

OpenSource Project

이창수 박지환 김진혁

2017년도 2학기

[프로젝트 수행 과정]

1. 오픈 소스 SW를 하나 선택
2. 선택한 오픈소스 SW의 사용 설명서를 LaTeX, 마크다운 중 하나를 선택하여 작성한다.
3. 각 조원들은 GitHub를 통해 SW사용 설명서 작성을 협업한다.
4. 설명서의 최종 버전은 12월 8일 자정까지 조장의 GitHub계정에 최종적으로 업로드한다.

[프로젝트 결과 채점 기준]

1. 오픈 소스 SW 선택의 이유 및 개요 (10)
2. 오픈 소스 SW에 대한 각 기능별 설명 (10)
3. 오픈 소스 SW 사용 설명에 대한 쉬운 이해가 가능한지 (10)
4. 사용 설명서 작성 중 발견한 오픈 소스 SW의 버그 및 원인 또는 사용 설명서 작성 중 느낀 기능 향상을 위한 제안 (10)
5. 각 팀원 별 협업을 증명할 자료 첨부(예: GitHub에 나와 있는 협업 히스토리) (20)

Contents

1 오픈 소스 SW 선택의 이유 및 개요	3
1.1 과제 설명	3
1.2 VRTK란?	3
1.3 VRTK 기능 개요	3
1.4 VRTK (VIVE) 권장사양	4
2 오픈 소스 SW에 대한 각 기능별 설명	5
2.1 Intro.unity	5
2.1.1 설명	5
2.1.2 예시	5
2.2 TouchPad Input	5
2.2.1 설명	5
2.2.2 예시	5
2.3 Grab Objects	5
2.3.1 설명	5
2.3.2 예시	6
2.4 Gaze UI	6
2.4.1 설명	6
2.4.2 예시	7
2.5 Sword Grab	7
2.5.1 설명	7
2.5.2 예시	10
3 오픈소스SW의 버그 및 원인 또는 기능 향상을 위한 제안	12
3.1 후기	12

Chapter 1

오픈 소스 SW 선택의 이유 및 개요

1.1 과제 설명

1. 오픈 소스 SW를 하나 선택
2. 선택한 오픈소스 SW의 사용 설명서를 LaTeX, 마크다운 중 하나를 선택하여 작성한다.
3. 각 조원들은 GitHub를 통해 SW사용 설명서 작성률 협업한다.
4. 설명서의 최종 버전은 12월 8일 자정까지 조장의 GitHub계정에 최종적으로 업로드한다.

1.2 VRTK란?

- VRTK는 Unity에서 VR연동시 VIVE컨트롤러 사용을 위한 툴킷으로 현재 저희 조원3명이 트랙제수업 VR/AR을 듣는 공통점에서 이 오픈소스를 선택하게되었습니다.오픈소스로 나와있는 VRTK는 VIVE연동시에 기초적인 기능들이 제작되어있으므로 이를 활용해서 VR프로젝트를 수행하는데 도움이 될 것이라 판단
- VRTK는 vive를 활용한 기초기능들의 예시 5가지

1.3 VRTK 기능 개요

- Unity에서 빠르고 쉽게 VR 프로젝트을 구축하는 데 도움이되는 유용한 스크립트 및 오브젝트 모음
- 가상 공간 내에서의 활동

- 객체 접촉, 잡기 및 사용과 같은 상호 작용
- 포인터 또는 터치를 통해 Unity3d UI 요소와 상호 작용
- 가상 공간 내의 물리엔진
- 버튼, 레버, 문, 서랍 등과 같은 2D 및 3D 컨트롤

1.4 VRTK (VIVE) 권장사양

- OS : 윈도우 7 SP1 이상의 운영체제
- CPU : 코어 i5-4590 / FX 8350
- 지포스 GTX 1060 / 라데온 RX 480
- 4GB 이상
- HDMI 1.3 포트 + USB 2.0 포트

