ACE Maintainer's Notes
Created: 4/29/85
Revisions:

INTRODUCTION

This document is intended for programmers who intend to fix, update or otherwise significantly modify the ACE code. A general description and several specific points about the Animation Compiler and Environment will be brought out This is neither a user's guide nor a guide for animation. A user's guide may be found filed under {ICE}<TURNER>ACE>ACE-USERS-GUIDE; a technical document is filed under {ICE}<WHOKNOWS>ACE>WHOKNOWS.

OVERVIEW

The Animation Compiler and Environment is a program for creating, editing, displaying and storing/retrieving frame-oriented animations. It is based on traditional animation principals; successive frames (each frame usually slightly different from the previous frame) are displayed at a rate fast enough to create the illusion of motion (atleast 10 frames/second; usually more). To the user, the ACE system presents a sequence of whole frames (represented as bitmaps with the same width and height); the analogy being a stack of paper. In reality, however, the program maintains virtual frames. This decreases the storage requirements and allows the very rapid display of frames.

VIRTUAL REPRESENTATION

The virtual frames maintained by ACE are differential frames; that is, a frame only contains the information that is different from the previous frame. By this mechanism, the first frame is a complete frame (complete bitmap) providing the intial information. The second frame, then, is just the differences that exist between the first frame and a hypothetical complete second frame (i.e. just the differences between two bitmaps). And so on with all successive frames. The details of data structures and the compilation process are given below.

TRILLIUM vs STAND-ALONE

Currently, ACE is designed to operate as either a stand-alone program or in the Trillium environment. Conceptually, as a stand-alone, ACE should be considered more procedure oriented. When ACE is called (top level fn: (ACE)), a control window (ACE.CONTROL.WINDOW) is brought up which contains a menu. This window and menu remain up and stay active; once ACE is activated, it becomes a part of the user's current environment. In Trillium, ACE operates more functionally; that is, ACE is called as a one-time function which returns a value (although, ofcourse, it produces side-effects). The control window and menu are only active during the function call and are taken away when the ACE session is concluded. Also, a special animation-run-time function is provided for Trillium to use outside of ACE to run animation item types. At present, Trillium expects all editing (and most loading and storing) of animation sequences to be performed inside ACE.

ORGANIZATION

ACE is organized in a top-down fashion which allows a general to specific description of it. What the components do and how (where appropriate) will now be outlined.

FILES: There are four files to the ACE system (and one utility: RS232): ACE, ACE-MAIN, ACE-EDIT and ACE-PRIM. The divisions exist partly for organizational purposes and partly for ease in locating and working with segments. The ACE file contains some macros and variables, but exists primarily to load in all the necessary files. ACE-MAIN contains all the functions necessary to define, manipulate, run and load/store animations. ACE-EDIT contains all the editing functions (including interfacing to the MM1201 graphics tablet). ACE-PRIM contains the compiler funtions; it is only concerned with compiling two frames to create a virtual frame.

ACE-MAIN

This is the guts of the animation environment. There are four main divisions in MAIN (you can see them by looking at the file coms): Top level fns, Trillium-geared fns, I/O fns, and "helper" fns. In addition, there are several macros and a GLOBALVARS declaration. All the main functions that work with animation sequences are in the first division (including the startup fn: ACE). The I/O section is just for reading and writing files; user input and output fns are in the helper fns section. Control window operation and clipping region fns are also located in the helper fns section.

ACE-EDIT

This segment also consists of four parts (again, note the file coms). The first section provides

entry/interfacing to MAIN and calls to the actual editing routines. LINEART contains all the major fns for line art drawing. The third part contains all the other editing fns (e.g. painting, text, moving, etc.). The last part consists of tablet access fns and helper fns (all MM1201 code is located in this section and the code to read the current input device). The RS232 package is used by ACE-EDIT to read the MM1201 Summagraphics tablet. At present, the RS232 package (RS232.DCOM) is loaded by the ACE file. However, the maintainer should keep aware of changes in this package and its whereabouts.

ACE-PRIM; How the Compiler Operates:
The compilation process is concerned only with compiling two frames at one time (MAIN and EDIT take care of administration). The entry function is ACE.COMPILE.FRAME with arguments: BM.ORIG (the first or original bitmap), BM.CHANGED (the successor bitmap), VERTICAL.BLOCK (defines the height in scanlines for primitive regions of change; the horizontal component is 16 so as to take advantage of the 11xx series word size), and THRESHOLD (the percent as an integer representing the minimum amout of "changed area" allowable in a combination; more on this later). Take note that there are two variables defined in PRIM: ACE.PIXPERWORD (16, just the 11xx word size) and ACE.BITMAP.MASK (a mask used for ignoring extra bits in the last raster word of a bitmap).
The compiler works by comparing each word in BM.ORIG to BM.CHANGED (NOTE: these bitmaps must have the same dimensions) VERTICAL.BLOCK words at a time. If a changed word is found, a region specification is entered on list denoting this "region of change". After the whole area of the bitmaps is checked, the compiler attempts to merge these smallish changed-regions (see ACE.MAX.REGIONS). The algorithm first attempts to combine regions which result in no space wastage (this is always desirable). When no more 100% regions can be formed, the algorithm tests all pairings of primitive regions to find the highest efficiency (i.e. least space waste). If this effeciency is greater than or equal to TRESHOLD, the two regions are combined, otherwise, the algorithm terminates. The region merging process is the slowest aspect of ACE; the problem of merging regions is thought to be NP-complete.

DATA STRUCTURES
- A sequence is of the form: (FRAME FRAME FRAME ...).
- A FRAME is of the form: (DELAY BLITS).
- A DELAY is an integer representing a time delay in milliseconds.
- BLITS is a list: (BLIT BLIT BLIT ...).
- A BLIT is of the form: (BITMAP XCOOR . YCOOR).

A BLIT is essentially a small changed area with information on where it should be placed (relative to the animation). The record defintions for the above structures are in the file ACE. The compiler uses the structures: REGION : (LEFT BOTTOM WIDTH HEIGHT); a modified REGION : (REGION . AREA) for merging; and lists or both of these types.

GLOBALVARS
The following are important global variables with explanations where needed.

- ACE.CONTROL.WINDOW contains the top-level menu and status information.
- ACE.DIRECTORY is a default directory used to store/retrieve files.
- ACE.SEQ.WINDOW the current window where animations are displayed.
- ACE.SEQ.WIDTH and ACE.SEQ.HEIGHT refer to the current sequence.
- ACE.SEQ.WINDOW.XOFF and "."."YOFF the offset in the ACE.SEQ.WINDOW.
- ACE.CURRENT.SEQUENCE points to data which is the current animation sequence.
- ACE.CURRENT.SEQUENCE.NAME for retaining file name information.
- ACE.FRAME.TAIL tail of frames starting one after the current frame.
- ACE.CURRENT.FRAME a tail of frames starting with the current frame.
- ACE.VERTICAL.BLOCK value to use when compiling (see above on compiler).
- ACE.AREA.THRESHOLD for compiling.
- ACE.RUNNING.UNDER.TRILLIUM T if ACE was called by Trillium.
- Various .CURSORs are just cursors.

There are also a GLOBALVARS list of menus in MAIN and EDIT. The approach on menus was that they should be created only once to save both time and space.