ACE

By: Michel Denber (Denber.wbst@Xerox.com) Compiled for Medley by Larry Masinter (Masinter.PA@Xerox.COM)

Files: ACE.LCOM

Data files: ACE-APPLEDEMO.ACE, ACE-BOUNCINGBALL.ACE, ACE-FOUETTE.ACE

Animation Compiler and Environment

Introduction

ACE is a system for computer-assisted animation. It is based on the traditional cel-oriented animation process with the computer taking over many of the tedious jobs. You enter a succession of frames which represent a *sequence*. The system then plays back your frames to create the animated effect. It lets you draw pictures, enter text, and edit your work. The animated images you make are displayed on the screen in real-time. The two main parts of ACE system are a frame compiler and an environment. The environment provides the editing tools, frame manipulation, and display capabilities. The compiler operates automatically to produce a compressed-storage representation for frames.

You can also use the graphic editing features in ACE to make individual pictures, whether or not they're intended to be used for animation. Finally, you can use the compiler directly to compress any bitmap image so that it take up less space on your disk.

The majority of the code for ACE was originally written by Paul Turner, a student at the University of Rochester. I am currently maintaining the system. Please send all bug reports, comments, and suggestions directly to me, Denber.WBST, or Denber.WBST@Xerox.COM (Arpanet). This document describes the features available in ACE version 2.1.

Background

In this document: *holding* the mouse on a menu selection means to press down a mouse button on a menu item (inverting the item) and keeping it down for about 1.5 seconds (at this point, you can release the button or move to another selection). *Clicking* the mouse means pressing a mouse button down and releasing it. Unless otherwise stated, the left mouse button is used for selecting items from menus and to click at objects.

In addition to the mouse, ACE supports a graphics tablet (*Summagraphics MM1201*) . [LMM: The graphics tablet hasn't been tested in Medley.] The tablet is more convenient for doing free-hand drawing; in fact, most commercial animation systems include a graphics tablet. The pen has two buttons: the stylus *tip* (which is activated by pressing down on it) and a blue button on the *barrel* of the pen (activated by pressing with the forefinger). We have adopted a convention with regard to the tablet: the stylus button acts like the left button on the mouse and the barrel button acts like the middle button on the mouse.

Terms and System Organization

A *region* is simply a rectangular area.

A *frame* is a region that contains one complete "picture" in an animation sequence; it is a rectangular bitmap with a fixed width and height.

A *sequence* is a collection of frames defining one complete animated segment.

The *current frame* is the frame in a sequence to which operations will be applied. As all the frames in any given sequence are the same size, you may say that one characteristic of a sequence is a region of a particular size. Frames are referred to by number for convenience; the numbering is from 1 to n.

There are two principal windows used in ACE. The *sequence window* is the window on the screen where a particular sequence will be created, edited and displayed. Typically, you define the shape of the sequence window to give you just the area you want to work in, although a sequence can also be edited and displayed in any existing window.

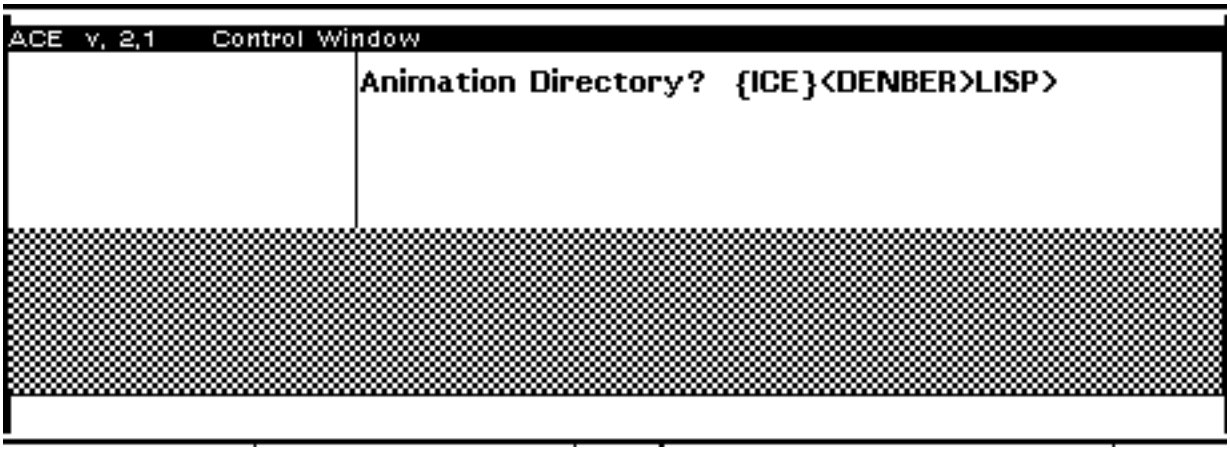
The *ACE Control Window* holds a menu of commands and displays animation state, prompt, and help information. The upper left region in the control window, referred to as the *status region*, tells which frame is currently being displayed (which frame is the current frame); which device (mouse or tablet) is being used in line art and painting operations; what operation is currently being performed; and, the size of a region (width, height) or the location (x, y) of the cursor within the sequence. The upper right portion of the control window is the *prompt region*; it is used to get user input and display helpful information. The bottom part of the control window is a menu of animation functions.

