```
#=============================================
#   Purpose:        Evaluates eligible squares for a pawn movement
#   Parameter:       square ; color ;{ type }
#   Called by:      Select Square ( square )
#   Author:         Joe Cellino jrcellin@gmail.com
#   Notes:          none
#=============================================

#Load variables using #( name , value ) custom function. Exit if required variables are null.
If [ not #AssignScriptParameters ]
    Exit Script [  ]
End If

Set Error Capture [ On ]
Allow User Abort [ Off ]

If [ $type = "p" ]
    Perform Script [ “Pawn Protected Squares ( square ; color )”; Parameter: # ( "square" ; $square ) &
        # ( "color" ; $color ) ]
    Exit Script [  ]
End If

Set Variable [ $oppositeColor; Value:Case ( $color = "W" ; "B" ; "W" ) ]

If [ $color = "W" ]
    Set Variable [ $vars; Value:Let ( [

        $letter = SquareInfo ( $square ; "L" ) ;
        $number = SquareInfo ( $square ; "N" )

        ] ; Null ) ]

    # (1) Check Normal movement
    Set Variable [ $square; Value:$letter & $number + 1 ]
    Go to Object [ Object Name: $square ]
    If [ Get ( ActiveFieldContents ) = Null ]
        Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
            # ( "startSquare" ; $$SELECTED.SQUARE ) &
            # ( "piece" ; "pawn" ) &
            # ( "color" ; $color ) &

            # (         "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
            # (         "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
            # (         "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
            # (         "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
            # (         "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
            # (         "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

            # (         "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
            # (         "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
            # (         "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
            # (         "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
            # (         "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
            # (         "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
```

```
        Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
        If [ $notCheck ]
            Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
            # First square needs to be eligible in order to check next square
            Set Variable [ $checkNextSquare; Value:True ]
        Else If [ not $notCheck ]
            Set Variable [ $checkNextSquare; Value:True ]
        End If
    End If

    # Check for starting position
    If [ $number = 2 and $checkNextSquare ]
        Set Variable [ $square; Value:$letter & $number + 2 ]
        Go to Object [ Object Name: $square ]
        If [ Get ( ActiveFieldContents ) = Null ]
            Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
                # ( "startSquare" ; $$SELECTED.SQUARE ) &
                # ( "piece" ; "pawn" ) &
                # ( "color" ; $color ) &

                # (       "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
                # (       "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
                # (       "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
                # (       "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
                # (       "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
                # (       "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

                # (       "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
                # (       "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
                # (       "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
                # (       "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
                # (       "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
                # (       "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
            Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
            If [ $notCheck ]
                Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
            End If
        End If
    End If


    # (2) Check for attacking pieces
    Set Variable [ $square; Value:NextPrevLetter ( $letter ; "next" ) & $number + 1 ]
    Go to Object [ Object Name: $square ]
    If [ Get ( ActiveFieldContents ) ≠ Null and GetPieceColor ( $square ) = $oppositeColor ]
        Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
            # ( "startSquare" ; $$SELECTED.SQUARE ) &
            # ( "piece" ; "pawn" ) &
            # ( "color" ; $color ) &

            # (           "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
            # (           "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
            # (           "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
            # (           "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
            # (           "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
            # (           "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &
```

```
            # (            "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
            # (            "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
            # (            "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
            # (            "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
            # (            "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
            # (            "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
        Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
        If [ $notCheck ]
            Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
        End If
    End If
    Set Variable [ $square; Value:NextPrevLetter ( $letter ; "prev" ) & $number + 1 ]
    Go to Object [ Object Name: $square ]
    If [ Get ( ActiveFieldContents ) ≠ Null and GetPieceColor ( $square ) = $oppositeColor ]
        Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
            # ( "startSquare" ; $$SELECTED.SQUARE ) &
            # ( "piece" ; "pawn" ) &
            # ( "color" ; $color ) &

            # (            "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
            # (            "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
            # (            "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
            # (            "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
            # (            "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
            # (            "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

            # (            "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
            # (            "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
            # (            "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
            # (            "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
            # (            "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
            # (            "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
        Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
        If [ $notCheck ]
            Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
        End If
    End If


    # (3) Check for En Passant
    # Must be on the 5th rank
