SDL2 - Get started

Graphics programming

Setup SDL2

Installation

- First, download the installation script
- This installation works for **Mac OS X** and **Linux/Debian** only. If you need SDL2 for an other platform, please visit SDL installation guide, along with SDL2_ttf.

To install `SDL2` with the provided script, just open a terminal and go to the directory where the script is, and execute it with root privileges:

```
alex@ubuntu:/tmp/SDL2$ ls
install_SDL2.sh
alex@ubuntu:/tmp/SDL2$ chmod 755 install_SDL2.sh
alex@ubuntu:/tmp/SDL2$ sudo ./install_SDL2.sh
Password:
[...]
All set !
alex@ubuntu:/tmp/SDL2$
```

Note: You can also launch the installation in one command like follows:

```
alex@ubuntu:/tmp/SDL2$ curl https://s3.amazonaws.com/intranet-projects-files/holberton
school-low_level_programming/graphics_programming/install_SDL2.sh | sudo bash
Password:
[...]
All set !
alex@ubuntu:/tmp/SDL2$
```

And that's it !

At the end you should have the message `"All set !"`. If you don't, it means that an error occurred, so check the output of the script to know why.. As specified after the installation,if you want to use SDL2 library don't forget to compile your sources with **`sdl2-config --cflags`** and link your compiled sources with **`sdl2-config --libs`**

First program with SDL2

The following code will initialize an SDL2 instance. (SDL_Init)

Then, we can create a window, and a renderer.

The renderer is associated to the window and will be used to draw stuff on this window.

```c
#include <SDL2/SDL.h>

int main(void)
{
    SDL_Window *window;
    SDL_Renderer *renderer;

    /* Initialie SDL */
    if (SDL_Init(SDL_INIT_VIDEO) != 0)
    {
        fprintf(stderr, "Unable to initialize SDL:  %s\n", SDL_GetError());
        return (1);
    }
    /* Create a new Window instance */
    window = SDL_CreateWindow("SDL2 \\o/", SDL_WINDOWPOS_CENTERED,
        SDL_WINDOWPOS_CENTERED, 1260, 720, 0);
    if (window == NULL)
    {
        fprintf(stderr, "SDL_CreateWindow Error: %s\n", SDL_GetError());
        SDL_Quit();
        return (1);
    }
    /* Create a new Renderer instance linked to the Window */
    renderer = SDL_CreateRenderer(window, -1,
        SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC);
    if (renderer == NULL)
    {
        SDL_DestroyWindow(window);
        fprintf(stderr, "SDL_CreateRenderer Error: %s\n", SDL_GetError());
        SDL_Quit();
        return (1);
    }
    return (0);
}
```

Let’s declare a structure that will hold the addresses of the window and the renderer.

```h
#ifndef _DEMO_H_
#define _DEMO_H_

#include <SDL2/SDL.h>

typedef struct SDL_Instance
{
    SDL_Window *window;
    SDL_Renderer *renderer;
} SDL_Instance;

int init_instance(SDL_Instance *);

#endif
```

Then, let’s put the initialisation in a separated function.

```c
int main(void)
{
    SDL_Instance instance;

    if (init_instance(&instance) != 0)
        return (1);
    return (0);
}
```

```c
int init_instance(SDL_Instance *instance)
{
    /* Initialie SDL */
    if (SDL_Init(SDL_INIT_VIDEO) != 0)
    {
        fprintf(stderr, "Unable to initialize SDL:  %s\n", SDL_GetError());
        return (1);
    }
    /* Create a new Window instance */
    instance->window = SDL_CreateWindow("SDL2 \\o/", SDL_WINDOWPOS_CENTERED,
        SDL_WINDOWPOS_CENTERED, 1260, 720, 0);
    if (instance->window == NULL)
    {
        fprintf(stderr, "SDL_CreateWindow Error: %s\n", SDL_GetError());
        SDL_Quit();
        return (1);
    }
    /* Create a new Renderer instance linked to the Window */
    instance->renderer = SDL_CreateRenderer(instance->window, -1,
        SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC);
    if (instance->renderer == NULL)
    {
        SDL_DestroyWindow(instance->window);
        fprintf(stderr, "SDL_CreateRenderer Error: %s\n", SDL_GetError());
        SDL_Quit();
        return (1);
    }
    return (0);
}
```

We can now start an infinite loop.

On each loop, we're gonna:

- Clear the renderer
- Draw stuff on the renderer
- Flush the renderer

Each loop represents a frame

```c
int main(void)
{
    SDL_Instance instance;

    if (init_instance(&instance) != 0)
        return (1);
    /*
     * C will always be awesome
     * This is an infinite loop
     */
    while ("C is awesome")
    {
        SDL_SetRenderDrawColor(instance.renderer, 0, 0, 0, 0);
        SDL_RenderClear(instance.renderer);
        /*
         * Draw some stuff here
         */
        SDL_RenderPresent(instance.renderer);
    }
    return (0);
}
```

Example

In this quick example, we call a function (draw_stuff), that will just draw a line.

```
while ("C is awesome")
{
    SDL_SetRenderDrawColor(instance.renderer, 0, 0, 0, 0);
    SDL_RenderClear(instance.renderer);
    draw_stuff(instance);
    SDL_RenderPresent(instance.renderer);
}
```

```
void draw_stuff(SDL_Instance instance)
{
    SDL_SetRenderDrawColor(instance.renderer, 0xFF, 0xFF, 0xFF, 0xFF);
    SDL_RenderDrawLine(instance.renderer, 10, 10, 100, 100);
}
```

Events

SDL allows you to retrieve some events like:

- A key was pressed on the keyboard
- A key was released
- The mouse moved
- A button of the mouse was pressed/released
- The ‘exit’ button of the window was pressed
- ...

Retrieve events

You can retrieve events on each frame.

Events are pushed in a queue by SDL, so it’s a good practice to iterate on the function ‘SDL_PollEvent’ until the queue is empty:

```c
while ("C is awesome")
{
    SDL_SetRenderDrawColor(instance.renderer, 0, 0, 0, 0);
    SDL_RenderClear(instance.renderer);
    if (poll_events() == 1)
        break;
    draw_stuff(instance);
    SDL_RenderPresent(instance.renderer);
}
```

```c
int poll_events()
{
    SDL_Event event;
    SDL_KeyboardEvent key;

    while (SDL_PollEvent(&event))
    {
        switch (event.type)
        {
        case SDL_QUIT:
            return (1);
        case SDL_KEYDOWN:
            key = event.key;
            /* If 'ESCAPE' is pressed */
            if (key.keysym.scancode == 0x29)
                return (1);
            break;
        }
    }
    return (0);
}
```

Quit SDL

At the end of your program (when you break your infinite loop),

Don’t forget to release all the stuff you initialized with SDL.

In our example, we release the window and the renderer

But you can also have textures, surfaces, ...

```c
int main(void)
{
    SDL_Instance instance;

    if (init_instance(&instance) != 0)
        return (1);
    /*
     * C will always be awesome
     * This is an infinite loop
     */
    while ("C is awesome")
    {
        SDL_SetRenderDrawColor(instance.renderer, 0, 0, 0, 0);
        SDL_RenderClear(instance.renderer);
        if (poll_events() == 1)
            break;
        draw_stuff(instance);
        SDL_RenderPresent(instance.renderer);
    }
    SDL_DestroyRenderer(instance.renderer);
    SDL_DestroyWindow(instance.window);
    SDL_Quit();
    return (0);
}
```

Read the documentation :)