

뇌-컴퓨터 인터페이스 소프트웨어 플랫폼: Openvibe2Unity(O2U) Development Guide

February , 2019



Contents

1 뇌파 게임 개발에 앞서

1.1 어디까지 가능한가? OpenViBE2Unity

2 OpenViBE2Unity 개발환경 및 플랫폼 구조

2.1 OpenViBE

2.2 UIVA

2.3 Unity

2.4 뇌파 데이터의 통신 과정

3 Openvibe에서의 O2U 사용 및 활용

3.1 OpenViBE 사전 설정

3.2 OpenViBE & UIVA

3.3 OpenViBE & Python & UIVA

4 Unity에서의 O2U 사용 및 활용

4.1 함수 설명

4.2 함수 사용 예제

5 O2U로 개발된 게임 실행 순서

6 O2U로 개발된 뇌파 게임

6.1 MindCar 2.0

6.2 BCI World Tour

1. 뇌파 게임 개발에 앞서

안녕하세요. 한동대학교 Brain-Computer Interface Lab(BCI Lab)입니다. 저희 연구실은 다양한 BCI 연구를 진행 중에, P300 과 하이퍼스캐닝 연구에 있어서 게임 플랫폼의 수요를 느껴, P300 을 이용한 세계일주, MindCar 2.0 등 다양한 게임을 개발하고 실험 중에 있습니다. 게임 개발 진행에 있어, 가장 큰 장벽이라고 느꼈던 부분이 OpenViBE 플랫폼과 Unity 3D 와의 연동이었습니다. 저희 뿐만 아니라, 뇌파 게임 개발자들이 뇌파 측정 장치로부터 뇌파 원 신호를 게임 플랫폼으로 전달 및 처리 하는 것에 어려움을 느끼는 점을 인지하여, 게임을 개발하면서 UIVA 소프트웨어를 활용하여 OpenViBE 와 Unity 간의 실시간 데이터 전달을 위해 개발했던 부분을 API 화 한 OpenViBE2Unity 1.0.0 를 제공하려고 합니다. 본 API 는 OpenViBE 에서 Unity 간 데이터 전달을 목표로 하며, 뇌파 신호 데이터 전달에 국한되지 않고, UIVA 에서 지원해주는 Analog 와 Button 신호를 이용하여 다양하게 활용될 수 있습니다. OpenViBE, Unity 각 오픈소스에서 어떻게 해주어야 하는지를 기준으로 단원 별로 자세히 명시되어 있습니다. 본 Development Guide 는 O2U 를 통한 뇌파 게임 개발에 필수적인 정보만을 담고 있어 사용되고 있는 오픈 소스에 대한 상세 설명은 기대할 수 없습니다.

1.1 어디까지 가능한가? OpenViBE2Unity

OpenViBE2Unity 1.0.0(이하: O2U)는 다시 말해, OpenViBE 에서 Unity 로 사용자의 뇌파 원 신호를 보내는 것이 최우선 목적입니다. 하지만, O2U 의 확장성은 여기서 그치지 않습니다. O2U 의 기반이 되는 UIVA 는 Analog 형 신호와 Button 형 신호를 제공하는데 이를 활용한 기능을 O2U 는 지원하여 뇌파 게임에 적용할 수 있습니다. Analog 형 신호는 실시간으로 int 값을, Button 형 신호는 bool 값으로 OpenViBE 로부터 Unity 로, 또한 Unity 에서 OpenViBE 로 전달합니다. OpenViBE 에서 Button 신호를 받을 때에는 bool 값으로 받아오게 되는 반면, Unity 에서 Button 신호를 받을 때에는 True 일 때에 대문자, False 일 때에 소문자를 출력하게 됩니다. 이를 이용하여 다양하게 뇌파 게임에 적용할 수가 있는데, Analog 형 신호로 뇌파 원 신호를 Unity 로 받아오고, Button 신호를 OpenViBE 로 보내 Mark 를 찍어주는 형식으로, 더 나아가서 Python 에서 신호처리 조건 인자로도 쓰일 수 있습니다(6.1~2 참고). 또한, 다수의 뇌파 장비를 이용하는 것 또한 한 게임 내에서 지원합니다(3.2 SimpleSingalProcessingMulti 참고).

2 OpenViBE2Unity 플랫폼 구조 및 개발환경

오픈소스 기반 뇌파 게임을 개발하기 앞서 환경설정이 필요합니다. 각 오픈 소스에 대한 설치 방법에 대한 설명과 유의점들을 서술합니다. OpenViBE2Unity는 Window 7 64bit, Window 10 64bit 환경에서 작동함을 확인하였습니다.

2.1 OpenViBE

<http://openvibe.inria.fr/downloads/>

위 주소에서 최신 버전 OpenViBE (guide 기준 2.1.0)를 설치합니다. 본 O2U 의 기능은 소스코드 버전을 다운 받지 않아도 무관합니다.

2.2 UIVA

UIVA 는 본 연구실의 링크를 통하여 환경설정 해주어야 O2U 를 활용할 수 있습니다.

Window 7 64bit 인데스크탑에서는 빌드하지 않고 Release 된 파일을 사용할 수 있습니다. 그렇지 않은 경우에는 환경설정이 필요합니다. UIVA 환경을 빌드하기 위해서는 Visual C# 2008 express edition, Visual C++ 2008 express edition 설치가 요구됩니다.

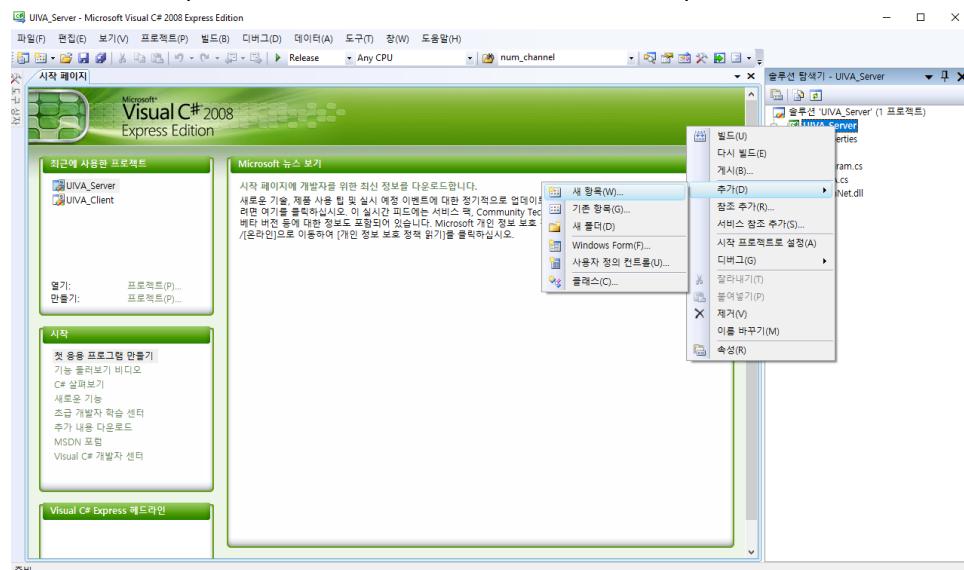


그림 1 Visual C# 2008 에서 UIVA_Server 프로젝트에 VrpnNet.dll 을 추가하는 모습

Visual C# 실행 -> UIVA_Server -> 솔루션 탐색기 우클릭 -> 추가(C) -> 기존항목 VrpnNet.dll 선택

UIVA_Client 는 따로 빌드하지 않아도 되며, UIVA_Server 만 빌드합니다. 그림 1 의 과정을 거쳐 VrpnNet.dll 파일을 추가하여 빌드하면 UIVA 환경이 성공적으로 설정됩니다.

빈번한 빈도로 하단의 2 가지 에러가 나게 됩니다.

