SDL In Practice

Keep It Small and Simple

SDL_image

Setup Environment

&

Load Image By SDL_image

inviassassin@gmail.com

Setup Environment

Install

```
sudo apt-get install libsdl2-image-dev
```

Load Image By SDL_image

hello_sdl_image.cpp

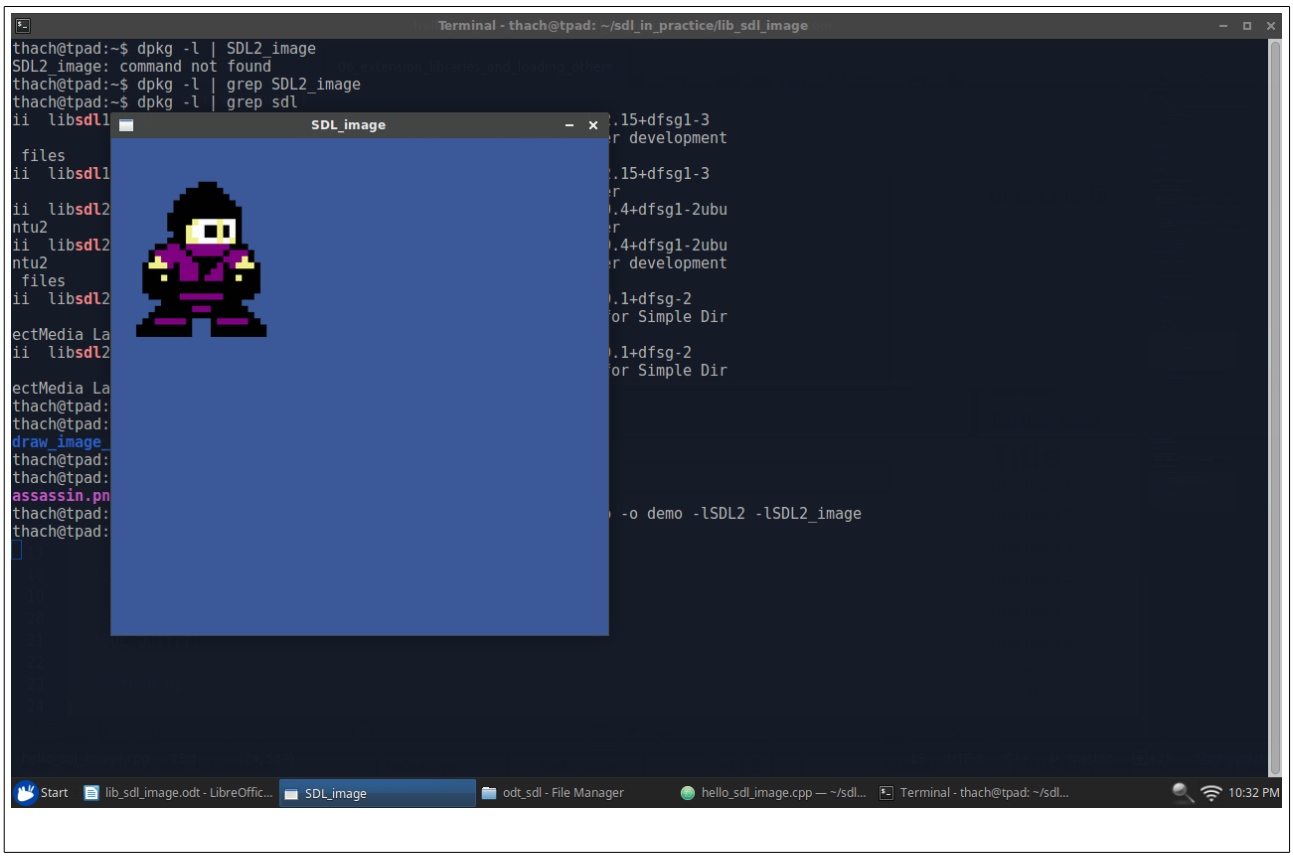
```
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>

int main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Window *pWindow = SDL_CreateWindow("SDL_image", 100, 100, 500, 500, 0);
    SDL_Surface *pWindowSurface = SDL_GetWindowSurface(pWindow);

    IMG_Init(IMG_INIT_PNG);
    SDL_Surface *pImageSurface = IMG_Load("assassin.png");

    SDL_FillRect(pWindowSurface, NULL, 0x3b5999);
    SDL_BlitSurface(pImageSurface, NULL, pWindowSurface, NULL);
    SDL_UpdateWindowSurface(pWindow);

    SDL_Delay(5000);

    SDL_FreeSurface(pImageSurface);
    SDL_DestroyWindow(pWindow);
    IMG_Quit();
    SDL_Quit();

    return 0;
}
```

Compile

```
g++ hello_sdl_image.cpp -o demo -lSDL2 -lSDL2_image
```

Run

```
./demo
```



SDL_image In Theory

SDL image is a library used to load images of various formats to SDL surfaces.

Support image formats: ico, bmp, png, xpm, lbm, pcx, gif, jpeg, pnm, tga, tiff, xv.

The flow to use SDL_image

- Init SDL_image

- Load image to surface

- Render surface

- Free the surface when it is no longer needed

- Quit SDL_image

Init SDL_image

In this step, we have to register which image format we will use.

```
int IMG_Init(int flags)
```

For example, we will use png format

```
IMG_Init(IMG_INIT_PNG)
```

If we will use png and jpg formats

```
IMG_Init(IMG_INIT_JPG | IMG_INIT_PNG)
```

Load Image To Surface

```
SDL_Surface *pImageSurface = IMG_Load("assassin.png");
```

This function can load all supported image files. We have to call SDL_FreeSurface to release the returned surface pointer. If the image format supports a transparent pixel, the color of this pixel will be the color of the surface.

Render Surface

```
SDL_BlitSurface(pImageSurface, NULL, pWindowSurface, NULL);
```

Free Surface

```
SDL_FreeSurface(pImageSurface);
```

Quit SDL_image

```
SDL_Quit()
```

This function is used to unload all dynamically loaded image: clean up all dynamically loaded library handles and release memory.