Visual Library for Game Maker Documentation
v0.1 (First Testing Release)

December 28, 2009

The Visual Library for Game Maker is an extension written in C++ using the OpenCV libraries. The purpose of this extension is to provide Game Maker developers with a set of functions to handle and interpret visual data from a webcam.

Contents

1 General Functions

1.1 vl_init

vl_init(x, y, width, height, flip, file, background, path)

Initialize the Visual Library for use. ***This function is necessary to use all the features of the Visual Library.***

Parameters

x - X position to draw the webcam image.

y - Y position to draw the webcam image.

width - Width of the webcam image.

height - Height of the webcam image.

flip - If set to true, the webcam image is flipped horizontally.

background - Background to store the webcam image into.

path - Location of Visual Library dll.

Return Value 1 on success, 0 on failure.

1.2 vl_update

vl_update()

Processes the current frame and updates necessary information.

1.3 vl_set_mode

vl_set_mode(mode)

Set the processing mode of the Visual Library.

Parameters

mode - Sets the mode and enables functions according to predefined variables:

- VL_MODE_MOTION - Detect motion in the webcam frames (See Motion Detection 2).
- VL_MODE_COLOR - Detect color in the current webcam frame (See Color Detection 4).
- VL_MODE_TRACK - Track points added (See Point Tracking 3).

Return Value 1 on success, 0 on failure.

1.4 vl_draw

vl_draw()

Reloads the webcam image and draws it with the position and size according to the parameters passed to vl_init 1.1.

Parameters None

Return Value None

1.5 vl_flip

vl_flip()

Flips the webcam image horizontally.

Parameters None

Return Value None

1.6 vl_unload

vl_unload()

Frees the Visual Library and cleans up.

Parameters None

Return Value None

2 Motion Detection

For the function in this section to work properly, the mode needs to be set to VL_MODE_MOTION. See vl_set_mode 1.3.

2.1 vl_check_motion

vl_check_motion(x, y)

Detects motion that occurs in a single pixel of the webcam image.

Parameters

x - The x position on screen to detect motion.

y - The y position on screen to detect motion.

The coordinates represent the position of the desired pixel of the webcam image as it is drawn on screen, so the position and size of the webcam image will affect the pixel that is selected.

Return Value A value representing the magnitude of the motion. Returns -1 if the coordinates are out of range.

3 Point Tracking

For the functions in this section to work properly, the mode needs to be set to VL_MODE_TRACK. See vl_set_mode 1.3.

3.1 vl_point_add

vl_point_add(x, y)

Adds a point to be tracked.

Parameters

x - Initial x position of tracking point.

y - Initial y position of tracking point.

Return Value The point identifier used in the other point tracking functions.

3.2 vl_point_get_x

vl_point_get_x(identifier)

Get the current x position of a tracking point.

Parameters

identifier - The tracking point identifier (obtained from vl_point_add 3.1).

Return Value X position of the point.

3.3 vl_point_get_y

vl_point_get_y(identifier)

Get the current y position of a tracking point.

Parameters

identifier - The tracking point identifier (obtained from vl_point_add 3.1).

Return Value Y position of the point.

3.4 vl_point_set

vl_point_set(x, y)

Sets the position of a tracking point.

Parameters

x - X position of tracking point.

y - Y position of tracking point.

Return Value 0 on success, -1 if the coordinates are out of bounds.

4 Color Detection

For the function in this section to work properly, the mode needs to be set to VL_MODE_COLOR. See vl_set_mode 1.3.

4.1 vl_check_color

vl_check_color(x, y, color)

Compares the color of the pixel at the given coordinates to a color value by finding the vector distance between the RGB components of each color.

Parameters

x - The x position on screen to detect color.

y - The y position on screen to detect color.

color - Color value to compare with the webcam pixel.

The coordinates represent the position of the desired pixel of the webcam image as it is drawn on screen, so the position and size of the webcam image will affect the pixel that is selected.

Return Value A value from 0–255 representing the vector distance of the webcam's pixel color to the provided color. Returns -1 if the coordinates are out of range.