1. 오류: “응용 프로그램의 side-by-side” 구성이 잘못되어 응용 프로그램을 시작하지 못했습니다.

명령줄 sxstrace.exe 도구를 사용하십시오.”

2. 오류: “vrpnNet.dll 파일의 NET version 이 프로그램의 버전보다 높다.”

우선, 두 번째 오류부터 해결한 뒤에, 첫 번째 오류를 잡아주어야 합니다.

두 번째 오류는 자신의 OS version에 맞는 vrpnNet.dll 파일을 다운로드 후 위의 절차를 따르면 해결됩니다. (본 O2U에서 지원하는 vrpnNet.dll 파일은 window 7 64bit 기준입니다.) 두 번째 오류를 해결했거나, 뜨지 않지만 첫 번째 오류가 뜨는 경우에는 C++ runtimes 버전이 맞지 않거나, 설치되지 않아 생기는 오류입니다. (또는 visual C++ 2008이 깔리지 않아서 생기기도 합니다.) 그러기에, Visual C# 2008 express edition, Visual C++ 2008 express edition 설치여부를 한번 더 확인 한 뒤에, 해당 PC 내에 있는 C++ redistributable exe 파일을 전부 삭제 후 재설치 시켜줍니다.

위 작업이 끝나면 다시 처음부터 빌드를 진행합니다.

2.3 Unity

https://store.unity.com/kr/?_ga=2.12388276.1622912927.1548047424-1547254793.1545891157

위 주소에서 Unity를 설치합니다. O2U는 Unity 3D Personal version을 이용해도 무관합니다.

2.4 뇌파 데이터의 통신 과정

O2U 를 활용하여 개발되는 게임의 시스템 구조는 ‘그림 2’와 같습니다. ‘그림 2’은 뇌파 측정을 위한 뇌전도 장비로부터 뇌파 게임에 이르기까지의 뇌파 데이터 통신을 초점으로 보고 작성된 그림입니다. 뇌전도 장비를 통해 측정된 뇌파는 전처리 단계로써 OpenViBE 내의 Script Box 를 이용하여 채널 및 주파수 대역 선택을 통한 시·공간적 정보로 필터링을 수행 할 수 있습니다(3.2~3 참고). 이후 연구 방향에 맞게 뇌파 원 신호를 사용자의 의도에 맞게 Python 코드를 통해 추출하는 것, Unity 로부터 입력 받은 Button 신호를 통해 Python 내 조건문으로 상황에 따른 뇌파 처리 과정 구분 또한 가능합니다. 이후 데이터 수집 서버인 VRPN(Virtual-Reality Peripheral Network)을 통해 UIVA_Server 와 통신합니다. 이후, VRPN, UIVA, 그리고 Application 인 Unity3D 와의 통신은 정방향, 역방향 두방향으로 통신이 가능합니다.

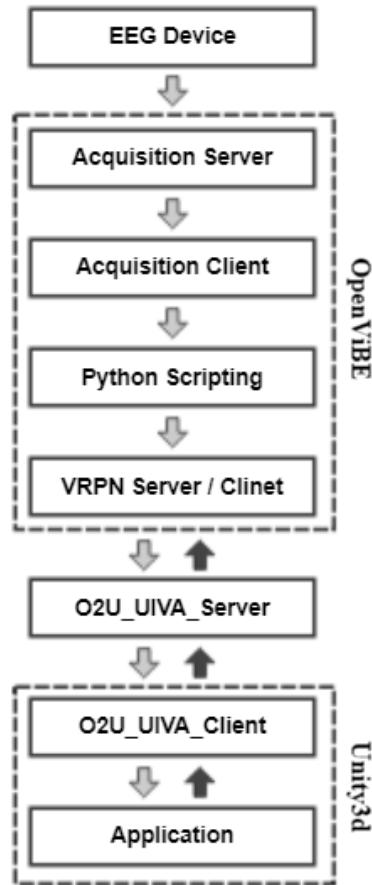


그림 2 O2U 를 통한 EEG Device 로부터 Unity 3D 까지의 통신 과정

여기서 O2U 는 정방향 통신(OpenViBE to Unity)으로 Analog 와 Button 신호를, 역방향 통신(Unity to OpenViBE)으로 Button 신호 만을 제공합니다. Analog 와 Button 에 대한 설명은 1.1 을 참고하십시오.

3 플랫폼 사용방법

본 게임 개발을 위해 O2U 가 지원하는 점은 OpenViBE 와 Unity 사이의 통신이지만, O2U 를 사용하기 위해 선행되어야 할 OpenViBE, Unity 에 대해서도 추가로 언급하며, 시나리오 예제 또한 지원합니다. 설명의 진행 순서는 OpenViBE 사전 설정, O2U 의 사용과 활용 , 그리고 Unity 내 O2U 적용 순으로 진행됩니다.

3.1 OpenViBE 사전 설정

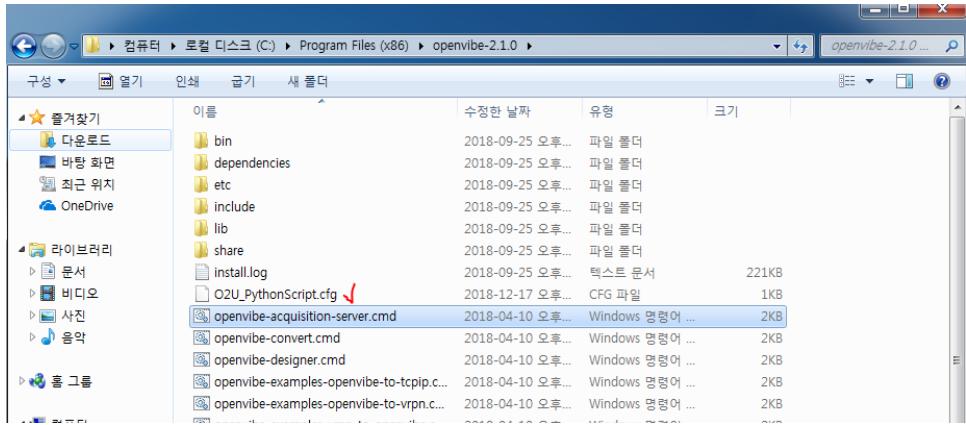


그림 3 openvibe-acquisition-server.cmd 화면

OpenViBE 가 설치가 완료되면, C:/Program Files 에 OpenViBE 가 위치하게 됩니다.(예: openvibe-2.1.0) 이후 openvibe-2.1.0(버전에 따라 상이할 수 있음)폴더에 들어간 후, openvibe-acquisition-server.cmd 를 더블 클릭합니다(그림 3.) openvibe-acquisition-server.cmd 는 OpenViBE 의 습득 서버를 통해서 디자이너 내부 시스템으로 사용자의 뇌파 원 신호를 보내는 역할을 담당합니다. 이후 '그림 4'의 창을 볼 수 있는데, 우측의 Connect 버튼을 클릭하면 EEG device 와 OpenViBE 를 연 결한다는 의미이고 Play 를 누르게 되는 순간부터 디자이너에 뇌파 원 신호를 보내기 시작합니다.

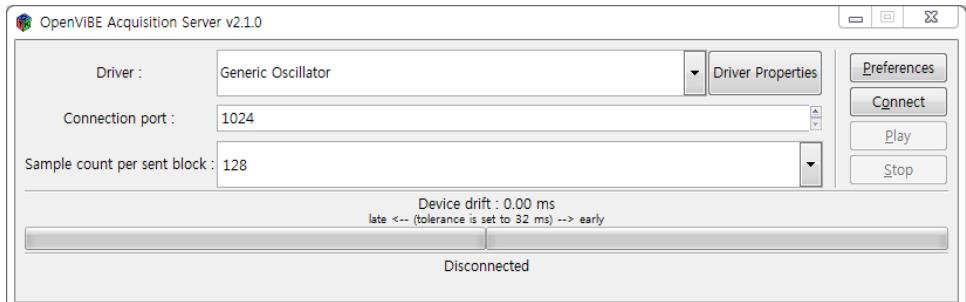


그림 4 Openvibe Acqisition Server 창 화면.

