

gamedeveloper

THE LEADING GAME INDUSTRY MAGAZINE

SKUNK WORKS

EXCEPTIONAL

GREAT

FAIR

POOR

UNFORTUNATE

LIPSYNC SDK 3.3



STATS
ANNOSOFT LLC
2050 N. Collins Blvd.
#117
Richardson, TX 75080
972.234.6985
www.annosoft.com

PRICE
\$3,000-\$5,000

SYSTEM REQUIREMENTS
Windows 98 or higher.
Pentium 400MHz or higher.
Mac OS X or higher, G4 or higher.
These requirements are for Lipsync SDK 3.3 only (as opposed to Text Based, Textless, or Lipsync's desktop tools). For system requirements of other versions, see Annosoft's web site.

- PROS**
1. Easy to integrate into an existing pipeline.
 2. Produces good animation results.
 3. Provides good text timing analysis and allows for custom user data.

- CONS**
1. Should provide better sample model mouth shape references.
 2. Pronunciation override feature needs improvement to completely ignore the sound file in deciding final mouth shape.
 3. Would be nice if some sample Maya plug-ins and batch scripts were made available to make the SDK plug-and-play ready for production pipelines.

ANNOSOFT'S LIPSYNC SDK 3.3

BY BIJAN FORUTANPOUR

CHARACTER ANIMATION HAS ALWAYS been a labor-intensive mixture of pleasure and pain. Not surprisingly, many new technologies strive to tip the scale toward fun and increased efficiency: motion-capture, ragdoll physics, inverse kinematics, and the list goes on.

Annosoft has a new piece of magic: its Lipsync SDK with speech analysis using statistical modeling techniques and Hidden Markov Models. In simpler terms, it's "free" lip-synch animation. Put the sound file in, and animation curves come out.

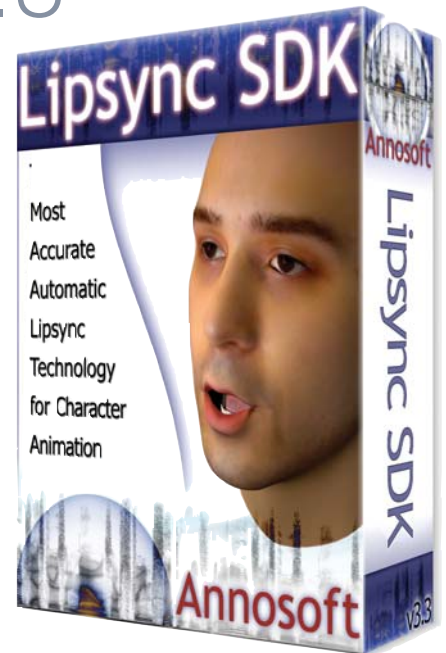
THE PROBLEM

Annosoft's Lipsync is a software development kit that creates character lip-synched animations by analyzing an audio file of spoken dialog. There may be hundreds if not thousands of spoken phrases in any given game, with each phrase taking anywhere from a few seconds to a minute or more.

Let's estimate that an animator (using a standard 3D package) requires approximately four hours to keyframe mouth blend shapes

for a 20 to 60 second segment of dialog. For a game with 500 segments of dialog, it would take (500 x 4 hours) 2,000 work hours, or one work year. Many games easily have 2,000 pieces of dialog, and some MMOGs currently contain between 50,000 and 100,000 segments of dialog. Clearly, creating realistic-looking mouth shapes for this volume of dialog is not worth the investment to most game development companies.

In contrast, any artist tool or plug-in developed using the Lipsync SDK will be able to generate animation curves corresponding to the input dialog sound file in approximately 5 seconds for every 60 seconds of dialog. Therefore, using a batch process it would be possible to process 2,000 dialog sound files in less than an hour. Of course an animator would still have to preview it and perform quality control of the results, but even that step could be greatly accelerated by developing a batch render process to generate one or more movie files of the resulting animations.



Annosoft Lipsync SDK is targeted specifically toward 3D game development (although it may also be used for animated 2D Flash presentations) and is also available as an ActiveX control for scripting using Microsoft Visual Basic. There are actually three versions of the SDK: Text Based, Textless, and Realtime. As one would expect, each version produces a level of quality depending on the amount of information input to the system. The company also licenses a standalone desktop application called Lipsync Tool (not

an SDK), which I'll discuss briefly at the end of this article.

