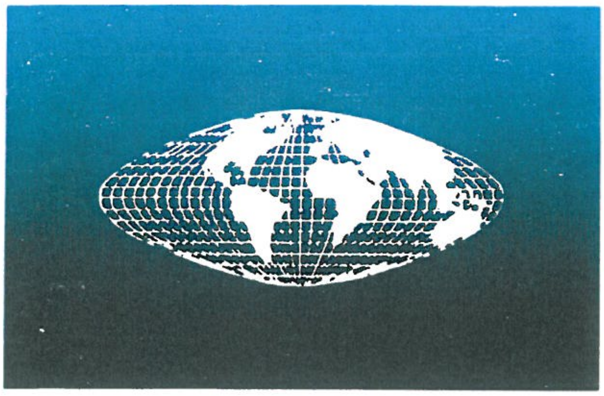

THE JOURNAL OF EDUCATIONAL TECHNIQUES AND TECHNOLOGIES

A Publication of the
International Association
For Learning Laboratories
IALL

VOLUME 21 NUMBER 2/3 SUMMER-FALL 1988

PRACTICES AND PRODUCTS FOR TODAY'S LANGUAGE LEARNING

Text Tanglers: A Vance Stevens RDA/Mind Builders Software

Vital Statistics

Courseware Name: Text Tanglers
Application: English as a Second Language
Instructional Method: Decoding
Vendor: RDA/Mind Builders
Cost: $39.95
Copy Policy: No copying allowed
Equipment: IBM or compatibles
Prerequisites: 2 disk drive or 1 drive & hard disk

General Description

Basically, *Text Tanglers* (TT) offers a set of activity driver programs into which English as a Second Language (ESL) teachers can insert their own texts for students to work with. The activities which TT enables are **Hangman, Cryptogram, Jumbled Sentences, Jumbled Paragraphs,** and **Cloze;** students are also allowed simply to read through any of the texts included in the text files. Twelve texts come with the program, and the teacher can easily create and add others as desired; any file so added can then be the basis for any of the activities.

The programming is, for the most part, competent with the notable exception of scoring the game-like activities. Although the student is provided with a lot of information about the activity and about his or her progress through it, the pedagogical approach is the overall problem with this programming. If the activities presented in *Text Tanglers* appeal to a teacher as worthwhile learning tasks, this package may be quite attractive. This reviewer suspects, however, that many teachers will not want their students to spend much time doing this kind of task.

The Activities

In **Hangman,** one word at a time from the text selected appears as the number of question marks representing the correct number of letters. The alphabet appears across the middle of the screen; when the student guesses a letter which is, in fact, represented in the word one or more times, it replaces the appropriate question marks. An incorrect answer brings forth a buzz and a piece of the scaffold. Letters guessed correctly appear below their place in the alphabet in green; incorrect guesses appear in red. The student thus can keep track of what has already been guessed. The student can ask to see the correct answer, after which the program advances to the next word, or the student can ask for a hint, after which **Hangman** suggests one of the correct letters. Status information includes the number of words available in the text, how many have already been attempted, and scoring information (to be discussed later).

In **Cryptogram,** one sentence from the selected text appears in an alphabetic code in which each letter has systematically been "encrypted" as another. Below the sentence an equal number of dots appears. Below that, the alphabet is lined up across the screen in a fashion similar to that found in **Hangman.** The student puts the cursor on any letter in the code sentence and makes a guess as to what it really is; if correct, the letter automatically appears in *all* its locations in the sentence, and the real letter appears beneath its counterpart in the alphabet line. If incorrect, the buzz sounds. The student can ask to see the solution (in which case the program advances to the next sentence), or the student can ask for a hint about the letter in the code under which the cursor is located. Hints consist of the statement "X has been encrypted as Y," where X is the "real" letter and Y is the code equivalent. The reviewer found this confusing, since Y is what one is asking about. "A Y in the code stands for X" would be a more helpful, less confusing hint. If the user inadvertently asks for a hint on a letter which has already been provided, the program gives the hint which refers to the next unguessed letter in the sentence. Again, this can be quite confusing.

In **Jumbled Sentences,** the sentences from the selected text appear on the screen one at a time in the order in which they are found in the text. On the first screen, the sentence appears correctly with the directions "Please study this sentence." With the next key press, it appears in scrambled word order. Capital letters and punctuation remain attached to the words as they appear in the original. The student uses the arrow keys to move highlighting from one word to another, pressing ENTER to select a word. If the word selected is the next one in the original sentence, it disappears from the jumble and reappears below in order. If not, the buzz sounds but nothing happens. The Hint suggests the next word, and the solution is available.

