AcroTeX.Net
The AeB Pro family

The AcroMemory Package
Manual of Usage

D. P. Story

Last Revision Date: November 26, 2006

Table of Contents

1. What is the AeB Pro Family?

Through the years, I have tried to make my AeB software (AcroTEX eDucation Bundle) compatible with pdftex and dvipdfm; however, during that time, I've developed a number of techniques that require the use of Acrobat and distiller. Therefore, I have set off in a new direction and will be publishing a new line of LATEX packages, ones that require the use of Acrobat.[1]

The current package, AcroMemory, requires the use of Acrobat Pro 7.0 or later.

2. Introduction

At the prompting and encouragement of my erstwhile friend, Jürgen, I present to you AcroMemory, and for the life of me, I can't remember why.

Oh, yes, AcroMemory is a memory game in which you find the matching tiles. There are two versions—available as options of this package—for your enjoyment, `acromemory1` and `acromemory2` (the default).

- `acromemory1`: Here you have a single game board, a rectangular region divided by rows and columns. The total number of tiles should be even, each tile should have a matching twin. The game begins with all the tiles hidden. The user clicks a tile, then another. If the tiles do not match, they become hidden again (you did remember the position of those tiles, didn't you?); otherwise, they remain visible and are now read-only. The game is complete when the user, with a lot of time on his/her hands, matches all tiles. There is a running tabulation kept on the number of tries. There is also a button which resets the game and randomizes the tiles.
- `acromemory2`: For this game you have two identical rectangular images subdivided into tiles (or slices), which are arrayed in rows and columns. The tiles for one of the two images is randomly re-arranged. The object of the game is to find all the matching tiles by choosing a tile from one image and a tile from the other image. As in the first case, if the selected tiles do not match, they are hidden after a short interval of time (you did remember the position of those tiles, didn't you?); otherwise, they remain visible and are now read-only. The game is over when all tiles are matched; when this occurs, end-of-game special effects occur that will dazzle the senses. There is an option to view a small image to help you locate the matching tiles on the non-randomized; useful if the image is complex. There is no reset button at this time, to play again, the user must therefore close and open the document.

The demo files are `acromemory1_1.tex`, `acromemory1_2.tex`, `acromemory2_1.tex` and `acromemory2_2.tex`. These files show how to lay out the various elements of this package.

Requirements: Acrobat 7 Professional.

3. Distribution and Installation

The distribution comes with the following files:

[1] I will, however, continue to develop the original AeB, never fear!

- `acromemory.dtx`, `acromemory.sty` and `acromemory.ins`: These are the program files. The file `acromemory.dtx` has additional documentation with more technical details than this manual.
- The files `acromemory1_1.tex`, `acromemory1_2.tex`, `acromemory2_1.tex` and `acromemory2_2.tex`: The demo files for this package.
- `aeb_pro.js`: A JavaScript file that contains security restricted methods.
- The AeB Slicing Batch Sequence: The documentation, `aebslicing_sequ_doc.pdf`, for this batch sequence comes with the distribution. Installation details for AeB Slicing are contained in its documentation.
- Sample image files, `dinos` (for the `acromemory1` option) and `dpsweb` (for `acromemory2`), contained in their own folders.

When you unzip `acromemory.zip`, the folder `acromemory` will be created, the entire distribution will be placed in this folder. This folder must be on your LaTeX search path and you need to refresh your file name database if you are using a TeX system, such as MIKTeX, that has this database scheme. At the top level will be the LaTeX package and demo files. The subfolders are

- `aeb_pro`: In this folder is `aeb_pro.js`, a JavaScript file that needs to be installed on your hard drive. Copy this file to the User's `JavaScripts` folder. To find this folder, execute the script

  ```
  app.getPath("user", "javascript");
  ```

  in the JavaScript Debugger Console window.[2] The return value of this is the path to the `JavaScripts` folder; for example, on my system it returns

  ```
  /C/Documents and Settings/dps/Application Data/Adobe/Acrobat/8.0/JavaScripts
  ```

  After placing `aeb_pro.js`, restart Acrobat so it will read this file.
- `dpsweb`: Image files used in `acromemory2_1.tex` and `acromemory2_2.tex`.
- `dinos`: Image files used in `acromemory1_1.tex` and `acromemory1_2.tex`.

Follow the installation instructions for the AcroSlicing batch sequence.

4. Package Options

There are a few options of this package:

- `acromemory1` and `acromemory2`: As described earlier. The `acromemory2` option is the default, so it need never be used, actually.
- `iconfile`: There are two methods of delivering the slices of the game board(s) to this package:
  1. The default method is to have one file for each sliced image. There is a numbering system for the slices, the same system used by Aeb Slicing batch sequence, is to number them with a two digit number across rows: `<basename>_01`, `<basename>_02`, ...`<basename>_10`, `<basename>_11`.... The demo files `acromemory1_2.tex` and `acromemory2_2.tex` illustrate this method.