Chapter 2

오픈 소스 SW에 대한 각 기능별 설명

2.1 Intro.unity

2.1.1 설명

- RGB색상별(0-255)수치를 조정하면 cube에 바뀐수치별로 색상이 변화하는 예제
- 바이브 컨트롤러를 사용하여 RGB색상별(0-255)수치를 조정하면 cube에 바뀐수치별로 색상이 변화하는 예제

2.1.2 예시

2.2 TouchPad Input

2.2.1 설명

- 두번째 예제샘플인 TouchPad Color Picker은 HTC Vive(SteamVR) 기기의 컨트롤러의 터치패드의 좌표값을 받아온후 연산을 하여 HSV 색깔좌표의 값으로 받아와서 RGB색으로 변환하여 오브젝트에 적용을 시킨다.
- VRTK의 컨트롤러 함수를 참조해서 터치패드 좌표 X,Y와 트리거 버튼을 가져온다.

2.2.2 예시

2.3 Grab Objects

2.3.1 설명

- 하나의 물체를 오브젝트로 선언.

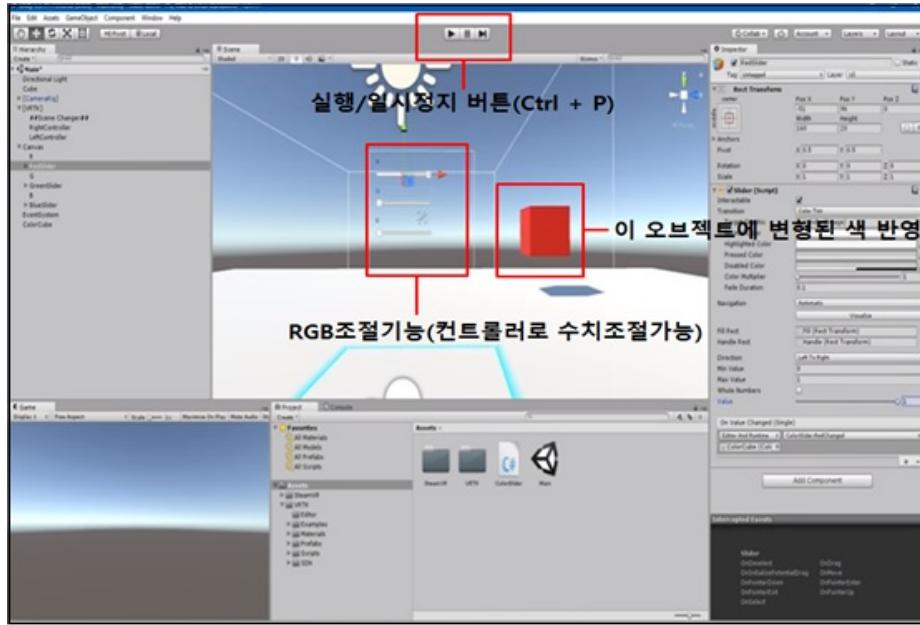


Figure 2.1: Intro 예시

- 레이캐스트와 콜리더를 통해 컨트롤러와 선언한 오브젝트의 충돌을 감지
- 충돌 후 컨트롤러 트리거 작동 시, 해당 물체를 컨트롤러로 잡은걸로 선언되어 컨트롤러가 움직이는대로 해당 오브젝트가 따라옴

2.3.2 예시

2.4 Gaze UI

2.4.1 설명

- 시선 처리에 따른 UI 설명
- 카메라 리그의 자식오브젝트 헤드 / 헤드의 자식오브젝트 아이의 시점을 시선처리로 설정.
- 대상은 캔버스 오브젝트의 자식오브젝트 버튼을 생성.
- 버튼 자식오브젝트 텍스트 생성 후
- 시선이 대상에 닿으면 색상이 변경되는 것을 확인할 수 있다.

