



Walkin'VR SDK

2018. 07. 18

기술개발실

박 광 석

목차

| | |
|--|---|
| 1. Visual C++ | 2 |
| SDK 구성..... | 2 |
| WalkinVR_SDK_Win32.dll | 2 |
| WalkinVR_SDK_Win32.lib | 2 |
| WalkinVR_SDK_Win64.dll | 2 |
| WalkinVR_SDK_Win64.lib | 2 |
| WalkinVR_SDK_Win.h..... | 2 |
| 인터페이스 함수 | 2 |
| int WalkinVR_Update() | 2 |
| void WalkinVR_GetData(WalkinData *dst) | 2 |
| Unreal Project 예시 | 3 |
| Unity Project 예시 (X64 Editor, Standalone)..... | 4 |

1. Visual C++

SDK 구성

WalkinVR_SDK_Win32.dll

WalkinVR_SDK_Win32.lib

WalkinVR_SDK_Win64.dll

WalkinVR_SDK_Win64.lib

WalkinVR_SDK_Win.h

인터페이스 함수

int WalkinVR_Update()

Walkin'VR 기기로부터 신호를 읽어와서 해석하고 내부 버퍼를 업데이트 한다. 리턴 값의 의미는 다음과 같다.

| | |
|-----|---|
| < 0 | Walkin'VR 기기와의 통신이 원활하지 않음 |
| = 0 | Walkin'VR 기기에서 새롭게 들어온 신호가 없음 |
| > 0 | Walkin'VR 기기에서 업데이트된 신호가 존재하며, Walkin'VR 기기의 n개의 샘플 데이터를 반영하고 있음. 자세한 내용은 부록을 참조. |

기기로부터 받은 데이터가 있는지 Polling하는 함수로, 주기적으로 호출해줘야 한다. 클라이언트의 frame rate가 USB HID의 통신 주기보다 짧으면 0을 리턴 할 경우가 있다. 만일 지속적으로 0이 나온다면 기기 자체의 결함이 있을 수 있다. 자세한 기기 결함을 판단하는 것은 차후에 업데이트 하도록 하겠다.

void WalkinVR_GetData(WalkinData *dst)

WalkinVR_Update()로 갱신한 내부 데이터를 읽어온다. WalkinData 구조체의 각 변수의 의미는 다음과 같다.

| | |
|----------|---|
| x1, y1 | 첫 번째 피사체의 좌표 |
| x2, y2 | 두 번째 피사체의 좌표 |
| vx1, vy1 | 첫 번째 피사체의 속도 |
| vx2, vy2 | 두 번째 피사체의 속도 |
| vx, vy | 두 피사체의 속도의 평균. 하나의 피사체만 존재할 경우 해당 피사체의 속도를 따른다. |

Walkin'VR은 사람의 발이 2개임을 감안하여, 최대 2개의 피사체 정보를 읽어올 수 있다. 즉, 두 피사체는 양 발에 대응시킬 수 있으나 기기의 특성상 각 피사체가 오른쪽인지 왼쪽인지는 판단할

수 없다. 좌표 공간은 (-100, -100)에서 (100, 100)까지다. 속도는 WalkinVR_GetData()를 호출한 시점 사이의 변위이기 때문에 클라이언트에서 가속도를 적용하지 않을 경우 WalkinVR_GetData()의 호출 주기와 관계없이 체감 이동 거리는 동일하다. 좌표 (x1, y1)이 (0, 0)일 경우 피사체가 존재하지 않는 것으로 본다. 마찬가지로 (x2, y2)가 (0, 0)이면 두 번째 피사체가 존재하지 않는다는 의미. 이는 중심봉의 좌표가 (0, 0)이기 때문에 사용한 표기법이다.

