```
--This code enables the puzzle pieces to snap to specif coordinates when released.

constant kHDist = 118 // horizontal distance between points
constant kVDist = 192// vertical distance between points

on SnapToGrid MyButtonName
  put the top of target into tTop
  put the left of target into tLeft
  locToSnap tTop, tLeft
  lock screen
  set the top of target to tTop
  set the left of target to tLeft
  unlock screen
  --put the topLeft of target
end SnapToGrid

on locToSnap @xTop, @xLeft
  put round(xTop / kVDist) * kVDist into xTop
  put round(xLeft /kHDist) * kHDist into xLeft
end locToSnap


--As all the puzzle pieces are rectangles, this code gets the topLeft of each rectangle as i
there was a grid of rectangles over the complete image. The topLeft coordinates are
stored in a container for use in the second process.

on GetTopLefts
   put "" into cd fld "PuzzlePositions"
   put 0 into x
   put 0 into y
   --1
   repeat 4
      put x & "," & y & return after cd fld "puzzlePositions"
      --put x & "," & (y + cd fld Yloc) & return after cd fld "puzzlePositions"
      put (y + cd fld "YLoc") into y
   end repeat
   --2
   put 0 into y
   put (x + cd fld "XLoc") into x
  repeat 4
       put x & "," & y & return after cd fld "puzzlePositions"
   put (y + cd fld "YLoc") into y
   end repeat
   --3
    put 0 into y
   put (x + cd fld "XLoc") into x
  repeat 4
       put x & "," & y & return after cd fld "puzzlePositions"
   put (y + cd fld "YLoc") into y
   end repeat
   --4
     put 0 into y
   put (x + cd fld "XLoc") into x
  repeat 4
       put x & "," & y & return after cd fld "puzzlePositions"
   put (y + cd fld "YLoc") into y
end repeat
--5
  put 0 into y
   put (x + cd fld "XLoc") into x
  repeat 4
       put x & "," & y & return after cd fld "puzzlePositions"
   put (y + cd fld "YLoc") into y
end repeat
  --6
  put 0 into y
   put (x + cd fld "XLoc") into x
  repeat 4
       put x & "," & y & return after cd fld "puzzlePositions"
       put (y + cd fld "YLoc") into y
       end repeat

end GetTopLefts

--This code creates the very first puzzle piece. It moves the puzzle piece behind the scenes,
--based on the list of topLeft coordinates determined in the first bit of code.
--Then the bottomRight coordinate is collected and placed after the topleft coordinate for each topLeft coordinate.
--The topLeft coordinate and the bottomRight coordinate are used to determine the appropriate rectangle for third step in process.
on GetRects
   put 1 into x
   put 1 into PuzzleIndex
   put 0 into theBottom
   put cd fld 1 into tWidth
     put line x of cd fld "puzzlepositions" into tLeft
     put (tWidth + theBottom) into theBottom
     --put (x * cd fld 2) & "," & theBottom into tBottomChords
     put (cd fld 2) & "," & theBottom into tBottomChords
     put tLeft & "," & tBottomChords into tRect
     --Answer tRect
     put "export snapshot from rect " & tRect & " of cd 1 of current stack to tVar as png"
     export snapshot from rect tRect of cd 1 of current stack to tVar as png

     new image
     put tVar into last image
     put line x of cd fld "PuzzlePositions" into tLoc
     set the topleft of  the last image to tLoc
     set the name of last image to "puzzlePiece" & PuzzleIndex
       put 2 into x
     repeat for number of lines in cd fld "PuzzlePositions" - 1
       put line x of cd fld "PuzzlePositions" into tTopLeft
       set the topLeft of last image to tTopLeft
       put "," & bottomRight of last image after line x of cd fld "PuzzlePositions"
       add 1 to x
     end repeat
     set the topLeft of last img to 0,0
     set the visible of last img to true
     delete line 1 of cd fld "PuzzlePositions"
end GetRects

--The third step in the process is to create the actual puzzle pieces. Using the coordinates from second step,
--the a screen capture of each specific rectangle is created from the intial image.
--Each puzzle piece (image) has code added to it to move it, bring it to the front, and change a
--couple of graphic settings when touched/clicked on and reset all the other pieces.
--The puzzle pieces are layed out in their correct positions, unscrambled.
on CreatePieces
   put 1 into x
   put 2 into tPuzzleIndex
   set the script of  img "PuzzlePiece1" to "on MouseDown" & return & "EliminateInnerGlow" & return & "Grab me" & return & "set the layer of me to top" & return & "set the outerGlow[color] of me to 0,0,0" & return & "set the innerGlow[color] of me to 0,0,0" & return & "end MouseDown" & return & "on MouseUp" & return & "SnapToGrid" & return & "end mouseUp"
   repeat for number of lines in cd fld "PuzzlePositions"
     put line x of cd fld "PuzzlePositions" into tRect
```

```
     export snapshot from rect tRect of cd 1 of current stack to tVar as png

    new image
    put tVar into last image
    set the name of last image to "puzzlePiece" & tPuzzleIndex
    set the visible of last image to true
    put item 1 of line x of cd fld "puzzlePositions" & "," & item 2 of line x of cd fld
"puzzlePositions" into tCorrectLoc
    set the topleft of last img to tCorrectLoc --100,-100


    set the script of last img to "on MouseDown" & return & "EliminateInnerGlow" & return &
"Grab me" & return & "set the layer of me to top" & return & "set the outerGlow[color] of me to
0,0,0" & return & "set the innerGlow[color] of me to 0,0,0" & return  & "end MouseDown" &
return & "on MouseUp" & return & "SnapToGrid" & return & "end mouseUp"
    add 1 to tPuzzleIndex
    add 1 to x

  end repeat
end CreatePieces

--Using the list of rectangle coordinates created earlier, the applications randomizes the
list of coordinates
--and moves each puzzle piece to a random "topLeft" location on the screen.

on StartPuzzle
  put cd fld "PuzzlePositions" into tOriginalPuzzlePositions
  put 1 into x
  repeat for number of lines in cd fld "PuzzlePositions"
    put random(number of lines in cd fld "PuzzlePositions") into tTargetPiece
    put (item 1 of line tTargetPiece of cd fld id 1032) & "," & (item 2 of line tTargetPiece of cd fld
id 1032) into tTargetLoc
    set the topleft of img ("PuzzlePiece" & x) to tTargetLoc --of cd fld "PuzzlePositions"
    delete line tTargetPiece of cd fld "PuzzlePositions"
    add 1 to x
  end repeat
  put tOriginalPuzzlePositions into cd fld "puzzlePositions"
end StartPuzzle
```