SPEAKEASY
GAMEMAKER LIPSYNC RUNTIME

OVERVIEW:

SpeakEasy by ShroomDoom Studios is a GameMaker runtime designed to play back keyframe data exported from the lipsynching program Papagayo.

Import `.dat` files from Papagayo as included files into your GameMaker project. Parse them and then play back their keyframe data in-game using the runtime.

SpeakEasy is ideal for animating character mouth shapes during in-game dialogue.

Papagayo is a GPL-licensed open source lipsynching program originally designed to work with the animation package Anime Studio.
It can be used to more quickly and easily keyframe out cartoon character mouth animation.

Download Papagayo for free from Lost Marble: *http://www.lostmarble.com/papagayo/*

CREDITS:

Photorealistic mouth shapes are from the **Preston Blair phoneme series** by **Gary C. Martin,** available for free use at *http://www.garycmartin.com/mouth_shapes.html.*

RESOURCES:

GMC bugs & support thread: *http://bit.ly/1sLmP6I*
GM Marketplace asset page: *http://speakeasy.shroomdoom.com/*
ShroomDoom Studios website: *http://blog.shroomdoom.com/*

TIPS:

Papagayo sometimes exports timelines in which keyframes are out of order; this may or may not be a bug. At runtime, the timeline handler just skips over out-of-order frames. This means that playback won't break altogether, but it won't follow the timeline exactly. You may want to go into your .dat files with a text editor and make sure there are no jarring out-of-order frames. (There may be an implementation to remove out-of-order frames from timeline data automatically in the future.)

Papagayo by default exports keyframe data at a 24 FPS framerate. GameMaker by default runs at 30 FPS, and your game is likely running at 60 FPS.

The runtime converts between framerates to sync up keyframe playback with audio. If your Papagayo data is exported at a framerate other than 24 FPS, you'll need to specify that framerate in your timeline by modifying `obj_timelineHandler`'s `timelineFPS` variable.

Sometimes the location of included files inside the GameMaker project gets messed up, especially if you start using group folders. Clean your project asset compiler cache. Make sure files in your .gmx file are in a location that matches where they are in the actual project folder.

Reserved names: `phoneme, tmln`

These are `enum` names, so don't use them as variables.

<u>DOCUMENTATION:</u>

`scr_timeline_create(`*`String`*` file)`

Parses input file and returns timeline as a 2d-array.

Returns: 2d-array

`scr_timeline_initEnums()`

Initializes the global variables and enums used by the runtime. Should be called at the beginning of room or creation of control object.

Returns: void

`scr_timeline_init(`*`instance`*` timeline, `*`audio`*` audioFile, `*`String`*` datFile)`

Initializes a specific timeline. MUST be called to enable timeline playback. Should call immediately after creation of timeline instance.

If you know Java, think of this as the timeline's constructor.

Returns: void

`scr_timeline_execute(`*`instance`*` timeline, `*`enum`*` tmln.ACTION)`

Adds a command to be performed to a timeline's command stack. See 'timeline commands' for a list of possible commands to execute.

Example:

```
scr_timeline_execute(tl,tmln.STOP); // reset animation, stop it
```

Returns: void

`scr_timeline_changeSource(`*`instance`*` timeline, `*`audio`* `newaudio, String newdata)`

Reassign the audio and data for a timeline; will stop and reset playback to frame 0 of timeline. If you want to automatically restart the animation,you'll have to explicitly call scr_timeline_execute(timeline,tmln.RESTART) after changing the source.

Returns: void

`scr_timeline_getDatForSound(audio snd, String sndprefix, String datprefix)`

Makes keeping track of corresponding sound files and .dat files easier. Ideal for projects with large numbers of sound files.

Returns .dat file that corresponds to a sound file.

Example:

Execute: `scr_getDatForSound(snd_lipsync_Woody2, snd_lipsync_, dat_lipsync_);`

Return: `"dat_lipsync_Woody2.dat",` *string name of .dat included file*

Returns: String, .dat file name

`scr_timeline_execute()` commands list

All commands should be used with the prefix `tmln.`

`TOGGLE,`	if playing, pause; if paused, play
`TOGGLELOOP,`	shouldLoop = !shouldLoop
`PAUSE,`	pauses timeline, saves position
`STOP,`	pauses timeline, resets position to 0
`RESTART,`	resumes timeline from 0
`RESUME,`	resumes timeline from current position
`LOOP,`	sets shouldLoop to true
`ENDLOOP,`	sets shouldLoop to false
`NULL`	null placeholder command - don't do anything

Not yet implemented:

`REVERSE,`	sets timeline direction to -1
`FORWARD,`	sets timeline direction to 1

TODO:

Pair sounds/data together in a data structure. (Sort of accomplished with `scr_timeline_getDatForSound`.)

Remove out-of-order frames from parsed timeline?
Fix HTML5 support. (File reading in-browser is currently broken in GM.)
Implement reverse-scrubbing of timelines.