BreakApp

BreakApp is a simple game program where the goal is to knock out the tiles in a playing field by bouncing a ball off the walls and a paddle without letting the ball drop past the paddle. Because BreakApp is meant to be an example of using NeXTSTEP, it's not full of features that would make it an terribly exciting game. That is left as an exercise to the reader. Instead, BreakApp demonstrates:

- Simple animation techniques
- Use of the SoundKit
- Use of NXImages with custom draw methods

BreakView, a subclass of View, is the central class in BreakApp; it implements all of the game functionality and provides several target/action methods with which the game play can be controlled. The BreakApp class, a subclass of Application, provides some housekeeping functionality (such as management of nib files, etc). SoundEffect is the class responsible for dealing with the SoundKit.

BreakApp uses a buffered output window as a means to fake double-buffered animation. The current frame is drawn directly into the window. However, because the window is buffered, the drawing goes to the backing store, and not the screen. Only when the frame is complete does BreakApp flush the windo@3ntents to the screen; this process is fast and provides a flicker-free update. BreakApp then proceeds to draw the next frame into the backing store.

Changes since 2.0

· BreakApp used to run with a timed entry set to fire as often as possible. Now it sets the period to 0.03 seconds. Because timed entries have been changed such that under 3.0 they do not add the time it takes for the timed entry function to execute to the period, this pretty much guarantees about 30 frames/second on machines capable of 30 frames/second or more. It also assures that on very fast machines BreakApp won't hog the CPU by running at hundreds of frames a second.

· The sound effects in BreakApp are now generated with the SoundKit. Under 2.0, because sampled sounds could not be mixed in real time, BreakApp used the MusicKit.

· Minimum for the game window size is now set through Interface Builder; thus, there's no need to get the windowWillResize:to: delegate method.