

Creating a Window with SDL

Note: This tutorial assumes that you already know how to set up SDL in your IDE.

Creating a Window

Creating a window with SDL involves the following steps:

- Initializing the video component of SDL.
- Creating the window.
- Setting the window's title (*optional*).
- Shutting down SDL.

Initializing the video component of SDL.

To initialize SDL's video component you call `SDL_Init()` and pass it `SDL_INIT_VIDEO`.

Creating the window.

You create a window with a call to `SDL_SetVideoMode()`, which takes four parameters.

The first parameter is the width of the window. The second parameter is the height of the window.

The third parameter is the bits per pixel you want to use for your game. Passing 0 for this value is the easiest thing to do, as it tells SDL to use the current display settings.

The fourth parameter is a flag variable. The two flags we'll be using are `SDL_HWSURFACE` and `SDL_DOUBLEBUF`. We can combine flags with the OR operator like so:

```
SDL_HWSURFACE | SDL_DOUBLEBUF.
```

`SDL_SetVideoMode()` doesn't just create a window. It also creates an area in memory called the "screen buffer" where we can draw to. This buffer is what gets displayed on the screen. The flag `SDL_HWSURFACE` specifies that we want the buffer created in video memory.

The `SDL_DOUBLEBUF` flag specifies that we want to use two buffers. One buffer is the front buffer. This buffer is what is being displayed. The other buffer is the back buffer. This buffer is what we draw to. When we're done drawing, we swap the front buffer and

the back buffer so the stuff we drew on the back buffer gets displayed. This technique is called double buffering and it's used to speed up the rendering process.

`SDL_SetVideoMode()` returns the screen buffer as an `SDL_Surface`. The `SDL_Surface` structure represents an area in memory that can store graphical information. When we start drawing things, we'll draw them to the surface returned by `SDL_SetVideoMode()`.

Setting the window's title.

We can set the title of our window with a call to `SDL_WM_SetCaption()`, which takes two parameters. The first is the title we want for our window. The second parameter is for specifying a custom icon in the window's title bar. We'll just pass in 0 to use the default icon.

Shutting down SDL. We call the `SDL_Quit()` function to shut down SDL. Note that this will free the surface returned by `SDL_SetVideoMode()`, so we should never free that surface ourselves.

Here's some code to create a window. If you run it, you'll only see the window for a brief moment. This is because we call `SDL_Quit()` immediately after creating the window. In the next section, we'll create a loop that runs until the user closes the window.

```
#include "SDL.h"

const int WINDOW_WIDTH = 640;
const int WINDOW_HEIGHT = 480;
const char* WINDOW_TITLE = "SDL Start";

int main(int argc, char **argv)
{
    SDL_Init( SDL_INIT_VIDEO );

    SDL_Surface* screen = SDL_SetVideoMode( WINDOW_WIDTH,
WINDOW_HEIGHT, 0,
        SDL_HWSURFACE | SDL_DOUBLEBUF );
    SDL_WM_SetCaption( WINDOW_TITLE, 0 );

    SDL_Quit();

    return 0;
}
```

Keeping the Window Open

To keep our window open, we'll create a while loop like the following:

```
bool gameRunning = true;
while (gameRunning)
{
}
```

When we want to exit the loop, we'll set `gameRunning` to false. We'll do this when the user chooses to close the window.

To detect that the user wants to close the window, we call `SDL_PollEvent()`.

`SDL_PollEvent()` takes an `SDL_Event` structure as a parameter and fills it with information about events that are happening. It returns 0 if there are no events, so we should check its return value to make sure there's an event to deal with.

The `SDL_Event` structure has all kinds of variables, but the only one we're considered with right now is `type`. If this variable is equal to `SDL_QUIT`, then we know that the user wants to close the window (ie. the user has pressed the X in the top right of the window).

Here is the code for creating a window and keeping it open. I've put the changes in bold.

```
#include "SDL.h"

const int WINDOW_WIDTH = 640;
const int WINDOW_HEIGHT = 480;
const char* WINDOW_TITLE = "SDL Start";

int main(int argc, char **argv)
{
    SDL_Init( SDL_INIT_VIDEO );

    SDL_Surface* screen = SDL_SetVideoMode( WINDOW_WIDTH,
WINDOW_HEIGHT, 0,
        SDL_HWSURFACE | SDL_DOUBLEBUF );
    SDL_WM_SetCaption( WINDOW_TITLE, 0 );

    SDL_Event event;
    bool gameRunning = true;
    while (gameRunning)
    {
        if (SDL_PollEvent(&event))
        {
            if (event.type == SDL_QUIT)
            {
                gameRunning = false;
            }
        }
    }
}
```

```
    }  
}  
SDL_Quit();  
return 0;  
}
```

That's all it takes to create a window with SDL.

© Copyright Aaron Cox 2004-2005, All Rights Reserved.