BioSemi Active Two (MkI and MkII)
Brain Products actiCHamp
Brain Products amplifiers (through BrainVision Recorder)
Brain Products BrainAmp Series
Brain Products LiveAmp
Brain Products V-Amp / First-Amp
Brainmaster Discovery and Atlantis
Cognionics (<i>unstable</i>)
CTF/VSM MEG (<i>unstable</i>)
EEGO
EGI Net Amps 300 (through AmpServer) (<i>unstable</i>)
Emotiv EPOC
Enobio3G (<i>unstable</i>)
Fieldtrip Driver
g.tec g.Nautilus using g.NEEDaccess (<i>unstable</i>)
g.Tec gMOBILab+ (<i>unstable</i>)
g.Tec gUSBamp Gipsa-lab
g.Tec gUSBamp Legacy (<i>deprecated</i>)
Generic Oscillator
Generic Raw File Reader
Generic Raw Telnet Reader
Generic Saw Tooth
Generic Time Signal
LabStreamingLayer (LSL) (<i>unstable</i>)
mBrainTrain Smarting
MCS NVX amplifier (<i>unstable</i>)
Micromed SD LTM (through SystemPlus Evolution)
MindMedia Nexus32B
Mitsar EEG 202 - A
NeuroRT Driver Collection [Brain Rhythm 8]
NeuroRT Driver Collection [Oscillator]
NeuroRT Driver Collection [Simulator]
NeuroRT Driver Collection [SmartBCI]
NeuroRT Driver Collection [Wearable Sensing Dry Sensor Interface]
NeuroServo (<i>unstable</i>)
NeuroSky MindSet (MindSet Dev. Kit 2.1+)
OpenAL audio capture (<i>unstable</i>)
OpenBCI
OpenEEG Modular EEG P2
TMSI amplifiers
TMSI Refa32B Legacy (<i>unstable</i>) (<i>deprecated</i>)

EEG Device 를 Connect 하기 이전에, 준비해야할 일련의 과정들이 있습니다. 먼저 Driver 의 화살표 버튼을 눌러보면 Openvibe Acquisition Server 에서 지원해주는 EEG Device 들을 보여줍니다. (그림 5) 추가적인 세부사항 설정도 필요로 합니다. Connection port 를 통해서 뒤에서 설명할 Designer 내의 Acquisition client 와 동일한 port 로 지정해주어야 하며(표준설정은 1024), Sample count per sent block 은 버퍼의 크기를 나타내며, 단일 버퍼에서 수집된 채널 당 몇개의 샘플을 전송해야 하는지 정의합니다. (기본 값은 23) 본인의 해당 device 를 선택하면 되며, 본 예제에서는 편의를 위해 기기를 연결 하지 않는 'Generic Oscillator'를 이용해서 설명을 이어 진행합니다. Preference 를 통해서는 채널 수, 버퍼의 지속시간을 결정하는 역할을 하는 샘플링 주파수를 선택할 수 있습니다.

그림 5 Openvibe Acqusition Server 내 지원 뇌전도장비 목록

(Driver Properties 와 Preferences 에 대한 더 자세한 설명과 문제해결은 <http://openvibe.inria.fr/acquisition-server/> 참고) 이후 Connection, Play 버튼을 누르게 되면, 성공적으로 Device 가 연결되고 cmd 창 내에서 Connection 과 Raw signal acquiring 로그를 확인할 수 있습니다. (그림 6)

```
[ INF ] Connecting to device [Generic Oscillator]...
[ INF ] Connection succeeded!
[ INF ] Starting the acquisition...
[ INF ] Now acquiring...
```

그림 6 Acquisition server cmd 화면.

3.2 OpenViBE 에서의 O2U 사용 및 활용

도입부에서 언급했듯이 O2U 를 이용하여 뇌파 원 신호를 OpenViBE 에서 Unity 로 보내는 것을 넘어서서 다양하게 O2U 를 활용할 수 있는데(1.2 참고), 이를 위한 OpenViBE Box 활용에 대한 설명입니다. 쓰여지는 오픈소스의 종류, 사례를 통한 활용 등의 기준으로 3 가지로 나누어 예제를 활용하여 설명합니다. (본 단원에서 기록된 예제들은 Github 수록)

3.2 Openvibe & UIVA

OpenvibeToUnityAnalog

EEG Device 에서 측정된 뇌파 신호를 OpenViBE 에서 Unity 로 Analog 신호를 전달합니다(그림 7).

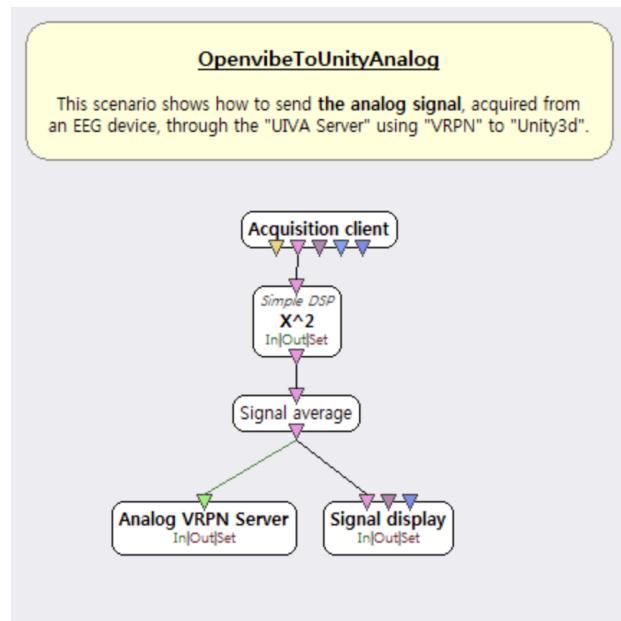


그림 7

사용된 Box 목록

Acquisition client, Simple DSP, Signal average, Analog VRPN Server, Signal display

(사용된 box 의 기능은 Openvibe box tutorial 참고)

과정

1. Openvibe Acquisition Server 을 실행하고 세부사항을 설정합니다(3.1 참고).

세부사항 예제) Driver = 'Generic Oscillator', Port = '1024', Sample count = '128', In the properties, # of channels = '1', Fs = '128' ..

2. OpenvibeToUnityAnalog.mxs 내 "Acquisition client"Box 의 Acquisition server port 를 Acquisition Server 과 동일하게 설정합니다.

세부사항 예제) 1024

3. OpenvibeToUnityAnalog.mxs 내 "Analog VRPN Server"Box 의 Peripheral name 을 "UIVA_Server.cfg"내의 DEV_OPENVIBE_ANALOG 와 동일하게 지정합니다.

세부사항 예제) openvibe_vrpn_analog@localhost

OpenvibeToUnityButton

OpenViBE에서 Unity로 Button 신호를 전달합니다(그림 8).