The Text Based SDK provides the highest quality animation. It requires as input the recorded dialog sound file (in .wav format), a simple text transcription of what is said, and a language-specific "rules of pronunciation" file. Annosoft provides customizable language pronunciation files in six languages: English, French, Spanish, German, Italian and Russian. When asked about the different dialects within languages (such as British, American, and Australian English), the company said that, to date, there haven't been any quality problems related to this issue, but dialect lexicons can be done upon request. The Textless Lipsync SDK only requires a dialog sound file. And finally, the Realtime Lipsync SDK just requires a live sound stream coming in, usually from a microphone. An important side note is that the Realtime SDK is not available for any game console at this time.

QUALITY VS. SPEED

The quality of animation delivered is very noticeably different between the SDKs. By "quality," I mean the accuracy of the results: Are the actual phonemes and mouth shapes detected correctly? Are the millisecond timings of the mouth shapes in sync and accurate?

I would highly recommend using the Text Based Lipsync SDK if real-time reactions are not required. The slight additional effort it takes to provide a text file of what is said is well worth the much larger time saving you'll get during the animation process. If an artist is still required to make manual adjustments to the final animation, the hours you saved will quickly disappear; in fact, repairing someone else's animation is often more difficult and time consuming than just doing it all over from scratch. Annosoft's Text Based SDK does deliver lip-synch animation that's high enough in quality to meet the demands of today's 3D games with extremely little artist tweaking, if any. It contains some hidden gems and secrets as well, such as customizable word pronunciation and custom user data hooks (blind data), which can be used for developing automated facial expressions and automated subtitling and closed captioning systems.

At the other end of the spectrum, the animation quality delivered by Annosoft's Realtime Lipsync SDK is very hit-or-miss. It correctly handles the silences and

pauses between words, and some sounds are handled better than others. The animations are not hero-quality, but may potentially be passable for an avatar or chat room talking head.

THE COMPROMISE

In the mid-range quality between the Realtime and Text Based versions lies the Textless SDK. There's a noticeable difference in quality between the Text and Textless, but if you're working in languages that are not currently supported by the Text Based, this is still a strong option for you. The documentation states that this version has been used successfully with Chinese, Japanese, Arabic, and Hebrew.

With the Textless version, having fast talkers actually helps. If the speaker on the sound file is talking quickly, the results are slightly better on the game characters because the program spends less time creating each mouth shape, so there's less room for error.

The difference in quality comes not so much in the timings of the mouth shapes, but the actual mouth shapes detected. For instance, 'n' and 'm' sound almost identical but have very different mouth shapes when a person speaks, which affects the look and believability of the animation.

PUTTING IT ALL TOGETHER

Sound markers—the set of information needed to define the mouth's shape, the time the sound starts, and its duration—can be read from a file or accessed directly by linking with the Lipsync library. Included in the documentation and code samples is a very illuminating and useful C++ project called "cmdlinesync." Cmdlinesync is a standalone command line executable that writes the sound markers to an ASCII file. Free of any windows or GUIs, it's perfect for using as-is, whether in a batch process or as part of a plug-in or script for 3D packages like Alias Maya or Softimage XSI. Even though Cmdlinesync was provided as a how-to file (on using the lip-synch library), it had such a clean interface and the output file was so simple that I decided to just write a quick parser for the output instead.

A sound marker is written out, one per line, in the output file and consists of: Start time (milliseconds), End time (milliseconds), Intensity or "how much of a mouth shape to use" (0–100), and the phoneme (a one or two letter label). One additional field is the mark-

er type, which is usually "phoneme," unless you're working in the Text Based SDK, in which case you can choose from a list of provided marker types: word, sentence, or XML marker.

The word and sentence markers provide the start time and end time of a word or sentence and can be used to implement an automated game subtitling or closed-captioning system. The XML markers provide a few important features.

