Direct Input Interface Library (DIIL)
Version 2.0

Bryan C Warren
Modified by Axel Koch
CSS 499: Final Project Report
Computing and Software Systems
University of Washington, Bothell
March 29, 2003

Version History.

Version	Date	Description
Original Release	June 14, 2002	Support for force feedback joystick.
V2.0	March 29, 2003	Added support for force feedback game pad. Modified documentation and provided DirInputTester utility.

Introduction.

The Direct Input Interface Library is designed to allow programmers to utilize a force feedback joystick or game pad without knowing how to program DirectX. It contains the necessary functions for accessing the positions of the various buttons, switches, and levers as well as communicating with the servo motors to provide constant, directional and periodic, vibrating forces.

Requirements.

Files:

- DirInput.h
- DirInput.lib

Software:

- Microsoft Visual Studios (recommended)
- DirectX 8 SDK

Assumptions.

1. Your gaming device is force feedback capable.
2. Your gaming device is the first device found.
3. Your gaming device has at least two axes.
4. The standard constant force and the simple, square-shaped periodic force are all you need.
5. Devices cannot be changed during run time, i.e. user must restart program after device change.

Using the Direct Input Interface Library.

The following instructions are for use with Microsoft Visual Studios 6. Other development software and other versions of Visual Studios may vary.

1. From the Project menu, select Settings (Alt+F7).
2. Click the Link tab.
3. Add the following libraries to the beginning of the Object/library modules list:

    a. dxguid.lib
    b. winmm.lib
    c. dxerr8.lib
    d. dinput8.lib
    e. dirinput.lib

4. Click OK.
5. From the Tools menu, select Options.
6. Click the Directories tab.
7. Choose to Show directories for Include files.
8. Click at the bottom of the list and type the path for the include files from DirectX 8 SDK (e.g. C:\DXSDK\INCLUDE).
9. Click and drag the new item to the top of the list.
10. Choose to Show directories for Library files.
11. Click at the bottom of the list and type the path for the library files fromDirectX 8 SDK (e.g. C:\DXSDK\LIB).
12. Click and drag the new item to the top of the list.
13. Click OK.
14. Add the following lines of code to the beginning of your program file:

    a. #include <windows.h>
    b. #include “dirinput.h”

15. Place the following files in your working directory:

    a. DirInput.h
    b. DirInput.lib

Available Functions.

BOOL ***CaptureFFdevice*** *(HWND, INT lowLimit, INT highLimit)*
Initialization of DirectInput device and feedback objects. Sets up the interface variables to DirectX and must be called first for other functions to work. LowLimit and highLimit allow the programmer to specify the numeric range of values returned by the DIIL. Default values are:

- lowLimit: -1000
- highLimit: +1000

DeviceType ***GetDeviceType*** *( )*
Returns the type of device that is currently connected. Return value is of type enum DeviceType and can be either JOYSTICK, GAMEPAD, NO_DEVICE_FOUND or UNKNOWN_DEVICE. This function should only be called after '*CaptureFFdevice*' has been called. The default return value is NO_DEVICE_FOUND.

void ***ReleaseFFdevice*** *( )*
Allows Direct Input objects to be deleted and released when no longer needed.

POINT ***GetXYStatus*** *( )*
Polls Device for joystick position. Corresponds to left joystick on game pad and the main axis on the joystick device (see Figure 1). Parameter returned is of type POINT, containing X & Y values, in the range of Low Limit to High Limit.

POINT ***GetABStatus*** *( )*
Polls device for secondary axis positions. Corresponds to right joystick on game pad and the rudder (A axis) and throttle (B axis) on the joystick device (see Figure 1). Parameter returned is of type POINT, containing A & B values, in the range of Low Limit to High Limit.