GETTING STARTED

Load ACE.LCOM from your lispusers directory (e.g., (FILESLOAD (SYSLOAD) ACE). When this is complete, type:

```
(ACE)
```

At this point an Ace control window will appear by the cursor. You can place it wherever you wish on the screen. The window initially contains a prompt "Animation Directory?" asking for a default directory to use for storing and retrieving animation files (The default selection is your login directory. Just press the return key to accept the default.) The control window can be moved around just like any other window. While you never need to "quit" from ACE, if you close the control window with the right mouse button menu it permanently aborts ACE; if you then re-type (ACE), the animation system will be restarted from scratch.

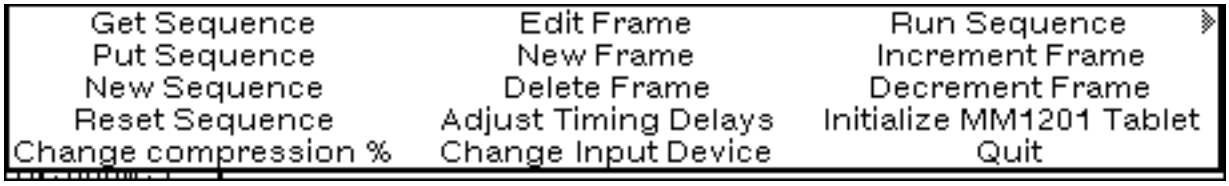


ACE Control Window at start-up

ACE Commands

Main Menu

The menu selections from the control window will now be described; an example of using ACE is given in the next section. The main menu is divided into three columns. The left-most one contains commands that affect the entire sequence, the middle column operates on frames, and the right-most column contains utility commands.



Sequence commands (left column)

Get Sequence Loads a sequence-file from a file server or local disk. You are prompted for the name of the file; incomplete file names will be completed from the directory given as the default ACE directory. You will then be prompted for a window specification. Unless you want to display the sequence in a particular window, select 'create window automatically'.

Put Sequence Saves the current sequence to a file. You will be asked for a file name and given the option to overwrite the existing version of the file (if any) or create a new version.

New Sequence Discards the current sequence (if any), and prompts you for a new sequence by requesting a size for the new sequence. Dragging out a rectangular region with the mouse; the exact size of the region is displayed in the status region of the control window. A blank first frame is created, ready for editing.

Reset Sequence "Rewinds" the current sequence to the beginning (i.e. there is no current frame and the next frame to be displayed will be the first frame).

Change compression % This can be used to change the amount of space compression performed by the animation compiler. For general use, there is no need to ever call this command.

Frame commands (middle column)

Edit Frame Allows editing on the current frame. Brings up a menu of editing options, described in the next section.

New Frame Inserts a new frame after the current frame (ie. before the next frame). The frame editor is then automatically invoked.

Delete Frame Deletes the current frame. The current frame is removed and the previous frame become the current frame. The first frame can not be deleted.