[2] Place the cursor on the line containing this script and press the `Ctrl+Enter` key.

2. All slices are placed in a single PDF in the same order just described, that is listed by rows. There is an option in the AeB Slicing batch sequence for "packaging" the slices in this way. The demo files `acromemory1_1.tex` and `acromemory2_1.tex` illustrate this method.

The AcroMemory package expects, by default, the first method described. By specifying the `iconfile` option, AcroMemory will get the slices for the single PDF.

- `includehelp`: When building a file with the acromemory2 option, you can also include a help image, a small picture of the game board to help the user to match the randomized slices with the ones on the non-randomized game board. Useful if the image is very complex. The demo files `acromemory2_1.tex` and `acromemory2_2.tex` both contain the necessary code for producing the help feature, the commands only create the help feature if the `includehelp` option is taken.

Important. You need to set the preferences of Acrobat as follows:

1. Start Acrobat, and select `Edit > Preferences`
2. Choose Trust Manager from the Categories panel on the left
3. Check the 'Allow external content' check box

The above settings allow the post-distillation assembly to take place.

5. Commands of the Package

This section describes the various commands available to you through this package. This section can be skipped on first reading, most of the commands are described less formally in subsequent sections.

The following commands are suitable for placement in the preamble of your document.

- `\theTotalTiles`: The total number of tiles in the game board. This parameter is required. For example, `\theTotalTiles{20}`.
- `\theNumRows`: The number of rows in the game board. This parameter is required. For example, `\theNumRows{5}`.
- `\theNumCols`: The number of columns in the game board. This parameter is required. For example, `\theNumCols{4}`.
- `\theImportPath`: The import path to the basename of the image. The path should use the path specification as defined in the *PDF Reference*, and the file name should have no extension. For example, `\theImportPath{myFig/myimages}`. Required.

  There is an optional argument that is typically used when the `iconfile` is in effect with the `acromemory2` option, and an image of the game board is different from the path given by the optional argument; for example,

  ```
  \theImportPath[dpsweb/dpsweb]{dpsweb/dpsweb_package}
  ```

  The required argument points the packaged icons, the optional argument points to a file showing the entire image. See `acromemory2_1.tex` for an example of this situation.
- `\theIconExt`: The extension of the image file(s). Required if different from `pdf`. (Acrobat can make the conversion to PDF.)

- \theTeXImageWidth: The scaled width of the rectangular game board. The game board will be rescaled so that its width is equal to the value specified by the argument of this command, e.g., \theTeXImageWidth{2in}.
- \provideDimensions: If the dimension of the game board is known, the width and height can be entered with this command using the two parameters. For example, \provideDimensions{2in}{2.5in} (width, height).
- \bDebug: A debugging command. When executed in the preamble, more is written to the Acrobat console as the document is opened the first time, also, the icons are initially visible so you can see the layout, and quickly play the game. This was used in development extensively to help develop the JavaScript.

The rest of the commands in this section are properly placed in the body of the document. They are the elements of the game board(s) and supporting elements.

Game Board(s)

- \ulCornerHere: Used with the acromemory1 option, this command sets the upper left corner of the game board. It is followed by either \reserveSpaceByDimension or \reserveSpaceByFile.
- \LulCornerHere, \RulCornerHere: Used with the acromemory2 option, these commands set the upper left corner of the two game boards, on the left and the other on the right. Each of these two commands is immediately followed by one of the two commands \reserveSpaceByDimension or \reserveSpaceByFile.
- \reserveSpaceByDimension: If the size of the image is known, you can reserve space for it by using this command. It has two arguments: the first argument is the width of the image, the second is the height. This command is useful for acromemory1 where the game board is made up of some many rows and columns of tiles whose dimensions you know.
- \reserveSpaceByFile: This command does an \includegraphics in draft mode to get the dimensions of the game board. The required space is made for the rescaled image. The optional argument can be used to insert a file that has the same aspect ratio as the puzzle, the default is the one specified by the optional argument of \theImportPath, which, if not specified, is the same as the required argument of \theImportPath

Supporting fields

- \messageBox: A message text field. As the user works the puzzle, the progress is reported to this field.

The following button is only appropriate when the acromemory1 option is taken.

- \playItAgain: For the acromemory1 option, this button can be placed to reset the game board, the icons are rearranged and hidden again.

When acromemory2 and the includehelp options are taken, these commands are available. See the acromemory.dtx documentation for indications of modifying these commands.

- \helpImage: The button that will contain an icon of the puzzle. There is one optional argument used to modify the appearance of the button.

- \setHelpImageWidth: Used to set the width of the \helpImage button. The default width is 1 inch. For example, \setHelpImageWidth{1.5in}.
- \theHelpCaption: Use this command to set the caption of the help image. For example, \theHelpCaption{I need help}.