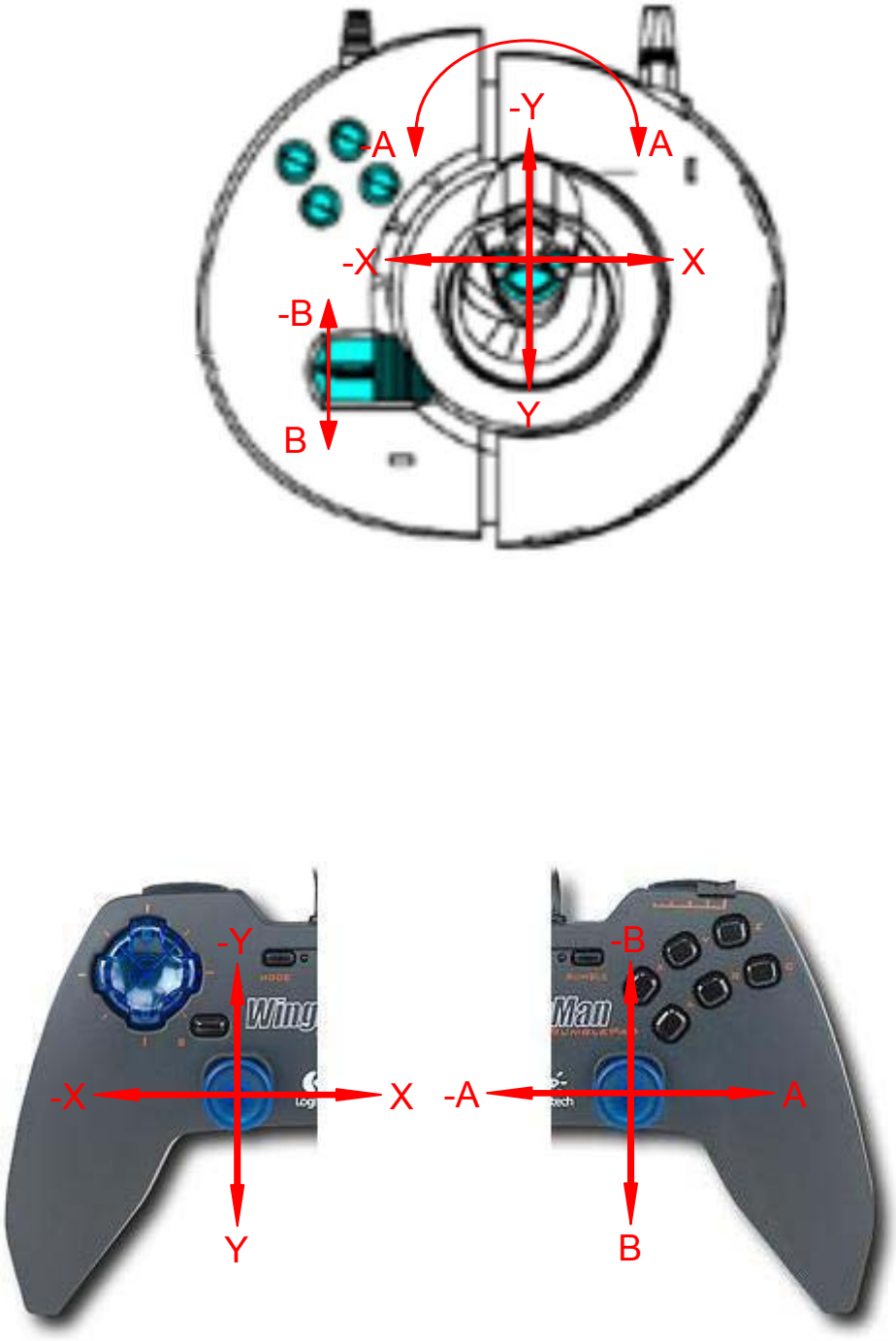

Figure 1: Main axis for force feedback joystick and game pad

LONG ***GetSliderStatus*** *( )*
Only applicable to game pad! Always returns 0 for joystick. Polls game pad for slider position. Parameter returned is of type LONG, in the range of Low Limit to High Limit.

BOOL ***ButtonPressed*** *(INT buttonNumber)*
Polls device to see if button is pressed. True indicates button is pressed. Button Number passed in by user, from range of 1 to 32, default is 1.

POVstatus ***GetPOVswitchStatus*** *( )*
Polls device for point of view (POV) switch position. Parameter returned is of type enum POVstatus. Values can be NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH, SOUTHWEST, WEST, and NORTHWEST. Value NILL indicates center position.

HRESULT ***SetupPeriodicForce*** *(double period)*
Establishes parameters for a pulsating force according to a defined wave pattern (i.e. shaking effect). Creates an instance of a periodic effect which can then be started and stopped as necessary. Input parameter is the period of the wave. Value defaults to 1/20 second.

HRESULT ***AlterPeriodicForce*** *(INT magnitude, double period,*
INT offset, INT direction)
Allows the periodic force effect to be changed without recreating the effect. Input parameters are magnitude, period, offset, and direction. For joystick, only input parameters magnitude and period will show a change in effect. Out of range values will not cause error.