Adjust Timing Delays Lets you set the amount of time (in milliseconds) that any particular frame is displayed; for example, a delay of 50 on the 5th frame would mean that the 5th frame will be visible for 50 milliseconds before the 6th frame is put up. You can change the entire sequence or a single frame at a time. For individual frame setting, you get a menu of frames and their current delays. Holding down a selection will display that particular frame in the sequence window (this is also a convenient way to rapidly move to an arbitrary frame); if you select a frame you will be prompted for a new delay value. When new frames are created, they always get a default delay time of 0.

Change Input Device Lets you select either 'mouse' or 'tablet' from a menu. This sets the device to be used for line art and painting operations. The status (upper left) region of the control window always shows which device is active. All menu selections have to be done with the mouse, even when the tablet is being used for drawing

Utility commands (right column)

Run Sequence Runs the *remainder* of the sequence. To run the entire sequence, select Reset Sequence before Run. This command has a submenu with two additional commands:

Loop Runs the entire sequence in a continuous loop. To stop the loop, hold down the space bar. This is checked only at the end of the sequence, so just tapping the space bar may not stop the loop.

Loop part Runs a portion of the sequence in a continuous loop. You can specify the starting and ending frame numbers. To stop the loop, hold down the space bar, as in Loop above.

Increment Frame Displays the next frame and makes it the current frame.

Decrement Frame Goes back to the preceeding frame and makes it the current frame.

Initialize MM1201 Tablet This performs the necessary RS232 port initializing, sets the baud rate, activates the tablet, etc. This must be called after the tablet is plugged in and before it is used. If your tablet doesn't seem to be responding, it may need to be reinitialized.

Edit Frame Submenu

For the commands described below, it is sometimes useful to know the exact coordinates at which a drawing operation will take place. If you hold the T key down, ACE will put the current coordinates in the status window. Release the key to stop this function.

When you select **Edit Frame,** a sub-menu of editing options appears. Their functions are as follows:

Paint This lets you to paint on and erase bits from the current frame. The painting operation is the standard Interlisp-D window paint command. Either the mouse or tablet can be used (which ever device is currently selected). Pressing the left mouse button or pen stylus draws; the middle mouse or pen barrel button erases. Pressing the left shift key brings up a menu. From this menu, you can quit painting, change the brush size or shape, and the color or texture of the "paint brush". Note: selecting items from this menu requires using the mouse (unfortunately, the tablet cannot be used for menu selecting). For more information on the paint command, please see page 19.20 in the Interlisp manual.

Line Art This lets you add straight lines to a frame by selecting one vertex, dragging out a line, and then selecting another vertex. In this way, an arbitrary string of connected line segments can be created. The left mouse button or pen stylus will "put down" vertices, the middle mouse button or pen barrel stops the line dragging. The right mouse button brings up a menu of line art options (paint or invert drawing; several line width choices); as with all menus, the menu selection must be made with the mouse.

Edit Bits This lets you use the Interlisp-D bitmap editor on the selected frame. The mouse is used to turn specific bits on or off (the tablet is not used as it isn't helpful for this application). A complete description of the the bitmap editor is given in the Interlisp Reference Manual. Always exit the editor by selecting **OK**.

Text Lets you put text into a frame. After selecting the 'Text' option, you will be prompted for font characteristics. Then you point (with the mouse) to where the text should begin and click the left mouse button. You may now type in text from the keyboard and it will show up in the frame. A press of the return key ends text entering (the return is NOT included in the text).

Move Region Lets you move an arbitrary rectangular region in the current frame. You first drag out the region to be moved, then you will be asked what to do with the old image (i.e. leave it alone, erase it). At this point, a ghost image will attached to the cursor and you can position the image in its new location. Clicking the mouse will set the position for the image and then you will be asked how to combine the image (i.e. paint it in, exclusive-or it in).

Combine Region This is similar to move region, except that you can select any region on the screen (not just inside the current frame). After selecting the region, bringing the mouse within the sequence window will show a ghost image; the rest of the procedure is the same as for **Move Region**. This command is extremely useful for bringing images created elsewhere into your frame. For example, you might have a drawing made using Sketch or an AIS file.