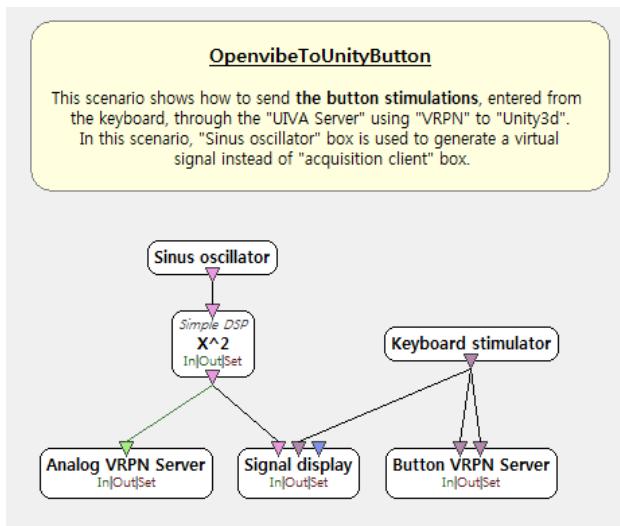


그림 8

사용된 Box 목록

Sinus oscillator, Simple DSP, Analog VRPN Server, Signal display, Keyboard stimulator, Button VRPN Server

(사용된 box 의 기능은 Openvibe box tutorial 참고)

순서

1. OpenvibeToUnityButton.mxs 내 “Sinus oscillator” Box 의 설정을 완료합니다.

세부사항 예제) # of channels = '4', Fs = '1', Sample count = '1'

2. OpenvibeToUnityButton.mxs 내 “Analog VRPN Server”Box 의 Peripheral name 을 “UIVA_Server.cfg”내의 DEV_OPENVIBE_ANALOG 와 동일하게 지정합니다.

세부사항 예제) openvibe_vrpn_analog@localhost

3. OpenvibeToUnityButton.mxs 내 “Button VRPN Server”Box 의 Peripheral name 을 “UIVA_Server.cfg”내의 DEV_OPENVIBE_BUTTON 과 동일하게 지정합니다.

세부사항 예제) openvibe_vrpn_button@localhost

Openvibe designer 실행 중에 키보드 ‘a’ 또는 ‘z’를 누르면, “Signal display”Box 에서 stimulation 을 확인할 수 있고, 추가적으로 Unity 에서도 전달받았음을 확인 가능합니다.
(4.2 참고)

UnityToOpenvibeButton

Unity에서 OpenViBE로 Button 신호를 전달합니다(그림 9).

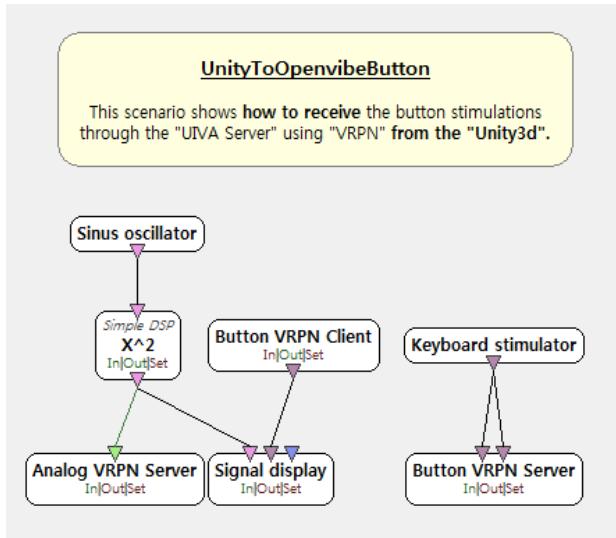


그림 9

사용된 Box 목록

Sinus oscillator, Simple DSP, Analog VRPN Server, Signal display, Keyboard stimulator, Button VRPN Server

(사용된 box 의 기능은 OpenViBE box tutorial 참고)

과정

- UnityToOpenvibeButton.mxs 내 “Button VRPN Server”Box 의 Peripheral name 을 “unity_button@localhost:50555”로 지정합니다.

“Button VRPN Server”Box 를 제외한 나머지 Box 설정은 OpenvibeToUnity 설정과 동일합니다.

3.3 OpenViBE & Python & UIVA

주의

Python script 를 사용하기 전에 ‘O2U.py’를 Python library 에 추가시켜야 합니다.

SimpleSignalProcessing

EEG device 에서 측정된 뇌파 신호를 Python script 처리 후 Analog 신호로 OpenViBE에서 Unity로 전송합니다(그림 10).

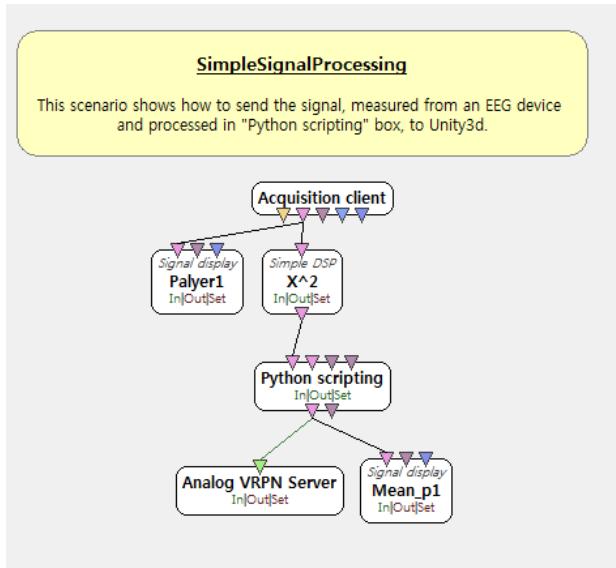


그림 10

사용된 Box 목록

Acquisition client, Signal display, Simple DSP, Python scripting, Analog VRPN Server

(사용된 box 의 기능은 OpenViBE box tutorial 참고)

과정

1. "SimpleSignalProcessing.mxs" 내 "Acquisition client"의 port number 를 Acquisition server 와 동일하게 설정합니다.
세부사항 예제) 1024
2. "SimpleSignalProcessing.mxs" 내 "Python scripting"의 경로를 지정해줍니다.
세부사항 예제) D:/~/script/SimpleSignalProcessing.py
3. "SimpleSignalProcessing.mxs" 내 "Analog VRPN Server" Box 의 Peripheral name 을 "UIVA_Server.cfg"내의 DEV_OPENVIBE_ANALOG 와 동일하게 지정합니다.
세부사항 예제)openvibe_vrpn_analog@localhost

SimpleSignalProcessingMulti

EEG device 2 기에서 측정된 뇌파 신호를 Python script 처리 후 2 개의 Analog 신호로 OpenViBE에서 Unity로 전송합니다(그림 11).

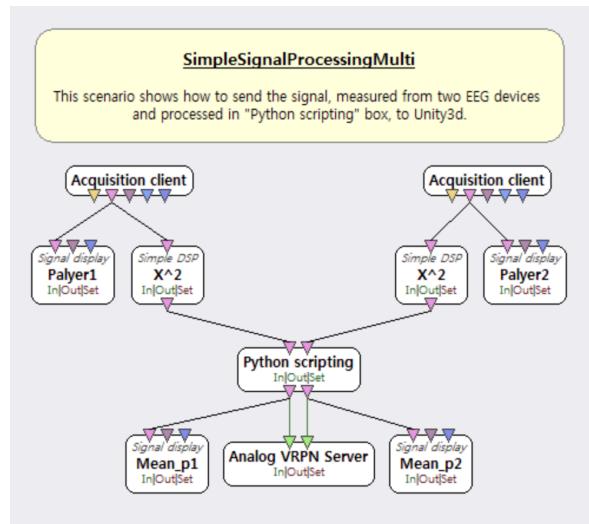


그림 11

※이하 내용은 SimpleSignalProcessing 과 동일 (참고 3.3)
Training

EEG Device에서 측정된 뇌파 신호를 OpenViBE 내에서 "Python Scripting" Box를 활용해 학습한 결과를 바탕으로 분류기를 생성합니다. 실행 중 키보드 'a'를 누르게 되면 설정된

시간 동안 학습을 진행하게 되고, 설정된 시간이 지나면 시나리오는 자동적으로 학습을 종료합니다(그림 12).

