Steps to get HInput working with Gamebryo ‘Kinslayer’ App (revision 2)

1. Unzip the HInput distro somewhere sensible (e.g. “YourGame/HInput/”)

2. Add the include dir ("HInput/include") and lib dir ("HInput/lib") to the project

3. The system requires "HInput.lib" (Release) or "HInput_Debug.lib" (Debug). However, if you use VisualStudio then (as long as you include “HInput.h”) these will be automatically added for you i.e. you don’t need to manually add them to the project settings.

4. Copy the “/bin/HInput” folder to your game’s working directory. This contains the required plugin dlls for both release and debug modes. It is also where the system will save your configuration file.

5. You need to initialize (once only), update (every frame) and shutdown (once only) the system. I chose to put the code in StateManager.cpp. You will need to #include "HInput.h"

```
#include “HInput.h”

// Initialization code
HInput::System::Initialize(true, true, true, NULL, HInputLogCallback);
HInput::GetManager().LoadInputSetForPlayer("Kinslayer");

// Update code (suggest this goes at the beginning of the update method)
HInput::System::Update();

// Shutdown code
HInput::System::Shutdown();
```

6. Define a logging callback (before the initialize call). I used a free function in the cpp:

```
// HInput log callback
void HInputLogCallback(const std::string& str)
{
       NiOutputDebugString(str.c_str());
       NiOutputDebugString("\n");
}
```

7. In CharacterControlInputHandler.cpp, include "HInput.h"

8. Around line 90 there are two int variables set from the gamepad. Replace these calls with the code below (we multiply by 100 to match the range gamebryo expects – the HInput range is -1 to 1 – and invert the Y value to match Gamebryo).

```
int iVertAxisValue = (int)(-100.0f * HInput::GetManager().GetInputValue("MoveY"));
int iHorzAxisValue = (int)(100.0f * HInput::GetManager().GetInputValue("MoveX"));
```

9. You can now remove or comment out any preceding lines of the original code which are now redundant.

10. Now do the same for ThirdPersonCameraControlInputHandler.cpp, only use "CameraY" and "CameraX" as the input names instead of "MoveY" and "MoveX"

11. If you run the game you should now be able to use the original pad inputs as well as alternate keyboard WASD and IJKL controls.

Troubleshooting

- If the compiler complains it can't find the files, check you haven't misnamed the folder, e.g. my first tester accidentally called it HI**m**put.
- If you use a free function for the callback, as shown above, ensure it is defined before it is used.
- Make sure you're using the debug lib in the debug version or you will get a CRT assert (or just let HInput.h configure it for you with its #pragma comment(“lib”…)).