    If [ $number = "5" ]
        # Opponent moves a pawn
        If [ Game::lastPieceMoved = "Pawn" ]
            Set Variable [ $leftSquare; Value:NextPrevLetter ( $letter ; "prev" ) & $number ]
            Set Variable [ $leftStartSquare; Value:NextPrevLetter ( $letter ; "prev" ) & $number + 2 ]
            If [ Game::lastMoveEndingSquare = $leftSquare

                and Game::lastMoveStartingSquare = $leftStartSquare ]
                Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
                    # ( "startSquare" ; $$SELECTED.SQUARE ) &
                    # ( "piece" ; "pawn" ) &
                    # ( "color" ; $color ) &

                    # (            "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
                    # (            "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
                    # (            "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
                    # (            "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
```

```
                    # (        "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
                    # (        "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

                    # (        "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
                    # (        "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
                    # (        "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
                    # (        "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
                    # (        "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
                    # (        "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
                Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
                If [ $notCheck ]
                    Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; NextPrevLetter ( $letter ;
                        "prev" ) & $number + 1 ) ]
                    Set Variable [ $$ENPASSANT.SQUARES; Value:List ( $$ENPASSANT.SQUARES ; NextPrevLetter ( $letter ;
                        "prev" ) & $number + 1 ) ]
                End If
            End If
            Set Variable [ $rightSquare; Value:NextPrevLetter ( $letter ; "next" ) & $number ]
            Set Variable [ $rightStartSquare; Value:NextPrevLetter ( $letter ; "next" ) & $number + 2 ]
            If [ Game::lastMoveEndingSquare = $rightSquare

                and Game::lastMoveStartingSquare = $rightStartSquare ]
                Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
                    # ( "startSquare" ; $$SELECTED.SQUARE ) &
                    # ( "piece" ; "pawn" ) &
                    # ( "color" ; $color ) &

                    # (        "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
                    # (        "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
                    # (        "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
                    # (        "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
                    # (        "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
                    # (        "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

                    # (        "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
                    # (        "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
                    # (        "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
                    # (        "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
                    # (        "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
                    # (        "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
                Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
                If [ $notCheck ]
                    Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; NextPrevLetter ( $letter ;
                        "next" ) & $number + 1 ) ]
                    Set Variable [ $$ENPASSANT.SQUARES; Value:List ( $$ENPASSANT.SQUARES ; NextPrevLetter ( $letter ;
                        "next" ) & $number + 1 ) ]
                End If
            End If
        End If
    End If

    # ################### BLACK
Else
    Set Variable [ $vars; Value:Let ( [

        $letter = SquareInfo ( $square ; "L" ) ;
        $number = SquareInfo ( $square ; "N" )

        ] ; Null ) ]
```

```
# (1) Check Normal movement
Set Variable [ $square; Value:$letter & $number - 1 ]
Go to Object [ Object Name: $square ]
If [ Get ( ActiveFieldContents ) = Null ]
    Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
        # ( "startSquare" ; $$SELECTED.SQUARE ) &
        # ( "piece" ; "pawn" ) &
        # ( "color" ; $color ) &

        # (         "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
        # (         "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
        # (         "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
        # (         "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
        # (         "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
        # (         "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

        # (         "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
        # (         "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
        # (         "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
        # (         "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
        # (         "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
        # (         "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
    Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
    If [ $notCheck ]
        Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
        # First square needs to be eligible in order to check next square
        Set Variable [ $checkNextSquare; Value:True ]
        # Check if moving pawn 2 spaces can get out of check
    Else If [ not $notCheck ]
        Set Variable [ $checkNextSquare; Value:True ]
    End If
End If

# Check for starting position
If [ $number = 7 and $checkNextSquare ]
    Set Variable [ $square; Value:$letter & $number - 2 ]
    Go to Object [ Object Name: $square ]
    If [ Get ( ActiveFieldContents ) = Null ]
        Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
            # ( "startSquare" ; $$SELECTED.SQUARE ) &
            # ( "piece" ; "pawn" ) &
            # ( "color" ; $color ) &

            # (       "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
            # (       "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
            # (       "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
            # (       "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
            # (       "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
            # (       "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

            # (       "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
            # (       "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
            # (       "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
            # (       "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
            # (       "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
            # (       "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
        Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
```

```
            If [ $notCheck ]
                Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