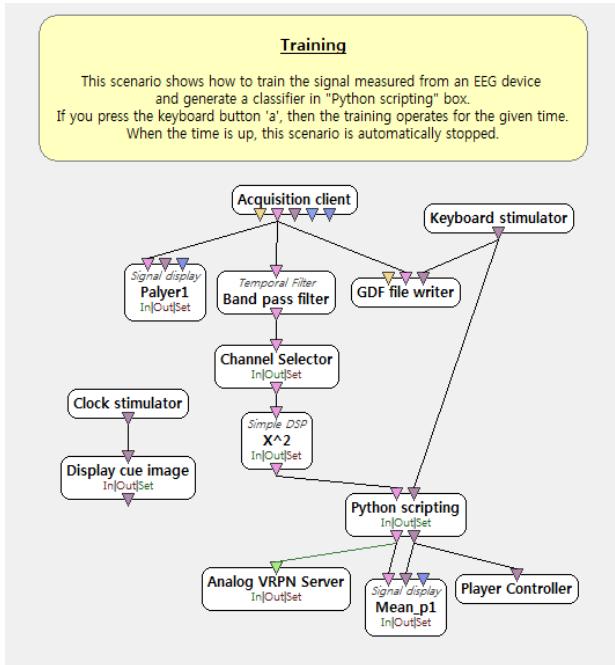


그림 12

사용된 Box 목록

Acquisition client, Signal display, Temporal Filter, Keyboard stimulator, GDF file writer, Channel Selector, Simple DSP, Clock stimulator, Display cue image, Python Scripting, Analog VRPN Server, Signal display, Player Controller
(사용된 box 의 기능은 OpenViBE box tutorial 참고)

과정

1. "Training.mxs" 내 "Display cue image"Box의 경로 설정을 해줍니다. 학습을 하는 동안 십자 모양의 이미지를 보여줍니다.

세부사항 예제) D:/~/image/fixation_cross.png

2. "Training.mxs" 내 "GDF file writer"Box의 경로 설정을 해줍니다.

세부사항 예제) D:/~/devTraining.gdf

3. "Training.mxs" 내 "Temporal Filter"Box의 다양한 필터 종류 중에서 신호 처리를 위한 설정을 완료합니다.

세부사항 예제) Band pass Butterworth filter of 4th order, 15~20Hz

4. "Training.mxs" 내 "Channel Selector"Box에서 학습에 사용되는 전극을 선택합니다.

세부사항 예제) Channel name => AF3;AF4;F3;F4;Fz or
Index => 6;8;13;15;17

5. "Training.py"내에서 'classifier.txt'로 경로 설정을 한 다음, 학습 시간을 설정해줍니다.

세부사항 예제) line 17, filename = 'D:\classifier.txt'
line 18, trainingTime = 30 //seconds

※ 추가적으로 SimpleSignalProcessing 의 과정 또한 선행되어야 합니다.

Playing

EEG Device를 통해 측정된 뇌파 신호가 OpenViBE 내에서 학습을 통해 생성된 분류기에 의해 분류되고 이 신호는 Unity3D로 전송됩니다(그림 13).

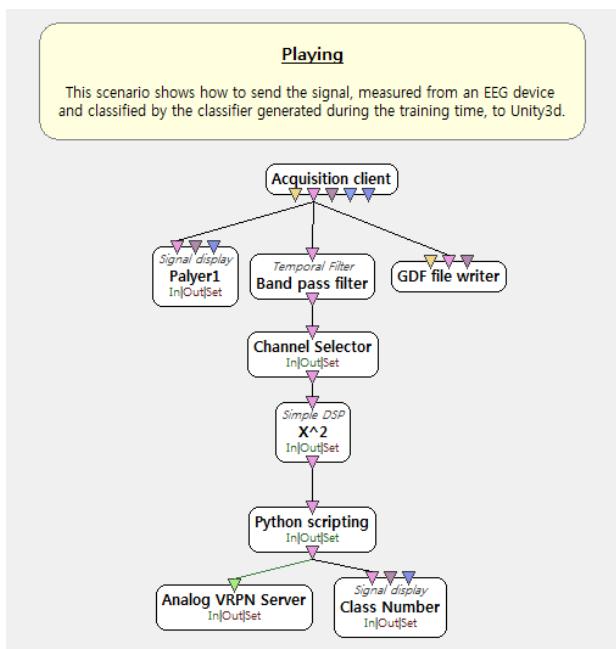


그림 13

사용된 Box 목록

Acquisition client, Signal display, Temporal Filter, GDF file writer, Channel Selector, Simple DSP, Python scripting, Analog VRPN Server

(사용된 box의 기능은 OpenViBE box tutorial 참고)

과정

1. "Training.py" 내의 읽을 파일("classifier.txt") 경로를 지정합니다.

세부사항 예제) line 15, filename = 'D:\classifier.txt'

4 Unity에서의 사용 및 활용

4.1 함수 설명

UIVA_Client.cs 내 함수들의 쓰임을 설명합니다.

OpenvibeToUnityAnalog

```
public void GetOpenvibeAnalog(int which, out DateTime analTS, out int numOfChannels, out List<double> signal)
```

인자 설명

1. which : Openvibe Analog ID (Analog VRPN Server 하나만 쓸 경우 생략가능)
2. analTS: 시작 시간으로부터, Analog 신호가 Unity로 전달 받는 시간 (Time Stamp)
3. numOfChannels: Analog 신호를 받아오는 각 EEG Device 전극의 개수
4. signal: Analog 신호

(List 형으로 다수의 EEG Device의 신호를 받아오는 것이 가능합니다.)

OpenvibeToUnityButton

```
public void GetOpenvibeButton(int which, out DateTime buttTS, out string buttA, out string buttZ)
```

인자 설명

1. which: Openvibe Button ID (Button VRPN Client 하나만 쓸 경우 생략가능)
2. buttTS: 시작시간으로부터 Button 신호가 Unity로 전달받는 시간 (Time Stamp)
3. buttA: keyboard 'A'키 입력 시, 대문자 'A', 입력이 들어오지 않을 시, 소문자 'a'
4. buttZ: keyboard 'Z'키 입력 시, 대문자 'Z', 입력이 들어오지 않을 시, 소문자 'z'

UnityToOpenvibeButton

```
public void PutOpenvibeButton(int buttNum)
```

인자 설명

1. buttNum: 신호를 보낼 Button VRPN Server 의 Button (범위: 0~5)

4.2 함수 사용 예제

O2U 를 잘 이해하기 위해, 예제를 제공합니다. for_experiment 폴더에 모든 파일이 첨부되어 있습니다. 예제 설명에 앞서 개발하려고 하는 게임 Scripts 파일에 UIVA_Client.cs 가 존재해야 한다는 것을 주의하여야 합니다.

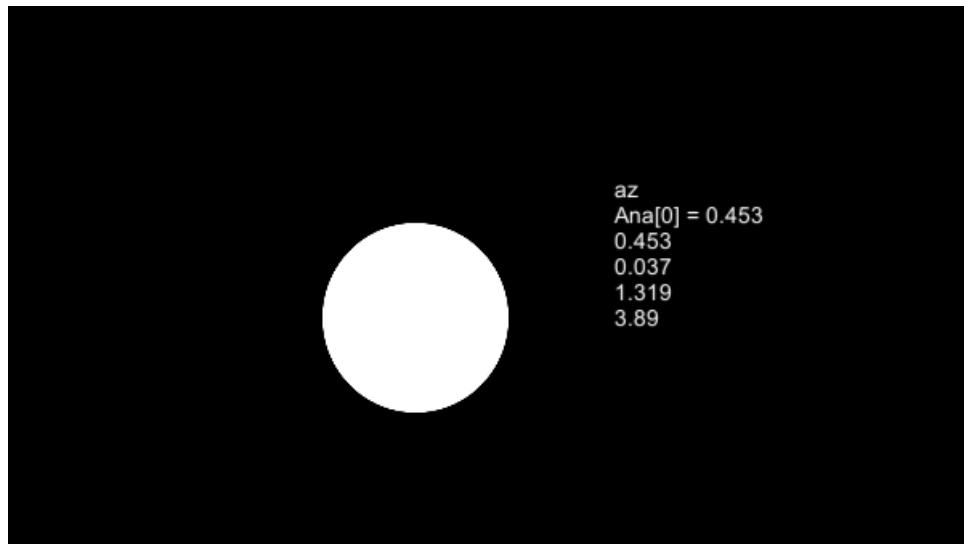


그림 14 사용 예제 실행 화면

'그림 14'은 검은 배경에 Analog 신호에 따라 화면에 보이는 공의 높낮이가 달라지며, buttA 와 buttZ 버튼 신호가 들어오는 순간 화면의 a,z 가 각각 대문자 A,Z 가 되는 간단한 예제입니다.