Unreal Project 예시

Build.cs 파일에는 추가 라이브러리 Path를 등록해준다.

```
public class Test001 : ModuleRules
{
    private string ModulePath
    {
        get { return ModuleDirectory; }
    }

    private string LibraryPath
    {
        get { return Path.GetFullPath(Path.Combine(ModulePath, "../../Binaries/")); }
    }

    public Test001(ReadOnlyTargetRules Target) : base(Target)
    {
        PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;

        PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "HeadMountedDisplay" });
        PublicAdditionalLibraries.Add(Path.Combine(LibraryPath, "Win64", "WalkinVR_SDK_Win64.lib"));
    }
}
```

라이브러리에 dll과 lib파일을 붙여넣는다.

| Program > Client > UnrealEngine > Test001 > Binaries > Win64 | | | | Win64 검... |
|--|------------------|---------------------|----------|------------|
| 이름 | 수정된 날짜 | 유형 | 크기 | |
| Test001Editor.target | 2018-05-16 오후... | TARGET 파일 | 235KB | |
| UE4Editor.modules | 2018-07-16 오후... | MODULES 파일 | 1KB | |
| UE4Editor-Test001.dll | 2018-05-16 오후... | 응용 프로그램 확장 | 209KB | |
| UE4Editor-Test001.pdb | 2018-05-16 오후... | Program Debug ... | 23,300KB | |
| UE4Editor-Test001-647.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-647.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| UE4Editor-Test001-1649.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-1649.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| UE4Editor-Test001-1981.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-1981.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| UE4Editor-Test001-2514.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-2514.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| UE4Editor-Test001-2559.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-2559.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| UE4Editor-Test001-3203.pdb | 2018-07-16 오후... | Program Debug ... | 21,340KB | |
| UE4Editor-Test001-9274.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-9274.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| UE4Editor-Test001-9515.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 206KB | |
| UE4Editor-Test001-9515.pdb | 2018-07-16 오후... | Program Debug ... | 23,292KB | |
| WalkinVR_SDK_Win64.dll | 2018-07-16 오후... | 응용 프로그램 확장 | 65KB | |
| WalkinVR_SDK_Win64.lib | 2018-07-13 오후... | Object File Library | 6KB | |

소스코드에서 다음과 같이 사용한다.

```
#include "WalkinVR_SDK_Win.h"
void ATest001Character::Tick(float DeltaSeconds)
{
    WalkinData data;
    int stat = WalkinVR_Update();
    if (stat < 0)
        UE_LOG(LogClass, Log, TEXT("Walkin'VR device not connected.));
    if (stat > 0)
    {
        WalkinVR_GetData(&data);
        FVector vec(data.vx, data.vy, 0);
        if (data.vx != 0 && data.vy != 0)
            UE_LOG(LogClass, Log, TEXT("%d, (%.3f, %.3f)", stat, data.vx, data.vy));
        AddMovementInput(vec, 1);
    }
}
```

Unity Project 예시 (X64 Editor, Standalone)

먼저 Unity 프로젝트 Root폴더에 WalkinVR_SDK_Win64.dll을 붙여 넣는다.

플레이어 클래스 내부에서 다음과 같이 코딩한다.

```
private struct WalkinData { public float x1, y1, x2, y2, vx1, vy1, vx2, vy2, vx, vy; }
[DllImport("WalkinVR_SDK_Win64.dll")]
extern private static int WalkinVR_Update();
[DllImport("WalkinVR_SDK_Win64.dll")]
extern private static void WalkinVR_GetData(ref WalkinData dst);

private void GetInput()
{
    // WalkinVR SDK에 의한 캐릭터 이동 컨트롤
    int stat = WalkinVR_Update();
    m_Input = Vector2.zero;
    if (stat > 0)
    {
        WalkinData wdata = new WalkinData();
        WalkinVR_GetData(ref wdata);
        m_Input = new Vector2(wdata.vx, wdata.vy);
    }
    if (stat < 0)
    {
        Debug.Log("Walkin'VR disconnected.");
    }
    m_Input *= m_SpeedMultiplier;
}
```

빌드 후 게임 실행파일이 있는 폴더에도 WalkinVR_SDK_Win64.dll이 있어야 한다.