In **Jumbled Paragraph,** the paragraphs appear in the order in which they are found in the text, one paragraph at a time. The individual sentences are presented as a list. The student uses the up and down arrow keys to select a sentence which is then highlighted, presses ENTER, uses the up and down arrow keys again to select a new location for it, and presses ENTER again to transpose to the new order. When the student thinks the sentences are all in order, he or she presses F2. Up to the point at which F2 is pressed, the program accepts all the student's moves as if they were correct; sentences can be rearranged for as long as the student wants to rearrange them. If F2 is pressed, however, and the order is incorrect, the feedback message is "You could be correct, but this is not the original sentence order." This message acknowledges that more than one order may sometimes be equally plausible. If the choice is correct, both sound and flashing messages "reinforce" it. If a Hint is requested, the program automatically highlights the first sentence in the list which is not in correct position and suggest *that* sentence be moved down lower, but without indicating how much lower. This may cause confusion in the minds of some students because it may cause students to keep switching the order of the first two sentences and continually get the Hint to move each one lower down. The solution shows the paragraph in its original form. At this point, the student may choose between going back and doing the same paragraph again or the next one.

In **Cloze,** sentences from the selected text appear in order on the screen, one at a time, with every fifth word replaced by a blank with the number of dots corresponding to the number of letters in the missing word. The cursor moves automatically to the first dot of the blank, and when that word has been correctly entered, it jumps to the next blank. If the word typed in is not correct, the "wrong" buzz sounds when ENTER is pressed, and all letters disappear; the cursor goes back to the first dot. Blanks can be filled in out of order, but if the student forgets to press ENTER after typing in a word and manually moves the cursor to the next blank, the just-typed letters disappear. In this activity, the Hint gives the letter directly over the cursor; the cursor then moves to the next dot. Again, the user can ask to see the solution or the whole sentence.

General Design Features

Text Tanglers is a very colorful package. This reviewer found it to be almost too colorful even though the colors are used consistently throughout. Audience response to the colorfulness of *Text Tanglers* will depend on age and level of sophistication. The sound effects are generally unharmonious, strident, and unpleasant. Once an activity commences, the sound can be turned off, but rebooting returns to the default setting (sound on) and the cacophony which accompanies the introductory logo cannot be avoided.

The menu offers the user choices: first, the activity; then, the text on which the activity will be based. The user is always provided with a great deal of information: the activity and text chosen, the number of items in an activity, and the number of correct and incorrect responses (see discussion of scoring below). A good range of options is provided: students can call for a help screen which spells out the directions for each activity, for a Hint, and for the correct solution, can turn off the sound, or can quit the activity. The latter returns the program to the main menu, and here, unfortunately, there is no way to quit to DOS, only the option to start over.

Programming

The package is competently programmed. The keys designated for offered choices seem to be always active; the unneeded keys are inoperative. This reviewer found no bugs in the program, but there are a few inconsistencies: the machine sound responding to a correct response is not always the same, and the message "Good-bye" appears sometimes when an activity is abandoned.

Scoring

The worst feature of *Text Tanglers* is the scoring and should, in this reviewer's opinion, be eliminated. Teachers should certainly not base any evaluation of a student's success or perseverance based on the reported score; they should tell their students to ignore the scoring. For example, in several of the activities, repeating a correctly guessed letter will increment the score infinitely; repeating an incorrect answer will increment the negative score. Except for **Hangman,** the activities are not particularly game-like, but rather exploratory. Even if the scoring mechanism in *Text Tanglers* were better, it would not add anything to the learning experience.

Inserting One's Own Texts

Text Tanglers allows teachers to create their own text files for students to use in any activity. Texts may be typed out using any word-processor that can create pure ASCII files, i.e., files without any formatting characters. Not a programmer, this reviewer found text insertion relatively easy to do and add to the available list. The documentation is adequate but generally not well-written. Teachers without any computer experience, however, will probably need help until they are used to the routines.

Use for Foreign Languages

Unfortunately, this package is not designed for use with any language but English. Although the program does allow the use of foreign language texts in the jumbled sentences and paragraph activities, the problem is that the programs neither allow for typing in foreign words nor for accented characters. In principle, the package could be easily adapted for use with foreign languages whose diacritical marks can be created with regular ASCII code.

Pedagogical Value

Whether or not teachers use *Text Tanglers* will probably depend on their assessment of the pedagogical value of its activities. This reviewer found the advantages claimed for *Text Tanglers* in its documentation to be exaggerated. Letter-by-letter decoding of words, sentences, and paragraph is not a very communicative or authentic activity no matter how many software programs use this technique. There seems to be no doubt that intensive focus on a text increases learners' "exposure" to it, and for teachers who feel that the fun of working with a program as lively and flexible as *Text Tanglers* is useful and pedagogically sound, this program has some merit.

J.E.T.T. Contributor Profile
Nina Garrett is a Research Associate at the University of Illinois and a *J.E.T.T.* contributing editor. Interested readers may write to her at the following address: G70 FLB, 707 South Mathews, University of Illinois, Urbana, IL 61801.

Text : Templates :: Fuel : Engines
Creating CALL materials using just your word processor

Presentation prepared for TESOL '92, Vancouver

by **Vance Stevens**
Sultan Qaboos University, Sultanate of Oman

The following programs have in common that they all work off ascii text files which can be created and altered using almost any word processor (in text editor mode). They are organized according to preparation of text required; i.e. almost nil, importation only, and some enhancement or alteration of text required.

