The Simple DirectMedia Layer, (SDL) 2.0 Library
A Brief Introduction

September 4, 2017

Brian A. Malloy

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries

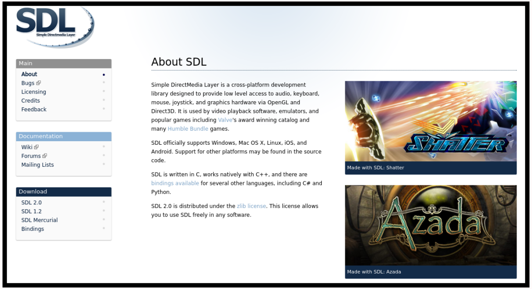
Slide 1 of 16

Go Back

Full Screen

Quit

1. Resources



About SDL

Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D. It is used by video playback software, emulators, and popular games including Valve's award winning catalog and many Humble Bundle games.

SDL officially supports Windows, Mac OS X, Linux, iOS, and Android. Support for other platforms may be found in the source code.

SDL is written in C, works natively with C++, and there are bindings available for several other languages, including C# and Python.

SDL 2.0 is distributed under the zlib license. This license allows you to use SDL freely in any software.

CLEMSON
UNIVERSITY

Resources

Why Use SDL?

What's new in 2.0?

Game Loop

General Facts . . .

External Libraries

◀◀ ▶▶

◀ ▶

Slide 2 of 16

Go Back

Full Screen

Quit

2. Why Use SDL?

- Compiles and runs on Linux, Windows, OSX
- Large user community
- Lightweight

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries
◀◀ ▶▶
◀ ▶
Slide 3 of 16
Go Back
Full Screen
Quit

3. What’s new in 2.0?

- 3D accelerated, texture-based rendering
- Hardware-accelerated 2D graphics
- Multiple Windows
- Audio 7.1
- Audio capture support
- And more . . .

Resources

Why Use SDL?

What’s new in 2.0?

Game Loop

General Facts . . .

External Libraries

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Quit

Nvidia's GTX Titan Graphics Card

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries
◀◀ ▶▶
◀ ▶
Slide 5 of 16
Go Back
Full Screen
Quit

4. Game Loop

- The game loop drives the video game.
- Our game loop will utilize the philosophy "run as fast as you can."

```
while ( true ) {
  processInput();
  update();
  render();
}
```

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries
◀◀ ▶▶
◀ ▶
Slide 6 of 16
Go Back
Full Screen
Quit

4.1. Alternative Game Loop

- The game loop drives the video game.
- Our game loop will utilize the philosophy "run as fast as you can."

```
while ( true ) {
  processInput();
  update();
  render();
  sleep( ms );
}
```

CLEMSON UNIVERSITY

Resources

Why Use SDL?

What's new in 2.0?

Game Loop

General Facts . . .

External Libraries

◀◀ ▶▶

◀ ▶

Slide 7 of 16

Go Back

Full Screen

Quit

5. General Facts About SDL 2.0:

- No backward compatibility with 1.2
- No SDL_main
- Multiple windows:

```
SDL_Window *window =
  SDL_CreateWindow("Game Window",
                    posX,
                    posY,
                    640, 480,
                    <flags>);
```

- SDL_Window is a struct with a specified position, dimension, and flags.

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries

Go Back
Full Screen
Quit

5.1. Some Window <flags>

- Must set posX & posY to SDL_WINDOWPOS_UNDEFINED for:
  - SDL_WINDOW_FULLSCREEN
- SDL_WINDOW_OPENGL
- Window can be centered with: SDL_WINDOWPOS_CENTERED

```
SDL_Window* window =
    SDL_CreateWindow( "Hello World", SDL_WINDOWPOS_CENTERED,
       SDL_WINDOWPOS_CENTERED, WIDTH, HEIGHT, SDL_WINDOW_SHOWN );
if( window == NULL ) {
   throw (std::string("Couldn't make a window: ")+SDL_GetError());
 }
```

CLEMSON UNIVERSITY
Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries
◀◀ ▶▶
◀ ▶
Slide 9 of 16
Go Back
Full Screen
Quit

5.2. Surface v Texture

- Surface: always in RAM and always operated on by CPU
- Texture: SDL2 has a new rendering API: moves software rendering into video RAM and onto the GPU.
  - If possible: use OpenGL or Direct3D
  - Faster blits
  - Scaling

5.3. Renderer

-
  ```
  SDL_Renderer *renderer =
     SDL_CreateRenderer(window, -1, 0);
  ```
- SDL_Renderer is a struct that handles all drawing.
- The SDL_Renderer draws textures.
- It's attached to an SDL_Window and it can only render in that window.
- Keeps track of settings attached to the window, such as the window clear color.
- Hides details of how render occurs: OpenGL, Direct3D, or software surfaces
- If want to reduce tearing, instead of 0 for third parameter, use: SDL_RENDERER_PRESENTVSYNC

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries
◀◀ ▶▶
◀ ▶
Slide 11 of 16
Go Back
Full Screen
Quit

5.4. Clearing the Window/Screen

```
SDL_SetRenderDrawColor(renderer,
   0, 0, 0, 255);
SDL_RenderClear(renderer);
SDL_RenderPresent(renderer);
```

- Black: (0, 0, 0)
- Alpha channel: 255 is opaque
- SDL_RenderClear() – paints window
- SDL_RenderPresent() – similar to SDL_Flip()

5.5. blitting Surfaces in SDL1.2

- In SDL 1.2 we loaded sprites from disk into SDL_Surfaces
- Surfaces were loaded once, and game loop went like this:
  - blit surface
  - SDL_Flip
  - repeat

5.6. rendering Textures from Surfaces

- In SDL2, we can load surfaces and then convert them to textures:

```
SDL_Surface *surface = IMG_Load("hello.bmp");
SDL_Texture *texture =
  SDL_CreateTextureFromSurface(renderer, surface);
SDL_FreeSurface(surface);  // throw surface away

Loop:
  SDL_RenderClear(renderer);
  SDL_RenderCopy(renderer, texture, NULL, NULL);
  SDL_RenderPresent(renderer);
```

Resources
Why Use SDL?
What's new in 2.0?
Game Loop
General Facts . . .
External Libraries
◀◀ ▶▶
◀ ▶
Slide 14 of 16
Go Back
Full Screen
Quit

5.7. rendering Textures Directly

```
SDL_Texture *texture =
  IMG_LoadTexture(renderer, "hello.bmp");

Loop:
  SDL_RenderClear(renderer);
  SDL_RenderCopy(renderer, texture, NULL, NULL);
  SDL_RenderPresent(renderer);
```

Of course, we have memory leaks if we don't:

```
SDL_DestroyTexture(texture);
SDL_DestroyRenderer(renderer);
SDL_DestroyWindow(win);
```

Resources

Why Use SDL?

What's new in 2.0?

Game Loop

General Facts . . .

External Libraries

◀◀ ▶▶

◀ ▶

Slide 15 of 16

Go Back

Full Screen

Quit

6. External Libraries

- The libraries we will use:
  - SDL_image: support for bmp, gif, png, jpg, tga, pcx, . . .
  - SDL_mixer: audio mixer library, supports mp3, midi, ogg.
  - SDL_ttf: support fo TrueType fonts
  - expat: SAX XML parser
- Additional libraries that we will not use:
  - SDL_rtf: for Rich Text Format
  - SDL_net: cross-platform networking library