            End If
        End If
    End If


    # (2) Check for attacking pieces
    Set Variable [ $square; Value:NextPrevLetter ( $letter ; "next" ) & $number - 1 ]
    Go to Object [ Object Name: $square ]
    If [ Get ( ActiveFieldContents ) ≠ Null and GetPieceColor ( $square ) = $oppositeColor ]
        Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
            # ( "startSquare" ; $$SELECTED.SQUARE ) &
            # ( "piece" ; "pawn" ) &
            # ( "color" ; $color ) &

            # (         "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
            # (         "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
            # (         "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
            # (         "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
            # (         "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
            # (         "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

            # (         "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
            # (         "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
            # (         "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
            # (         "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
            # (         "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
            # (         "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
        Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
        If [ $notCheck ]
            Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
        End If
    End If
    Set Variable [ $square; Value:NextPrevLetter ( $letter ; "prev" ) & $number - 1 ]
    Go to Object [ Object Name: $square ]
    If [ Get ( ActiveFieldContents ) ≠ Null and GetPieceColor ( $square ) = $oppositeColor ]
        Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
            # ( "startSquare" ; $$SELECTED.SQUARE ) &
            # ( "piece" ; "pawn" ) &
            # ( "color" ; $color ) &

            # (         "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
            # (         "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
            # (         "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
            # (         "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
            # (         "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
            # (         "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

            # (         "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
            # (         "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
            # (         "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
            # (         "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
            # (         "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
            # (         "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
        Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
        If [ $notCheck ]
            Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; $square ) ]
```

```
        End If
    End If

    # (3) Check for En Passant
    # Must be on the 4th rank
    If [ $number = "4" ]
        # Opponent moves a pawn
        If [ Game::lastPieceMoved = "Pawn" ]
            Set Variable [ $leftSquare; Value:NextPrevLetter ( $letter ; "prev" ) & $number ]
            Set Variable [ $leftStartSquare; Value:NextPrevLetter ( $letter ; "prev" ) & $number - 2 ]
            If [ Game::lastMoveEndingSquare = $leftSquare

                and Game::lastMoveStartingSquare = $leftStartSquare ]
                Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
                    # ( "startSquare" ; $$SELECTED.SQUARE ) &
                    # ( "piece" ; "pawn" ) &
                    # ( "color" ; $color ) &

                    # (        "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
                    # (        "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
                    # (        "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
                    # (        "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
                    # (        "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
                    # (        "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

                    # (        "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
                    # (        "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
                    # (        "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
                    # (        "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
                    # (        "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
                    # (        "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
                Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
                If [ $notCheck ]
                    Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; NextPrevLetter ( $letter ;
                        "prev" ) & $number - 1 ) ]
                    Set Variable [ $$ENPASSANT.SQUARES; Value:List ( $$ENPASSANT.SQUARES ; NextPrevLetter ( $letter ;
                        "prev" ) & $number - 1 ) ]
                End If
            End If
            Set Variable [ $rightSquare; Value:NextPrevLetter ( $letter ; "next" ) & $number ]
            Set Variable [ $rightStartSquare; Value:NextPrevLetter ( $letter ; "next" ) & $number - 2 ]
            If [ Game::lastMoveEndingSquare = $rightSquare

                and Game::lastMoveStartingSquare = $rightStartSquare ]
                Perform Script [ “Check For Check { Position }”; Parameter: # ( "endSquare" ; $square ) &
                    # ( "startSquare" ; $$SELECTED.SQUARE ) &
                    # ( "piece" ; "pawn" ) &
                    # ( "color" ; $color ) &

                    # (        "WHITE.PAWN.PLACEMENT" ; Game::placementWhitePawns ) &
                    # (        "WHITE.KNIGHT.PLACEMENT" ; Game::placementWhiteKnights ) &
                    # (        "WHITE.BISHOP.PLACEMENT" ; Game::placementWhiteBishops ) &
                    # (        "WHITE.ROOK.PLACEMENT" ; Game::placementWhiteRooks ) &
                    # (        "WHITE.QUEEN.PLACEMENT" ; Game::placementWhiteQueen ) &
                    # (        "WHITE.KING.PLACEMENT" ; Game::placementWhiteKing ) &

                    # (        "BLACK.PAWN.PLACEMENT" ; Game::placementBlackPawns ) &
                    # (        "BLACK.KNIGHT.PLACEMENT" ; Game::placementBlackKnights ) &
                    # (        "BLACK.BISHOP.PLACEMENT" ; Game::placementBlackBishops ) &
```

```
                    # (        "BLACK.ROOK.PLACEMENT" ; Game::placementBlackRooks ) &
                    # (        "BLACK.QUEEN.PLACEMENT" ; Game::placementBlackQueen ) &
                    # (        "BLACK.KING.PLACEMENT" ; Game::placementBlackKing ) ]
                Set Variable [ $notCheck; Value:Get ( ScriptResult ) ]
                If [ $notCheck ]
                    Set Variable [ $$ELIGIBLE.SQUARES; Value:List ( $$ELIGIBLE.SQUARES ; NextPrevLetter ( $letter ;
                        "next" ) & $number - 1 ) ]
                    Set Variable [ $$ENPASSANT.SQUARES; Value:List ( $$ENPASSANT.SQUARES ; NextPrevLetter ( $letter ;
                        "next" ) & $number - 1 ) ]
                End If
            End If
        End If
    End If

End If
```