ARCAN LUA API

Entry points

All entry-points are prefixed with themename_.

clock_pulse_() invoked at a fixed rate (default: once every 25ms)._

input_(inputtable) see IOTable._

video_event_(source, argtbl)_, source:vobjid, argtbl:vidtable format _resizes, completed transformations, ..._

themename_() invoked directly after initializing video, audio, events, ..._

on_show_() invoked after themename._

*All functions prefixed with * are considered experimental.*

Resource

resource_(name : searchmask(THEME_RESOURCE, SHARED_RESOURCE))_ → (exists ? true or false).

zap_resource_(name)_ → true or false, locates and deletes the referenced resource (in theme only).

open_rawresource)_(resource)_ → open resource for write (only one open allowed, in theme only).

write_rawresource)_(line)_, write line to opened rawresource.

close_rawresource)_()_, close/flush any opened rawresource.

Target

launch_target_(gametitle, launchmode)_, LAUNCH_INTERNAL ? → vid, aid.

target_input_(inputtable)_, inject an event into the running target.

suspend_target_(tgtvid)_, attempt to suspend target.

resume_target_(tgtvid)_, attempt to resume target.

System

kbd_repeat_(rate (ms))_ → enable keyboard I/O generating repeating events.

system_load_(resource)_ → load script and return function reference.

shutdown_()_ → kill targets, unload resources and shutdown.

Database

***store_key**_(key, value)_ → store key/value (strings) in database as theme-specific table.

***get_key**_(key)_ → get value (or nil) from theme-specific database table.

game_cmdline_(title)_ → generate execstr for a specific game.

list_games_(filtertbl)_ → query database for games matching filter, empty table for all.

list_targets_()_ → get a list of all targets.

game_info_(title)_ → get a gametable for title (or nil)

game_family_(title)_ → get the family for a title (or nil)

game_genres_()_ → list all distinct (unique) genres / subgenres.

Audio

stream_audio_(resource)_ → returns an aid for a stream (or nil)

play_audio_(aid)_ → start decode/buffer/playback for a stream

pause_audio_(aid)_ → try and pause (slightly unstable) audio stream playback.

delete_audio_(aid)_ → deallocate and stop playback (slight delay due to buffering).

play_sample_(resource, [gain])_ → load and immediately play sample (wav) with optional gain.

audio_gain_(aid, newgain (0..1), [time])_ → set or fade gain.

Video

load_image_(resource, [initial zval (0..255)])_ → vid

delete_image_(vid)_ → immediately delete and deallocate (won't emit event).

show_image_(vid)_ → wrapper for blend_image(vid, 1.0).

hide_image_(vid)_ → wrapper for blend_image(vid, 0.0).

move_image_(vid, absx (px), absy (px), [time])_ → reposition to absolute coordinates (absx, absy).

rotate_image_(vid, absangz, [time])_ →.

scale_image_(vid, xfact, yfact, [time])_ → relative scale (1.0 == initial size), xfact or yfact zero = force aspect.

resize_image_(vid, width (px), height (px), [time])_ → absolute resize, width or height zero = force aspect.

blend_image_(vid, opacity (0..1), [time])_

order_image_(vid, newz)_ → 0 = further back, 255 = foreground.

instance_image_(vid)_ → clone the vid (share resources, child dies if vid dies), returns newvid.

***link_image**_(vid, parent)_ → bind vid to parent coordinate system.

expire_image_(vid, lifetime)_ → invoke delete_image after lifetime ticks, emits event.

reset_image_transform_(vid)_ → remove all queued transformations.

instant_image_transform_(vid)_ → force all transformations, regardless of time.

image_mask_toggle_(vid, enumint)_ → toggle relative value lookup for a specific property.

image_mask_set_(vid, enumint)_

image_mask_clear_(vid, enumint)_

image_surface_properties_(vid, [time])_ → returns current (or future) surface properties as a surftbl

image_surface_initial_properties_(vid)_ → returns initial (at load time) surface properties as a surftbl

***image_program**_(vid, vertprog, fragprog)_ → load and associate a shader (GPU program).

render_text_(formatstr)_ → returns vid and table of lineheights.

fill_surface_(width (px), height (px), r (0..255), g, b)_ → generate a single-colored vid.

force_image_blend_(vid, boolint)_ → always alpha-blend (transparent images).

***push_video_context**_()_ → add to context stack (if there's any free stack slots).

***pop_video_context**_()_ → deallocate current context (won't emit events) and load pushed context.

Frameserver

play_movie_(vid)_ → start playback of preloaded movie.

load_movie_(resource, [loop])_ → launch frameserver, return vid,aid.

pause_movie_(vid)_ → unreliable (due to openAL).

resume_movie_(vid)_ → unreliable (due to openAL).