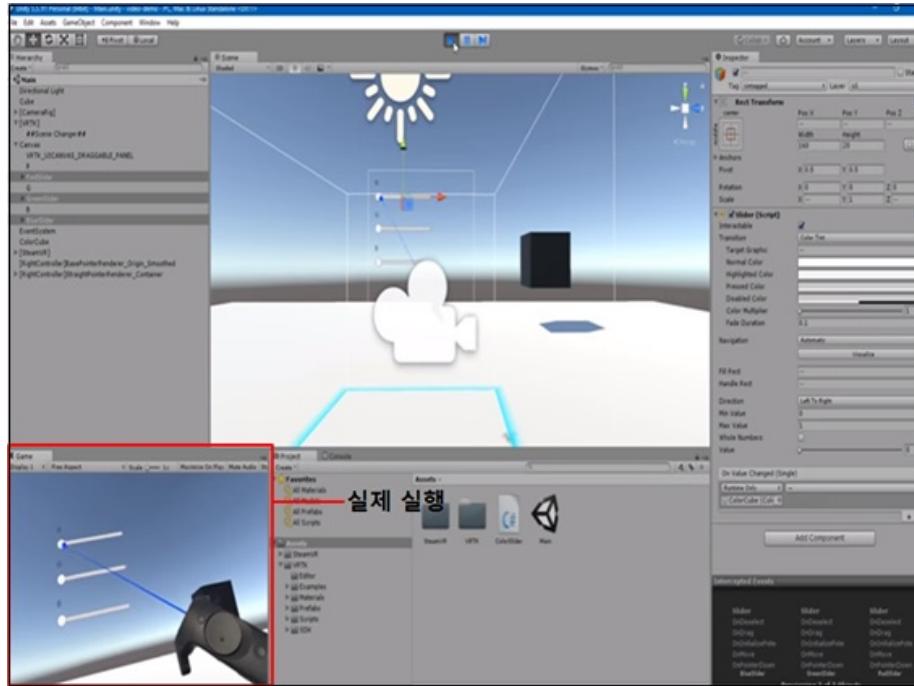


Figure 2.2: Intro 예시

```
// Use this for initialization
void Start () {
    if (GetComponent<VRTK_ControllerEvents>() == null) {
        Debug.LogError("ColorWheel is required to be attached to a Controller that has the VRTK_ControllerEvents script attached to it");
        return;
    }
    blackWheel = transform.Find ("CanvasHolder/Canvas/BlackWheel").gameObject;
    GetComponent<VRTK_ControllerEvent>().TouchpadAxisChanged += new ControllerInteractionEventHandler(DoTouchpadAxisChanged);
}

private void DoTouchpadAxisChanged(object sender, ControllerInteractionEventArgs e) {
    if (GetComponent<VRTK_ControllerEvents> ().triggerPressed) {
        ChangedValue (e.touchpadAxis);
    } else {
        ChangedHueSaturation (e.touchpadAxis, e.touchpadAngle);
    }
}

private void ChangedValue(Vector2 touchpadaxis) {
    Debug.Log ("ChangeValue: Trackpad axis at: " + touchpadaxis);

    this.value = (touchpadaxis.y + 1) / 2;
    Color currColor = blackWheel.GetComponent<Image> ().color;
    currColor.a = 1 - this.value;
    blackWheel.GetComponent<Image> ().color = currColor;

    UpdateColor ();
}
```

Figure 2.3: TouchPad 설명

2.4.2 예시

2.5 Sword Grab

2.5.1 설명

- 컨트롤러에 SWORD 오브젝트를 붙이고 펠수 있는 기능

```

private void ChangeHueSaturation(Vector2 touchpadAxis, float touchpadAngle) {
    float normalAngle = touchpadAngle - 90;
    if (normalAngle < 0)
        normalAngle = 360 + normalAngle;

    Debug.Log ("ChangeColor: Trackpad axis at: " + touchpadAxis + " (" + normalAngle + " degrees)");

    float rads = normalAngle * Mathf.PI / 180;
    float maxX = Mathf.Cos (rads);
    float maxY = Mathf.Sin (rads);

    float currX = touchpadAxis.x;
    float currY = touchpadAxis.y;

    float percentX = Mathf.Abs (currX / maxX);
    float percentY = Mathf.Abs (currY / maxY);

    this.hue = normalAngle / 360.0f;
    this.saturation = (percentX + percentY) / 2;

    UpdateColor ();
}

private void UpdateColor() {
    Color color = Color.HSVToRGB(this.hue, this.saturation, this.value);
    colorCube.GetComponent<Renderer>().material.color = color;
}