1. The following programs operate on raw ascii text:

Text Tanglers, by Vance Stevens and Steve Millmore
Availability: Research Design Associates / P.O. Box 848, Stony Brook, New York 11790 / USA

Super Close, by Steve Millmore and Vance Stevens
Availability: Examination copies available through CALL-IS/TESOL
Registered copies from Vance Stevens, Language Centre, Box 32493, Al-Khod, Oman

Hangman in Context, by Steve Millmore and Vance Stevens
Availability: Examination copies available through CALL-IS/TESOL
Registered copies from Vance Stevens, Language Centre, Box 32493, Al-Khod, Oman

These programs all create interactive lessons directly from ascii text files by permuting the texts and having students reconstruct their original content or order. As feedback, the programs let students know if they have managed to restore the text as it was originally, and in most cases provide hints, such as revealing bits (or all) of the text as requested by users.

Text Tanglers contains the following modules: cloze, cryptogram, hangman, jumbled sentence, jumbled paragraph, and a text viewer. Super Cloze and Hangman are enhanced versions of the Text Tanglers modules. **Super Cloze** allows users to specify deletion ratios, mark targets for deletion in the text itself, and create special word lists with wild card characters for optional use as targets for deletion. **Hangman in Context** is the familiar game with the addition of a feature that allows students to reveal bit by bit the context from which the targeted word has been taken in the original text.

Cloze, by Stuart Luppescu
Availability: Freeware, available through CALL-IS/TESOL

This program produces cloze passages directly from ascii text. Users have several options; e.g.: set deletion ratio, set number and length of blanks, print initial letter of targeted word, set number of sentences at start of passage with no deletions. The program outputs its result to a file which can then be printed and distributed to students.

Concordancers give you each instance in a text where a key word is used. Standard format for concordance output is KWIC, or key word in context/centre. Teachers who encourage students to use concordancers believe that the students will benefit from what Tim Johns has called data driven learning. There is an excellent comparative review of commercial concordancers in Higgins, 1991, *System 19*:91-117. Below are some readily available concordancers not mentioned there:

Concord, by Stuart Luppescu
Availability: Freeware, available through CALL-IS/TESOL

Concord, by Dave Poulton
Availability: Demo version available through CALL-IS/TESOL
Further information: Dave Poulton, Language Centre, Box 32493, Al-Khod, Oman

One-line concordancer
Just type this line at DOS prompt: *for %a in (*.*) do find "keyword" %a > > filename*
Replace *.* with your ascii files; *keyword* your concordance string; *filename* with any file name.

2. The following programs operate on raw ascii text imported into program:

Eclipse, by John Higgins
Availability: Examination copies available through CALL-IS/TESOL
Registered copies from John Higgins, 14 Alma Road, Clifton / Bristol BS8 2BY, UK

Eclipse is a whole-text deletion program whose success lies in the fact that students can play against each other or against the computer. In the latter mode, the program continually feeds students context from which to make intelligent guesses. Eclipse provides a rich range of options by which students can elucidate the text; for example, having word beginnings or endings exposed.

Sequitur, by John Higgins
Availability: Research Design Associates / P.O. Box 848, Stony Brook, New York 11790 / USA

With Sequitur, students are presented with the beginning of a discourse, and three possible follow-ons. One of the three is the next follow-on as it occurs in the text, and the others are distractors taken randomly from other parts of the text. Once the correct answer has been selected, it is incorporated into the discourse and the process continues with the next follow-on and two distractors until the text has been fully restored. Students may work in prose, dialog, or verse mode.

3. The following programs require enhancement or alteration of text:

Hype, by Bill and Bev Thompson, AI Expert Magazine, San Francisco
Availability: Public domain, available through CALL-IS/TESOL

Originally conceived as a hypertext template, Hype has potential for development of more imaginative CALL lessons. Preparation of text consists mainly of marking elements in text to act as hot keys and constructing and labeling passages to act as feedback when the hot keys are invoked.

Tic Tac Know, by Dave Poulton, Sultan Qaboos University
Availability: Academic domain, available through CALL-IS/TESOL

Tic Tac Know is a game awarding naughts and crosses only in return for correct answers to questions which exist (with answers and distractors) in the form of an ascii text file.

Speed Read, by Dave Poulton, Sultan Qaboos University
Availability: Academic domain, available through CALL-IS/TESOL

Though possibly not so effective as a device for improving reading speed, Speed Read is nevertheless an attractive template for presenting true-false, multiple choice, and short-answer questions based on timed text passages.

CALIS; latest version 2.3, 1992
Availability: Humanities Computing Facility, 104 Languages Bldg., Duke U., Durham NC 27706 / USA

CALIS can be complicated to work with, but it is nevertheless a template program operating off ascii texts created with word processors or text editors.