Collision / Picking

image_hit_(vid, x, y)_ → boolnum if x, y is on pickable image vid.

pick_items_(x, y)_ → returns ary of vids at x, y.

LED

set_led_(ctrl, led, state)_ → toggle led on or off.

***led_intensity**_(ctrl, led, val)_ → untested, for advanced led controllers.

***set_led_rgb**_(ctrl, led, rv, gv, bv)_ → untested, for advanced led controllers.

controller_leds_(ctrl)_ → number of leds associated with a controller.

IMGTable format

width, *height*, *x*, *y*, *angle*, *opacity*

GameTable format

gameid, *targetid*, *title*, *genre*, *subgenre*, *setname*, *buttons*, *manufacturer*, *players*, *input*, *year*

IOTable format

kind :- analog, digital
(kind == analog), devid, subid, source (mouse, joystick, ...)
(kind == digital), translated (bool), active (bool), devid, subid
(kind == digital, translated:true) number, keysym, modifiers

GameFilter

fill the table with desired options;
year (number or 0),
title (string),
genre (string),
subgenre (string),
players (number),
buttons (number)

Global Variables

- VRESW (num) set to the window/display width.
- VRESH (num) set to the window/display height.
- VCTXVIDLIMIT (num) vid stack context size.
- WORLDID (num) *vid* of a special object that refers to the outmost coordinate system.
- BADID (num) functions returning a vid can also return BADID if the function failed for some reason.
- CLOCK (num) built-in clock rate (default, 25hz).
- JOYSTICKS (num) number of detected (and opened) joysticks.
- LEDCONTROLLERS (num) number of detected LED controllers.
- THEME_RESOURCE (num) mask value for specifying themepath/themename/ as a search-path for resources.
- SHARED_RESOURCE (num) mask value for specifying resourcepath as a search-path for resources.
- API_VERSION_MAJOR (num)
- API_VERSION_MINOR (num)
- LAUNCH_EXTERNAL (num) mask value for specifying that the target should be launched in external mode.
- LAUNCH_INTERNAL (num) mask value for specifying that the target should be launched in internal mode.
- MASK_ORIENTATION, MASK_OPACITY, MASK_POSITION, MASK_SCALE, MASK_UNPICKABLE (num)
- THEMENAME (text) name of the currently loaded theme.
- RESOURCEPATH, THEMEPATH, BINPATH, LIBPATH (text)
- INTERNALMODE (text) estimated level of internal launch support.
- NOW (num) used to specify that a transformation (move, scale, blend, ...) should happen immediately.

Planned Features

There's no strict schedule for these features currently. A lot will depend on community interest but the main priority is stability for the currently available API / feature-set.

Input

- (RGB+intensity) LED controllers (don't have access to any currently).
- scripts to convert from keyconfig scripts to mame etc.
- Manymice support (includes the internal target launcher).
- Wiimote support.
- Touchscreen support.

System

- Replace IPC- esque commands for event- serialization using ProtocolBuffers.
- Yield alternative for SDL_delay in framequeue on EAGAIN.
- Android port.
- Support- library for cooperative internal-launch for more efficient / less aggressive internal-launch, pause- functions etc. with possibly patches for mplayer, mame, mess, ...
- Collision- / Intersection- tests for VIDs, basic 2d physics?

OSX

- Seems like a priority inversion problem when going into fullscreen / full-windowed mode, makes internal launch / movies slow.
- Full- screen options don't work at all on Lion (SDL- problem as well).
- Focus isn't properly moved between main window / external target and back (should use launch- hidden).
- Internal launch- mode drops focus.
- Overall "design" does not fit well with 'Bundle' style.

Linux

- Build-system cleanup and package maintenance (rpm, deb, ...)

Windows

- Internal launch-mode missing.

Video

- Multi-frame object (animations, important one).
- Direct mng support as animation format alternative to multiple single image files.
- Proper orientation (3 axis, quat-¿matrix) with interpolation.
- Two-sided objects (requires orientation + multiframe + updated picking).
- More advanced frameserver support (webcam, ...).
- Stream-export vid+aid (or world) to ffmpeg as external encoder / frameserver.
- *halign* (centered, left, right) and *valign* (top, middle, bottom with multiline-multisize) text support.

Audio

- slightly more format support, perhaps move to an audio_frameserver.
- mpd client support.
- speex chat.
- scriptable synthesis.

Database

- Target filters
- Game-/Target- based key/value tagging (current is only theme based)
- Launch history.
- Datamodel, support-scripts for movies / tv-series.

LUA

- Pass command-line arguments to theme() function.
- Better scheduling / preemption of lua- execution. Event multiplexing.
- Working sandboxing, currently there's directory traversal in resource calls.
- Restricted online features (gamelists, highscores, playcounts, ...)