First, it can pass pronunciation override commands to the Lipsync SDK. In rare and difficult cases, if a word doesn't look correct in the animation or has special pronunciation rules, one can specify how that word is to be pronounced correctly. For instance, for CIA, using `<pron value="see eye ay"/>`, ensures perfect animation every time. Another useful XML tag is the `<pause/>` tag for forcing a slight pause and distinguishing words.

For example, I tried an Australian accent of "g'day mate" a few times. One came out perfectly and the other one didn't. Lipsync had missed the "g'd" part, thinking it was part of the previous word even though there was a good pause in the sound file. I tried correcting it with the `<pron>` tag, to no avail, and finally hit success with a `<pause/>` tag, which helped to distinguish the previous word. My impression was that using the `<pron>` tag has varying levels of effectiveness or that it takes practice, and at some point it may be best to simply fix the issue directly in a 3D animation package.

The second and much more potent use of the XML markers is to add your own custom data, which Lipsync safely ignores but does timestamp. For instance, an expressionless face with the lips moving isn't very compelling. But with an `<angry value="5"/>`, `<sad value="3"/>`, or `<eyebrows value="2"/>` you can see how natural dialog draws out not only lip movement but character and emotion, too.

ART AND FINESSE

Even though the Lipsync SDK performs well, there remains the task of integrating it into a production pipeline. The first step developers need to take is to get the tool into their artists' hands. Whether they use Alias Maya, 3ds Max, Softimage XSI, or LightWave 3D, a custom plug-in needs to be written that will:

- allow artists to specify a sound file

- use the Lipsync library (directly or indirectly) to compute the analysis and return the sound markers
- create animation curves in the 3D animation package, setting keyframes corresponding to the given timestamps and corresponding mouth shapes at the resulting intensities.

In my case, a few weeks were allocated to the task, but after only a few days, with help from an animator, I had something up and running inside Maya.

With Lipsync SDK, the most important thing you can do to achieve ultimate animation performance is to build the mouth shapes carefully and label them correctly so they can be referenced by the plug-in.

Annosoft's web site offers some examples of how mouth shapes should look (check under Phonemes in the Display menu). However, the page has a disclaimer stating, "The author is not an artist."

After looking closely at the diagrams of the suggested plausible mouth shapes—not all of which I agreed with 100 percent (disclaimer: I am not an artist either)—the first thought that came to mind was "Then get an artist!"

The model in the sample applications and documentation looks very unnatural. Both a better model and

better texturing would vastly help make the animation look more convincing. Critics with an untrained eye have a hard time distinguishing between when to blame the art and when to blame the animation when something just doesn't look right.

It will take some time to get the perfect series of 10 to 12 practical mouth shapes that map into the 40 phonemes that Lipsync outputs. For the sake of time, I used the "plausible" set that Annosoft recommended, even though I knew it could be improved upon—but that is more of a lip-modeling improvement.

HISTORY AND CONCLUSION

The Annosoft Lipsync SDK has been under development for at least four years and is currently on version 3.3. Since version 2.0, new languages such as German, Italian, and Russian have been added, as well as many improvements to their internal algorithms for multiple pronunciations.

Lipsync Tool, the desktop application that I mentioned briefly at the start, is available as a standalone tool for about \$500. I feel that the application should either be included in the price of the SDK, or greatly improved to justify the price tag. The audio waveform

display is a bit odd and not very helpful, and the quality of the 3D head needs improvement to help judge the animation. The tool does output a 3ds Max script to create animations, but lacks other 3D file formats like Maya. Some simple things like window resizing while avoiding stretching the 3D view are essential.

Working with Annosoft SDK from a software development standpoint was very simple. It does one thing, it does it well, and has a simple interface. The bulk of the work involved as a game developer was focusing on mouth shapes and 3D animation package issues. Frankly, I'm surprised Annosoft hasn't produced plug-ins for the major animation packages already, but if they're still focused on improving the magic behind the SDK's speech analysis, I say more power to them. We need more compelling and convincing 3D characters in games. ❖

BIJAN FORUTANPOUR is a senior graphics programmer at Sony Online Entertainment in San Diego. He has worked in the visual effects and game industries for 11 years, four of them specifically in video games. Email him at bforutanpour@gdmag.com.

place logo here