대부분의 Analog 와 Button 신호에 따른, Scene 변화는 BallMovement.cs 에서 처리되는데, 이를 보면 4.1 의 함수들의 활용을 이해할 수 있습니다.

```

//for forward communication
public UIVA_Client theClient;
string ipUIVAServer = "127.0.0.1";

// analog data
public DateTime analogTS;
public int numOfChannels;
public List<double> signal = new List<double>();

// button data
public DateTime buttonTS;
public string buttA;
public string buttZ;

public Text scoreText;
public GameObject Ball;
public double conval = 5f;
public double curconval = 2f;

// Use this for initialization
void Start()
{
    theClient = new UIVA_Client(ipUIVAServer);
}

```

그림 15 BallMovement.cs 일부 코드 1

UIVA_Server 가 존재하는 IP 를 입력하여, UIVA_Client 를 선언하는 것과 함수들의 인자를 받아오기 위한 사전 변수 선언이 요구됩니다(그림 15).

```

theClient.GetOpenvibeAnalog(out analogTS, out numOfChannels, out signal);
theClient.GetOpenvibeButton(out buttonTS, out buttA, out buttZ);

//Use when you want to check UnityToOpenvibeButton
if(buttA.Equals("A"))
{
    theClient.PutOpenvibeButton(0);
}

if (buttZ.Equals("Z"))
{
    theClient.PutOpenvibeButton(1);
}

```

그림 16 BallMovement.cs 일부 코드 2

그림 16 를 통해 4.1 의 함수 사용 활용을 이해할 수 있습니다.

Unity 기준, GetOpenvibeAnalog 를 통해, Analog 신호를 signal 로 받아올 수 있게 됩니다.

GetOpenvibeButton 으로 키보드 'A','Z' 키의 입력을 받아올 수 있게 됩니다. PutOpenvibeButton 으로 OpenViBE 6 개의 버튼 입력을 줄 수 있게 됩니다. 추가적으로, 예제에서 키보드 'A','Z'를 입력 받을 시, 각각 OpenViBE 0, 1 버튼에게 출력을 주게 됩니다.

```
for(int i = 0; i < numOfChannels; i++)
{
    sigStr += signal[i].ToString() + "\n";
}
sigStr = buttA + buttZ + "\n" + sigStr;

scoreText.text = sigStr;
```

그림 17 BallMovement.cs 일부 코드 3

그림 17에서, GetOpenvibeAnalog 를 통해 받아온 numberOfChannels 와 signal 변수를 통하여 각 전극의 뇌파 원 신호를 받아옵니다. GetOpenvibeButton 을 통해 받아온 buttA, buttZ 를 통해서 keyboard 'A','Z' 키를 입력받을 때에 Unity 화면에서, 소문자 'a','z'가 대문자 'A','Z'로 변환된 상태로 유지됨을 확인할 수 있습니다.

※ 해당 코드의 전체보기는 GitHub 내에 BallMovement.cs 에서 확인하십시오.