Parameter	**Default value**	**Value Range**
magnitude	10000	0 – 10000
period	0.05	Game pad: 0.00 – 10.00 Joystick: 0.04 – 0.20
offset	0	Game pad: -10000 – 10000
direction	0	Game pad: 0 - 35999

void ***StartPeriodicForce*** *( )*
Starts effect instances created by Setup Periodic Force.

void ***StopPeriodicForce*** *( )*
Stops effect instance created in Setup Periodic Force.

HRESULT ***CreateConstantForce*** *(double g_nXForce, double g_nYForce)*
Initializes and starts a constant force effect. Uses coordinate math to create direction and magnitude of force. Input parameters allowed are X Coordinate and Y Coordinate, as doubles, of relative origin and magnitude of force. Default values are 0, which creates a zero force with no direction. Expected range of values is -10000 to +10000. Out of range values will not cause error.
Only applicable to joystick! A constant force cannot be created for the game pad device.

HRESULT ***AlterConstantForce*** *(double g_nXForce, double g_nYForce)*
Allows the constant force effect to be changed without recreating the effect. Input parameters allowed are X Coordinate and Y Coordinate, as doubles, of relative origin and magnitude of force. Expected range of values is -10000 to +10000. Out of range values will not cause error. Only applicable to joystick!

void ***StartConstantForce*** *( )*
Starts effect instance created in Create Constant Force.
Only applicable to joystick!

void ***StopConstantForce*** *( )*
Stops effect instance created in Create Constant Force.
Only applicable to joystick!

void ***StartForce*** *( )*
Starts all effect instances created.
Joystick: starts constant and periodic force effects.
Game pad: same as Start Periodic Force.

void ***StopForce*** *( )*
Stops all effect instances created.
Joystick: stops constant and periodic force effects.
Game pad: same as Stop Periodic Force.

DirInputTester Utility.

The DirInputTester utility is a simple tool for designing and testing periodic and static (constant) forces. Depending on the connected device, DirInputTester will provide different slider controls through which the effect parameters can be modified. The user can create any desired effect and can read the function parameters directly from the user interface (see Figure 2).

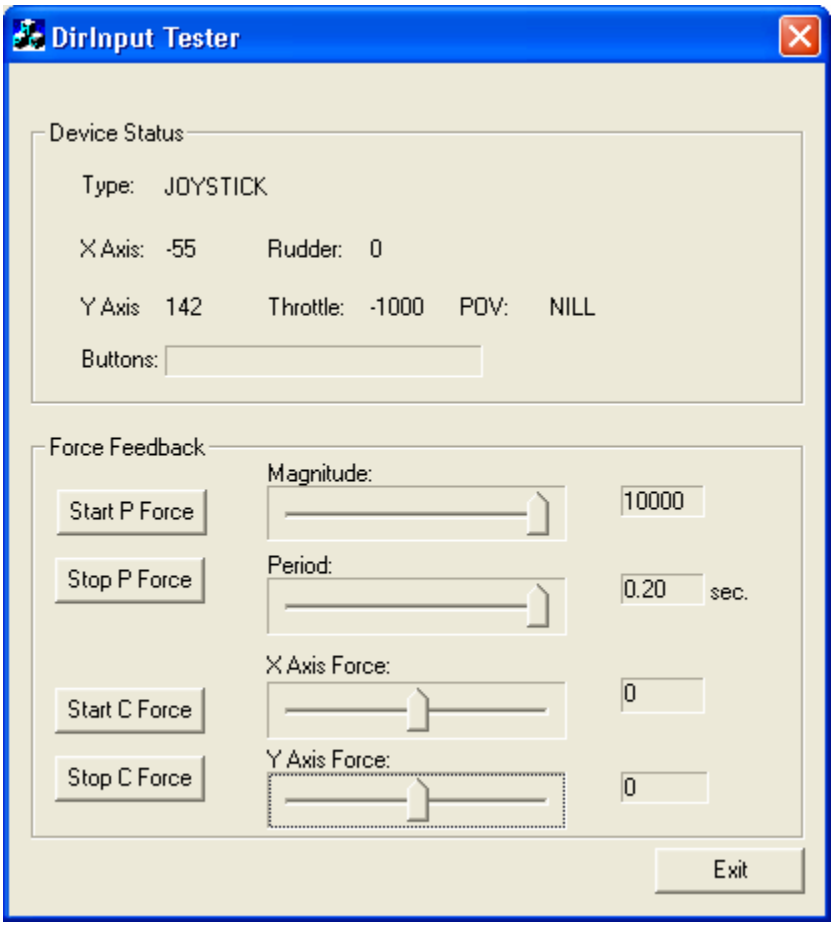

Figure 2: DirInputTester utility with connected joystick device

In Figure 2, the first two sliders modify the input parameters for function *AlterPeriodicForce*. The periodic effect can be started and stopped by pressing the *Start P Force* and *Stop P Force* buttons respectively. The shown parameters would create a very slow shaking effect but with maximum strength (magnitude).
Buttons *Start C Force* and *Stop C Force* enable and disable the constant force. The parameters *X Axis Force* and *Y Axis Force* get passed into function *AlterConstantForce*. The following table shows possible maximum values for *X Axis Force* and *Y Axis Force* and the resulting effect.