```

Figure 2.4: TouchPad설명

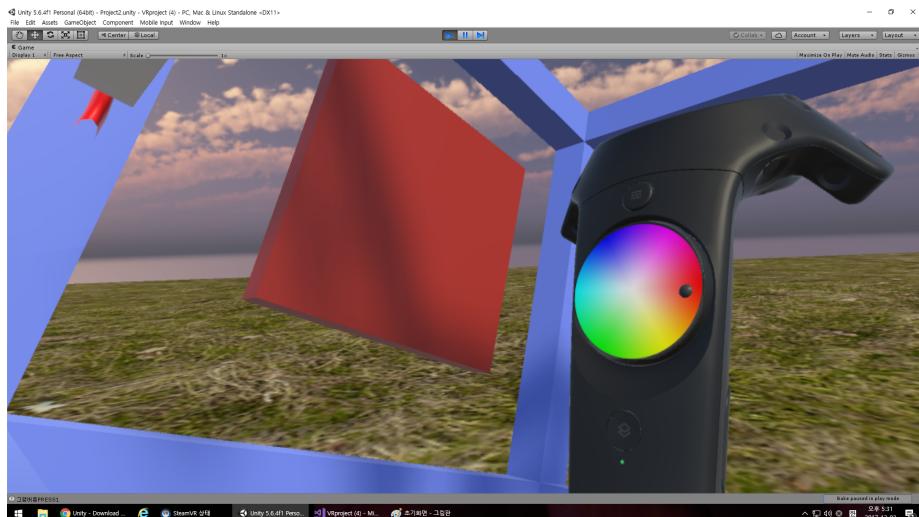


Figure 2.5: TouchPad예시

- HTC VIVE의 CameraRig에 각각의 left 와 right의 Controller로 기본 prefab 으로 있음
- VIVE의 CameraRig의 Controller를 VRTK의 Controller로 사용