Texture Area Fill This lets you fill in an arbitrary closed curve with a texture pattern. You are first prompted to select a bounding region. This reepresents a maximum area beyond which texturing will not occur, in the event that the texture "spills" outside the region being shaded. Next, select a starting point anywher inside the desired region. You will then be offered a menu of predefined textures. You can choose one of these, or create your own by selecting *** Other ***. The area is then filled with that texture. You can then confirm that the right thing happened by clicking left. Click any other button to undo the operation.

Texture Region Fill This is like Texture Area Fill, except that it is used to create filled-in rectangular boxes, rather than arbitrary areas.

Scale Region Lets you change the size of any rectangular area. You first select a region, and then indicate the shape of the scaled area by sweeping it out on the frame. Useful if you know how big you want the result to look but not what percent of the original it is. This command has a submenu:

To a new region Same as the top level Scale Region.

In x and y You first select a region as in **To a new region**, and then indicate the percentage of the original size to scale by, much like selecting reduction or enlargement on a copier. You can set the x and y scale factors independently.

In x only Use this if you only want to scale in the x direction, leaving y unchanged.

In y only Use this if you only want to scale in the y direction, leaving x unchanged.

Clear Region For clearing regions to white quickly. If you select a region within the sequence, it is immediately erased. There is no UNDO for this operation.

Quit - Compile This is the usual way of exiting the editor. This keeps the changes made to the frame and calls the compiler, resulting in adding the frame to the current sequence.

Quit - ABORT Exits the editor but does not update the frame. The sequence will be as it was before you selected **Edit**. Any changes made to this frame are lost.

CONCLUSION

Examples

You might want to see an existing animation before creating one of your own. There are several animation demos included in the Lispusers distribution of ACE: ACE-APPLEDEMO, ACE-BOUNCINGBALL, AND ACE-FOUETTE. The simplest one, BOUNCINGBALL, contains five frames showing a bouncing ball. To see this, start ACE running and select Get Sequence from the main menu. Type the name of the file, including the file server and directory if they are different from your current animation directory. Once the file is loaded ACE will let you position the sequence window. Then select Run Sequence. The system will display the five frames and then stop. To see the ball bounce continuously, select LOOP from the submenu on the Run Sequence command. The ball will now bounce until you hold the space bar down.

The file ACE-APPLEDEMO is a 125 frame sequence which shows an apple getting shot. ACE-FOUETTE is a six frame cycle of a ballet dancer performing a fouette turn.

Animation Hints

Remember that ACE is just a tool - it will not do any animating for you. Our goals were to provide a system that simplified the frame creation process and let you create animation on the computer without having to learn a special animation programming language. However, animation is an art form in itself. Experience is gained only through practice and experimentation.

Avoid moving objects too far between frames or the motion will appear jerky. The D machines screens are designed to mimimize flicker through the use of a long persistence phosphor. Unfortunately, this results in trailing streaks of light behind rapidly moving objects. Sometimes you can use this to artistic effect. It can be reduced by moving dark objects over a light background, rather than the reverse.

Sometimes you can simplify the animation process by creating frames out of order, especially for cyclic animation. For example, the bouncing ball was created by drawing frame 1 and then making a new frame (which is by default a copy of frame 1). Then we backed up to frame 1 and added a new frame between 1 and 2, showing the ball half-way down. Then this frame was copied, yielding four frames. Backing up again and adding the middle frame gave a symmetrical bounce sequence with frames 1, 5 and 2, 4 being identical.

It’s often convenient to keep a snapshot of the object you’re animating handy in a window next to the sequence window. It can then be brought into the frame whenever needed, for example in the case where it is being modified in some way. You are not limited to editing images with the ACE editor. In particular, you may want to use the *Sketch* editor (an Interlisp-D library package) to modify images, and then bring them into the sequence window for compilation.

References

Denber, Michel, Paul Turner, "A differential compiler for computer animation", to appear in *Computer Graphics*, **20**:3, 1986 (Proc. SIGGRAPH ’86)

Fox, David, Mitchell Waite, *Computer Animation Primer*, McGraw-Hill, N.Y. 1984

Magnenat-Thalmann, Nadia, Daniel Thalmann, *Computer Animation: Theory and Practice*, Springer-Verlag, Tokyo, 1985