Lazy Foo' Productions

News FAQs Games Tutorials Articles Contact Donations

Joysticks

Last Updated 11/15/07

Here we'll learn to use get input from a joystick/gamepad. This program is basically the motion tutorial, only this time a joystick will move around the dot instead of the keyboard.

```
//The joystick that will be used
SDL_Joystick *stick = NULL;
```

Joysticks have their own data type in SDL which is SDL_Joystick.

In this program we declare our joystick as a global variable.

```
//The dot
class Dot
{
    private:
    //The offsets of the dot
    int x, y;

    //The velocity of the dot
    int xVel, yVel;

    public:
    //Initializes
    Dot();

    //Handles joystick
    void handle_input();

    //Moves the dot
    void move();

    //Shows the dot
    void show();
};
```

As you can see, the Dot class has pretty much stayed the same. The only thing that has changed is how we handle the input.

```
bool init()
{
    //Initialize all SDL subsystems
    if( SDL_Init( SDL_INIT_EVERYTHING ) == -1 )
    {
        return false;
    }

    //Set up the screen
    screen = SDL_SetVideoMode( SCREEN_WIDTH, SCREEN_HEIGHT, SCREEN_BPP, SDL_SWSURFACE

    //If there was an error in setting up the screen
    if( screen == NULL )
    {
        return false;
    }

    //Check if there's any joysticks
    if( SDL_NumJoysticks() < 1 )
    {
        return false;
    }

    //Open the joystick
    stick = SDL_JoystickOpen( 0 );

    //If there's a problem opening the joystick
    if( stick == NULL )
    {
        return false;
    }

    //Set the window caption
    SDL_WM_SetCaption( "Move the Dot", NULL );

    //If everything initialized fine
    return true;
}
```

A key difference between using keys for input and using joysticks is that joysticks have to be initialized.

In the initialization function we use SDL_NumJoysticks() to check how many joysticks are plugged in. If at least 1 joystick is plugged in, we open the first joystick available using SDL_JoystickOpen(). The first available joystick is joystick 0 since in programming we always start counting at 0. When there's problem in opening the joystick, SDL_JoystickOpen() returns NULL.

```
void clean_up()
{
    //Free the surface
    SDL_FreeSurface( dot );

    //Close the joystick
    SDL_JoystickClose( stick );

    //Quit SDL
    SDL_Quit();
}
```

In the clean up function, we have to call SDL_JoystickClose() to close the joystick that was opened.

```
void Dot::handle_input()
{
    //If a axis was changed
    if( event.type == SDL_JOYAXISMOTION )
    {
        //If joystick 0 has moved
        if( event.jaxis.which == 0 )
        {
            //If the X axis changed
            if( event.jaxis.axis == 0 )
            {
                //If the X axis is neutral
                if( ( event.jaxis.value > -8000 ) && ( event.jaxis.value < 8000 ) )
                {
                    xVel = 0;
                }
                //If not
                else
                {
                    //Adjust the velocity
                    if( event.jaxis.value < 0 )
                    {
                        xVel = -DOT_WIDTH / 2;
                    }
                    else
                    {
                        xVel = DOT_WIDTH / 2;
                    }
                }
            }
```

When a joystick moves, a SDL_JOYAXISMOTION occurs.

First we check if the joystick that has moved is joystick 0. It's kind of pointless, since the only initialized joystick is joystick 0, but its a good habit to check no matter what.

Then we check which axis it has moved on. On most modern joysticks, the X axis is 0, and the Y axis is 1.

After that we check if the joystick X value is between -8000, and 8000. If it is, it's neutral and the dot stays still.

You may be thinking "how the hell is such a large range considered neutral?". The thing is a joystick's axis have a range of -32768 to 32767. You could have the joystick at 0 and if you sneezed on it, it would be at like 200.

If the joystick is not in the neutral range, we set the X velocity accordingly.

```
            //If the Y axis changed
            else if( event.jaxis.axis == 1 )
            {
                //If the Y axis is neutral
                if( ( event.jaxis.value > -8000 ) && ( event.jaxis.value < 8000 ) )
                {
                    yVel = 0;
                }
                //If not
                else
                {
                    //Adjust the velocity
                    if( event.jaxis.value < 0 )
                    {
                        yVel = -DOT_HEIGHT / 2;
                    }
                    else
                    {
                        yVel = DOT_HEIGHT / 2;
                    }
                }
            }
        }
    }
}
```

Then pretty much the same thing is done to the Y axis.

Handling SDL_JoyAxisEvent is the hardest event to handle when dealing with joysticks. Handling other events like SDL_JoyBallEvent, SDL_JoyHatEvent, and SDL_JoyButtonEvent should be easy to figure out with a quick look at the SDL API documentation.

Download the media and source code for this tutorial here.

News FAQs Games Tutorials Articles Contact Donations