5 O2U 로 개발된 게임 실행 순서

과정

1. Openvibe-acquisition-server.cmd 실행

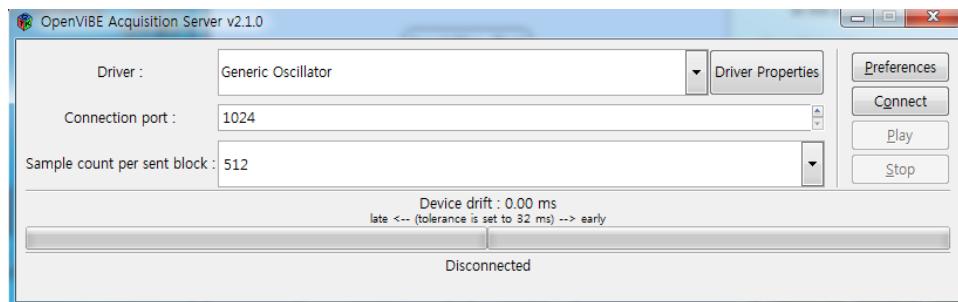


그림 18 Openvibe Acqusition Server 실행 화면

2. Openvibe-designer.cmd 실행

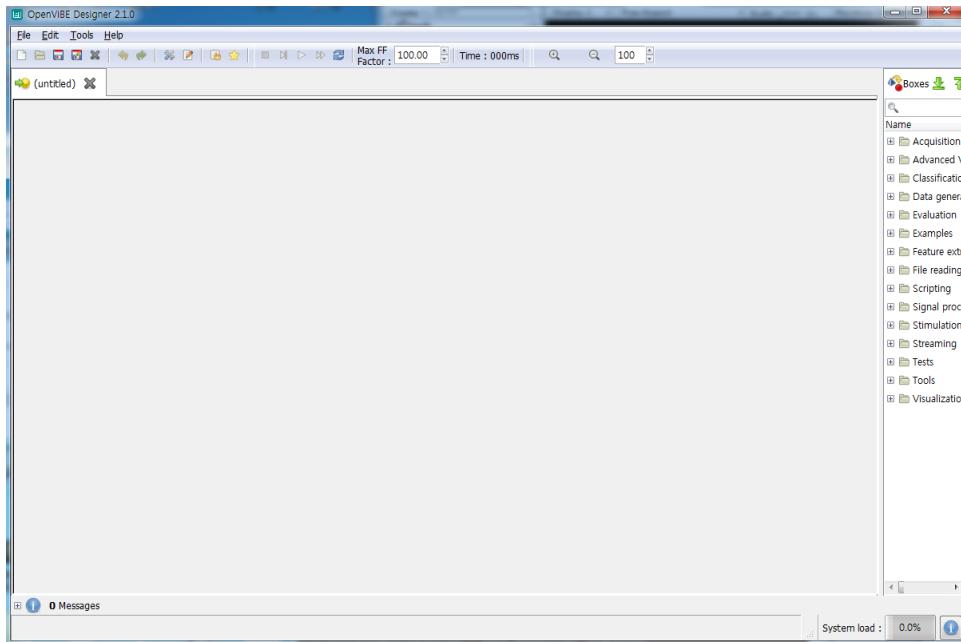


그림 19 Openvibe designer 실행 화면

3. Openvibe designer 내에서 상단 탭에서 'File-> Open', 필요한 시나리오 파일 열기.

예) OpenvibeToUnityAnalog.mxs

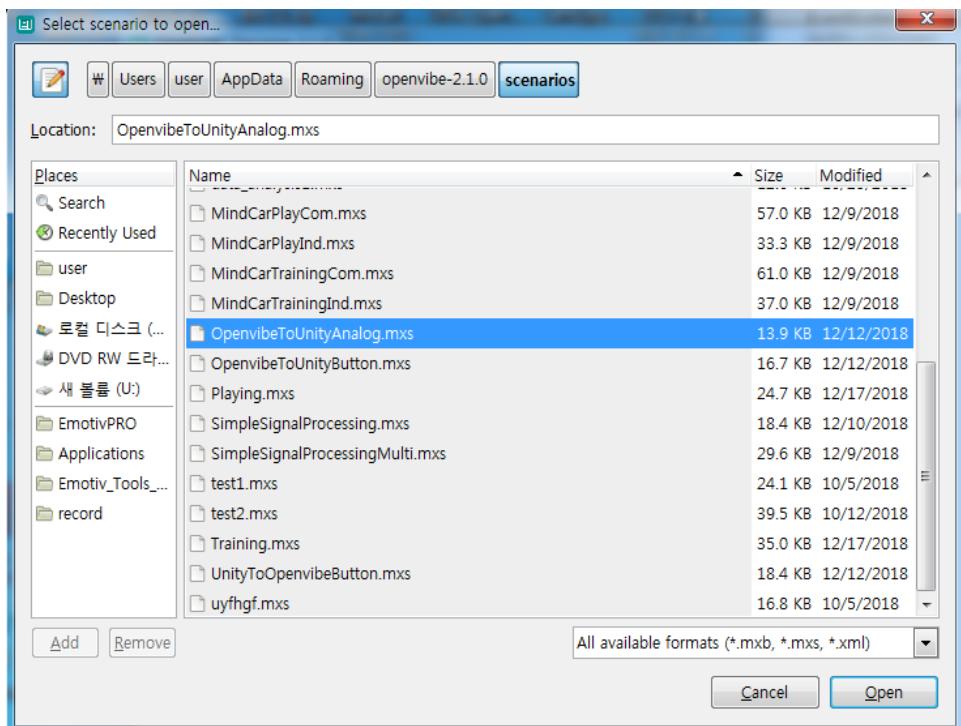


그림 20 Sernario file open 창

4. Openvibe acquisition server 설정 완료 후 Play (3.1 참고)

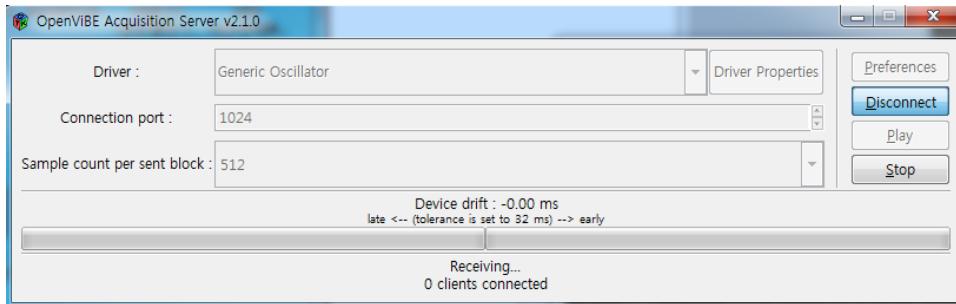


그림 21 Openvibe acquisition server play 중 화면

5. Openvibe designer에서 현재 시나리오 파일 설정 완료 후 (3.2~3 참고), 실행

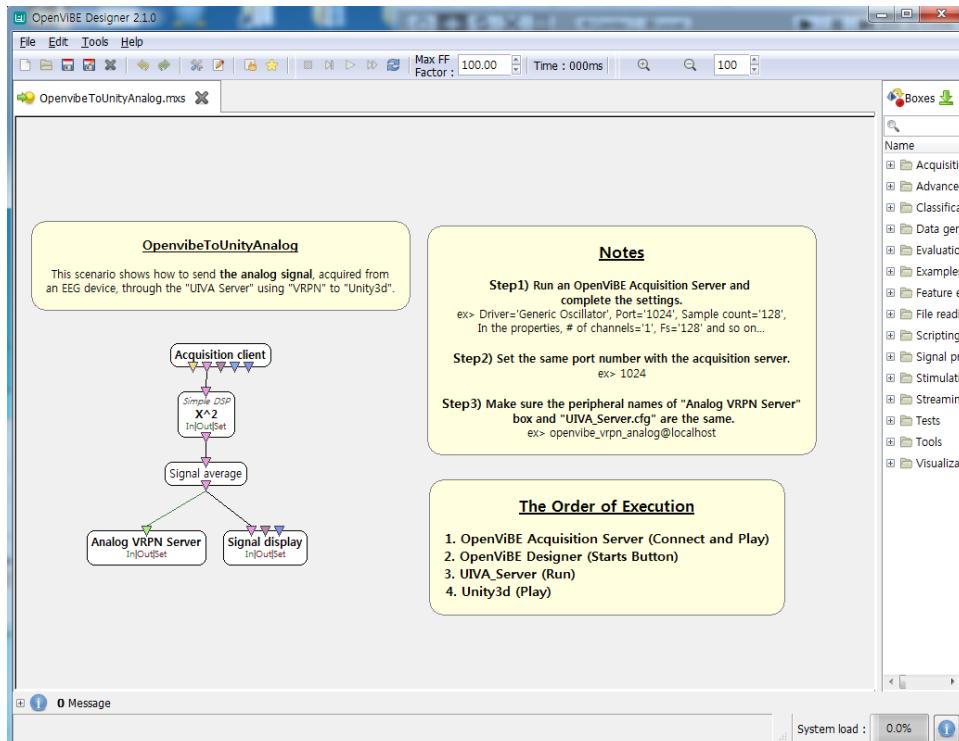
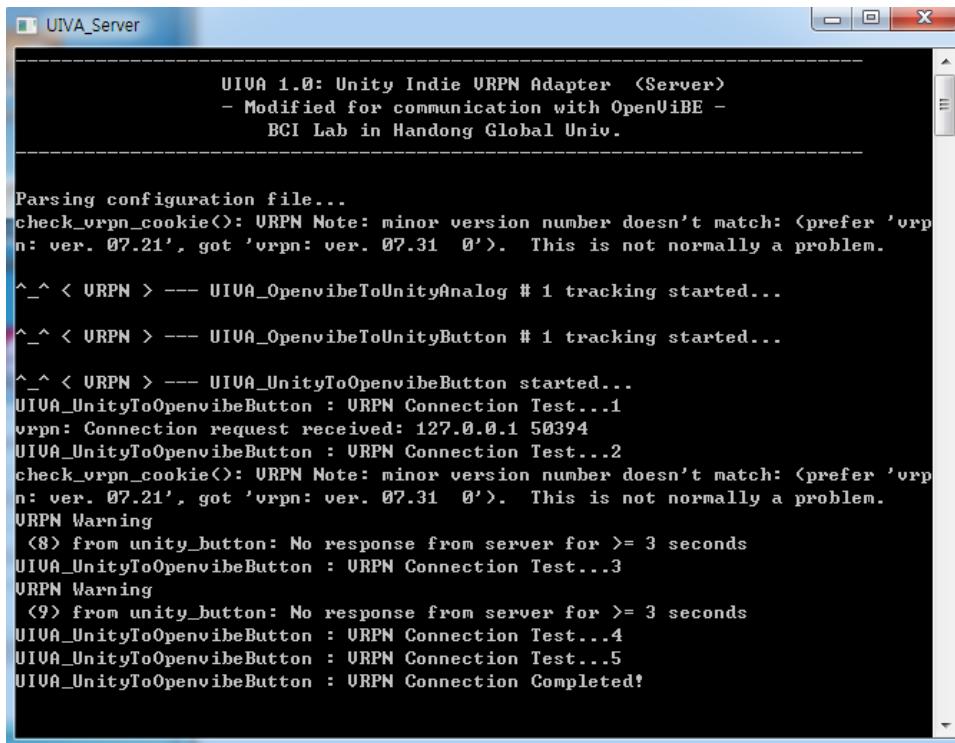


