IS EQUAL TO 1
ADD 1 TO WS-MASK-DETECTED
ADD 1 TO WS-DETECT-LOOP-COUNT
IF HUMAN-PLAYER
MOVE "LOSE" TO WS-MOVE-OUTCOME
MOVE "WIN" TO WS-MOVE-OUTCOME
END-IF
END-EVALUATE
WHEN "WIN"
WHEN "LOSE"
WHEN "STALE"
WHEN "FAIL"
MOVE "Enter a move" TO WS-MESSAGE
MOVE "WINNER! (^_^)" TO WS-MESSAGE
MOVE "YOU DIED (x_x)" TO WS-MESSAGE
MOVE "Stalemate! (>_<)" TO WS-MESSAGE
MOVE "Invalid move... (o_O)" TO WS-MESSAGE
SET WS-SWAP-PLAYERS TO 1
MOVE "WIN" TO WS-STATE
MOVE "LOSE" TO WS-STATE
MOVE "STALE" TO WS-STATE
ADD 1 TO WS-MOVES
INITIALIZE WS-SWAP-PLAYERS
EVALUATE WS-MOVE-OUTCOME
IF SWAP-PLAYERS
MOVE "O" TO WS-PLAYER
MOVE "X" TO WS-PLAYER
test
sequential next
perform-exit
iteration
true
false
outside-if