  Captions with accents and such, you need to use Unicode escape sequences, \uXXXX, where XXXX are four hexadecimal digits. For example, the caption specified by

  ```
  \theHelpCaption{J\string\\u00FCrgen needs help}
  ```

  will appear in the PDF document as 'Jürgen needs help'. Notice the \string\\ sequence that is needed from the LaTeX source.
- \rolloverHelpButton: The image is normally hidden until the user rolls over the \rolloverHelpButton. The icon appears with a caption under it, the content of the caption can be entered using \theHelpCaption. This command has three parameters, one of which is optional. The first (optional) parameter is used to change the appearance of the button; the next two required parameters are the width and height of the button.

6. The acromemory1 Option

In this game, there is only one game board with an even number of tiles. Each tile has an identical twin, and all the tiles are randomly rearranged. The object of the game is to find all the matching pairs. See the demo files acromemory1_1.tex and acromemory1_2.tex for examples of this game; the former file uses the iconfile option, whereas the latter does not. Both illustrate the \reserveSpaceByDimension and \provideDimensions. See the comments in these files for more details.

The tiles for this game were created by a font that I have called 'Mini Pics Lil Dinos'. The source file for the creation of the myDinos.pdf, the icon file, is myDinos.tex. In this tex file, the web package is used to create pages 2 inches by 2 inches. The multido package from PSTricks was used to produce the pages. See myDinos.tex for additional comments.

To create your own icon file, you will need either a font set with nice images on it, or you can be creative, perhaps using PSTricks to create a series of figures.

Given that you have created your icons in a single file, should you wish to save each page to a separate file, you can execute the following JavaScript in the console, while the icons file is open in Acrobat.

```
var thisPath = /.*\//i.exec(this.path)[0];
var filename = this.documentFileName.replace(/\.pdf$/i,"");
for (var i = 0; i < this.numPages; i++) {
    var j = i+1;
    var index = (j < 10 ) ? ("0"+j) : (""+j);
    this.extractPages({
        nStart: i,
        cPath: thisPath+filename+"_" + index +".pdf"
    });
}
```

The code is included in the file myDinos.tex.

7. The acromemory2 Option

This option was the original concept, it was only after I completed the two game board version did I decide to do the classic one game board version. This is why acromemory2 is the default. The acromemory2 version was much more interesting and challenging to create.

For this option you don't have to have a fancy font, any (interesting) picture will do. The first step is to decide how many rows and columns you want and then slice the image appropriately. This is why I wrote the AeB Slicing batch sequence. Read the documentation for AeB Slicing and slice your picture.

Not only do you need slices of your picture, you also need an EPS of your picture. This is used by latex to leave room for the game board. Acrobat can make the conversion for you, as follows:

To convert your image to EPS

1. Bring your image into Acrobat
2. Click the File > Save As menu item
3. In the Save As dialog box, choose Encapsulated PostScript (*.eps) from the Save as type list
4. Navigate to the folder containing your tiles, and click the Save button

See the demo files acromemory2_1.tex and acromemory2_2.tex for examples of this game; the former file uses the iconfile option, whereas the latter does not. Both illustrate the \reserveSpaceByFile. See the comments in these files for more details.

Use either of these two demo files as a template to create your own memory game. Don't forget that the web package has options to apply a background color or a graphic—this will jazz up your memory game.

8. Assembly

Try compiling one of the demo files. You must have the AeB already installed on your system, of course. Be sure to specify your dvi to postscript converter, for most of you that will be dvips. For example, the preamble of the demo docs, and later your own documents should appear as follows:

```
\documentclass{article}
\usepackage[designv,nodirectory,dvips]{web} % <-- specify dvips or dvipsone
\usepackage[execJS]{eforms}
\usepackage{graphicx}
\usepackage{acromemory}
```

LaTeX your source file, then invoke your dvi-to-ps converter to obtain a postscript file. Now open Acrobat Distiller and distill the new postscript document. If all goes well, Acrobat will start, if not already, and the newly created PDF document will open in it. Notice that the hour glass cursor appears; this means that the post-assembly process is ongoing: You note the one or two little dots on the page (where the game boards should be), these are the fields creates by the commands \ulCornerHere, \LulCornerHere or \RulCornerHere. They are soon replaced by the game board(s). Amazing, simply amazing!

After the hour glass cursor changes, perhaps to the hand tool, be sure to save your document, *this is important*. By saving the document, the game board does not have to be rebuilt every time

the file is opened. Once assembled and saved, the game can be played on Adobe Reader 7.0 or later.

Tip. Use the PDF Optimizer, under the Advance menu, to reduce the size of the file. For example, the file size of `acromemory2_2.pdf` after distillation was 462 KB, but after running the PDF Optimizer on it, the file size was reduced to 85 KB!

9. Final comments

Use any of the demo files as a template to create your own memory game. Don't forget (use your memory?) that the web package has options to apply a background color or a graphic—this will jazz up your memory game.

I do hope you find this game package fun, and that you will be creative in its use. Perhaps you can apply the techniques of this package to create your own game package, there are many possibilities.

Now, I simply must get back to my retirement! DS