그림 22 Openvibe designer 내 시나리오 파일이 open 된 화면

6. UIVA_Server.exe 실행 후 'VRPN Connection completed!' 확인



The screenshot shows a Windows application window titled "UIVA_Server". The window contains a black text area displaying log messages from the application. The messages include:

```
UIVA 1.0: Unity Indie URPN Adapter <Server>
- Modified for communication with OpenViBE -
BCI Lab in Handong Global Univ.

Parsing configuration file...
check_vrpn_cookie(): VRPN Note: minor version number doesn't match: (prefer 'vrpn: ver. 07.21', got 'vrpn: ver. 07.31 0'). This is not normally a problem.

^_< VRPN > --- UIVIA_OpenvibeToUnityAnalog # 1 tracking started...
^_< VRPN > --- UIVIA_OpenvibeToUnityButton # 1 tracking started...

^_< VRPN > --- UIVIA_UnityToOpenvibeButton started...
UIVA_UnityToOpenvibeButton : VRPN Connection Test...1
vrpn: Connection request received: 127.0.0.1 50394
UIVA_UnityToOpenvibeButton : VRPN Connection Test...2
check_vrpn_cookie(): VRPN Note: minor version number doesn't match: (prefer 'vrpn: ver. 07.21', got 'vrpn: ver. 07.31 0'). This is not normally a problem.
VRPN Warning
<8> from unity_button: No response from server for >= 3 seconds
UIVA_UnityToOpenvibeButton : VRPN Connection Test...3
VRPN Warning
<9> from unity_button: No response from server for >= 3 seconds
UIVA_UnityToOpenvibeButton : VRPN Connection Test...4
UIVA_UnityToOpenvibeButton : VRPN Connection Test...5
UIVA_UnityToOpenvibeButton : VRPN Connection Completed!
```

그림 23 UIVA_Server.exe 실행 화면

7. Unity 파일 실행

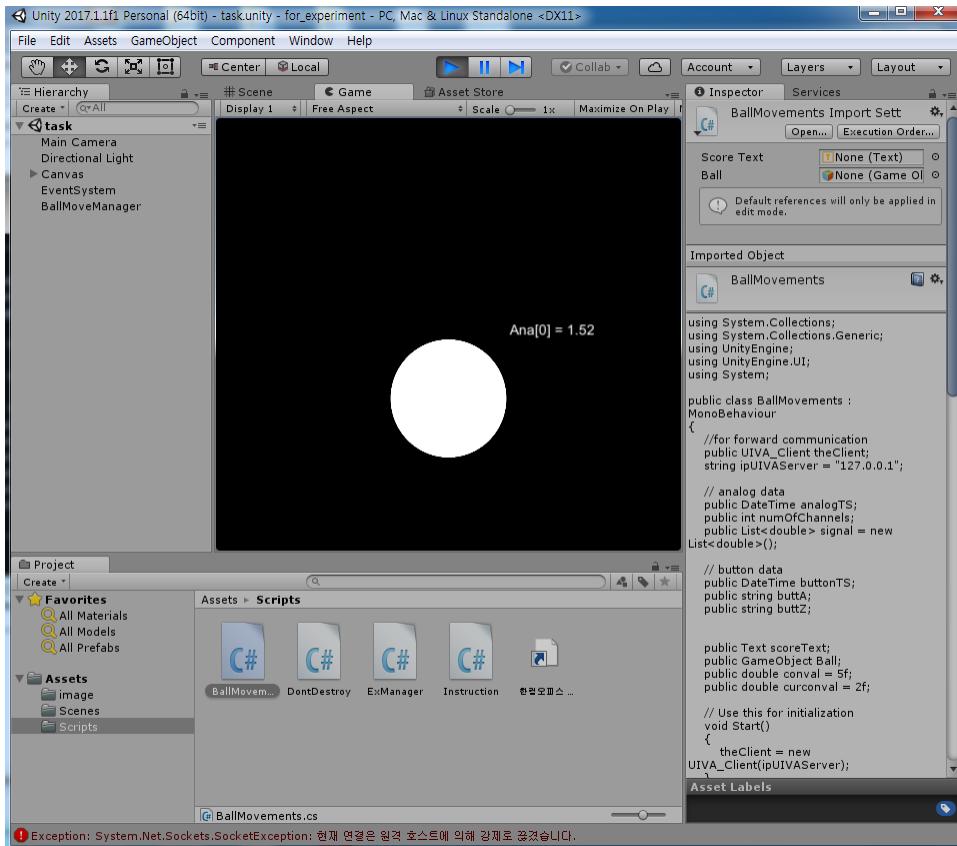


그림 24 Unity 예제 실행 화면

6 O2U로 개발된 뇌파 게임

다음은 O2U로 개발된 뇌파 게임입니다. 기술 상세를 하지 않고 간단한 게임 설명 후, 뇌파 처리과정에서 O2U의 Analog 와 Button 신호의 활용을 설명합니다.

6.1 MindCar 2.0

MindCar 2.0은 뇌파의 베타파를 분류하여 속도값으로 변환하여 달리는 자동차 게임입니다. 정방향 Analog 신호로 뇌파데이터를 실시간으로 Unity로 전해주고, 게임 시작, 게임 끝, 게임모드 지정에 따라 역방향 Button 신호를 사용하여 각각 상황에 맞게 OpenViBE 내 Python script에서 Button 신호를 받아 뇌파 데이터를 처리합니다.



그림 25 Unity MindCar 2.0 실행 화면

6.2 BCI World Tour

세계일주 뇌파 게임은 P300 feature를 유도하는 방식으로 진행됩니다. 우선 트레이닝 세션이 진행된 후, 온라인 세션이 진행되게 됩니다. 온라인 세션에서 여행자가 가고 싶은 도시를 유도하기 위해 각 대륙 그리고 각 도시 쪼ん에서 각각의 버튼들이 P300 뇌파를 유도하기 위해 일정 간격으로 깜빡입니다. 버튼들이 깜빡일 때 마다 O2U 는 역방향 Button 신호를 보내주면 OpenViBE에 Mark 되며, 이 때에 P300 신호 여부를 matlab에서 판단합니다. 모든 시도가 끝나면 가장 P300 신호 비율이 높았던 button 목적지를 지정되고, 목적지 소개 동영상이 실행됩니다.

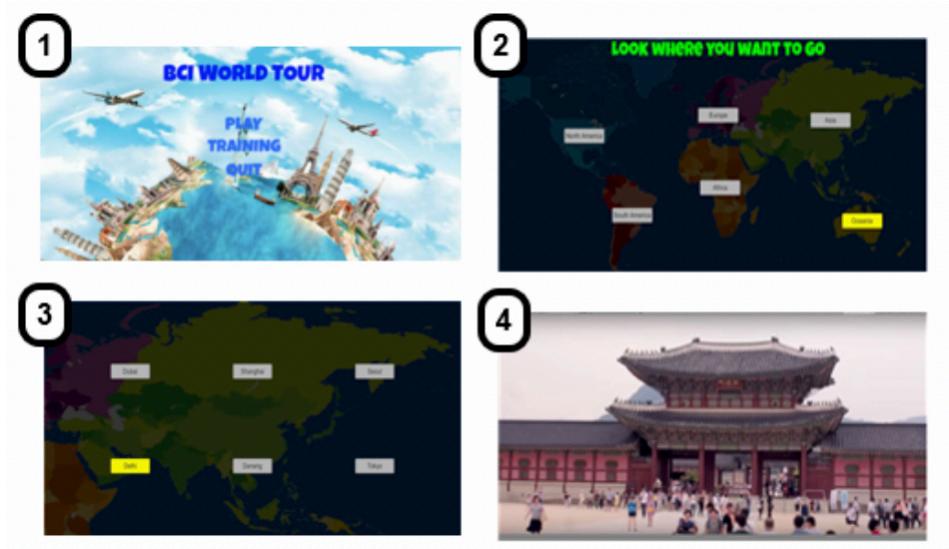


그림 25 Unity BCI WORLD TOUR 실행 화면

Contacts

안민규 minkyuahn@handong.edu

정성준 jdd01299@naver.com

김수용 dldl8675@google.com

이은민 21500534@handong.edu