X Axis Force	*Y Axis Force*	**Effect**
0	0	No force effect
-10000	0	Strong pull to the left
10000	0	Strong pull to the right
0	-10000	Strong push backward
0	10000	Strong pull forward
10000	10000	Strong pull to top left
-10000	10000	Strong pull to top right
10000	-10000	Strong push to bottom left
-10000	-10000	Strong push to bottom right

In case a game pad is connected to the PC, DirInputTester provides sliders that control the magnitude (= revolutions), the period, the offset and the direction of a periodic force. A constant force cannot be created for the game pad.

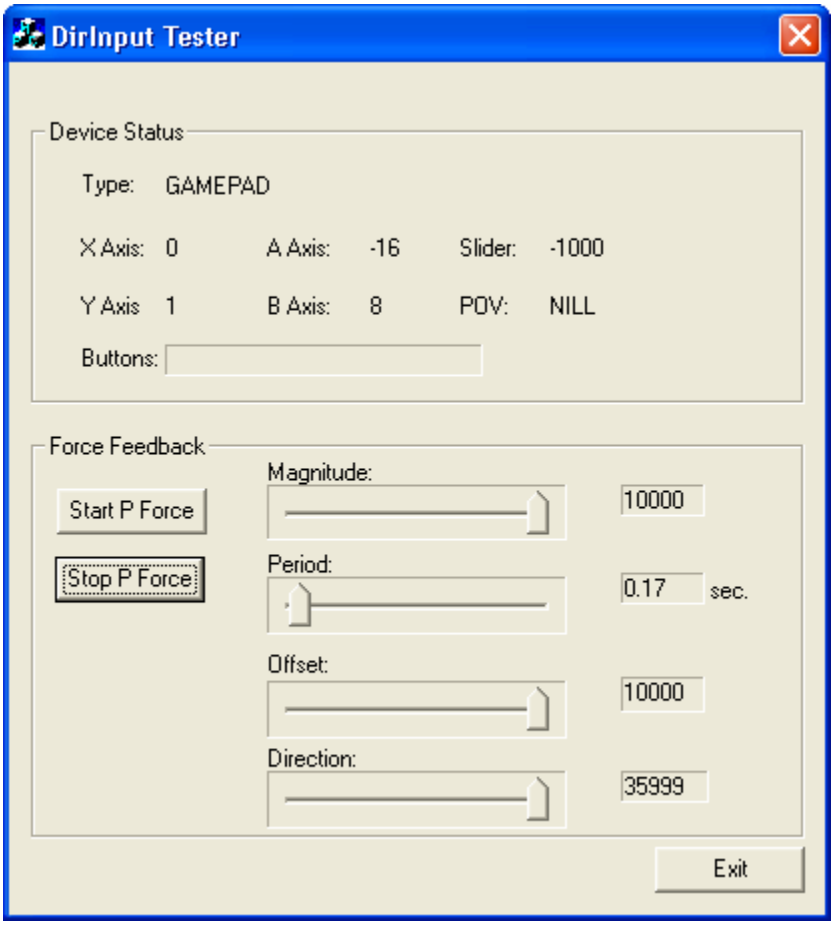

Figure 3: DirInputTester utility with connected game pad.

The parameters (Magnitude, Period, Offset, Direction) correspond directly to the parameters passed to function *AlterPeriodicForce*. The following table gives examples of some settings and the effect they create:

Magnitude	***Period***	***Offset***	***Direction***	**Effect**
10000	0.17	10000	35999	resembles a slow running airplane propeller
10000	0.00	0	35999	maximum vibration (max. revolutions and max. amplitude)
10000	0.00	0	9000	constant 'buzz' (minimum vibration)

Sources of Information and Assistance.

Microsoft MSDN Library
Microsoft DirectX 8 Sample Programs
Microsoft DirectInput Tutorial
Dr. Kelvin Sung