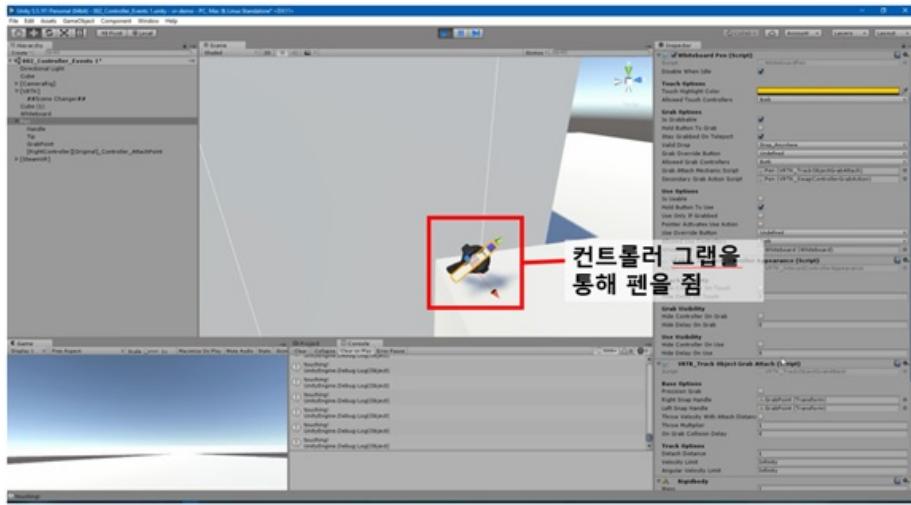


Figure 2.6: Grab 예시

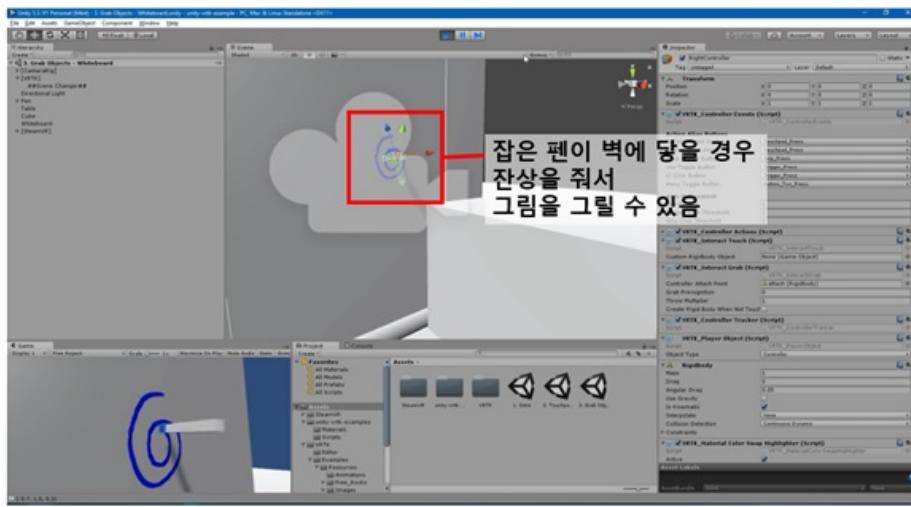


Figure 2.7: Grab 예시

- VRTK의 VRTK Interact Touch와 VRTK Interact Grab을 사용
- CreateEmpty 를 생성해서 Headset으로 주고, Transfrom Follow로 주고, Foller를 eye로 설정
- Backpack에 OnTriggerStay를 주고, InteractGrabdmf Collider로 줘서 사용
- Sword라는 스크립트를 추가하여, 이후 Controller graph에 Sword 생성

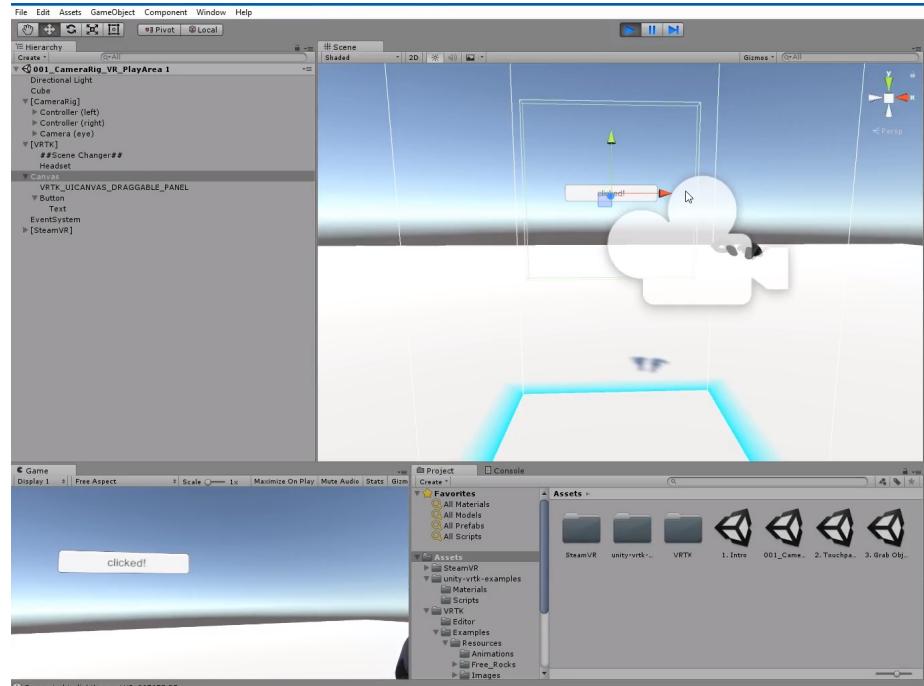


Figure 2.8: GazeUI 예시

- Grab해제 시 해당 Sword prefab은 rigidbody에 의해 땅으로 떨어지며, 다시 grab시 새로운 sword clone prefab이 생성

2.5.2 예시

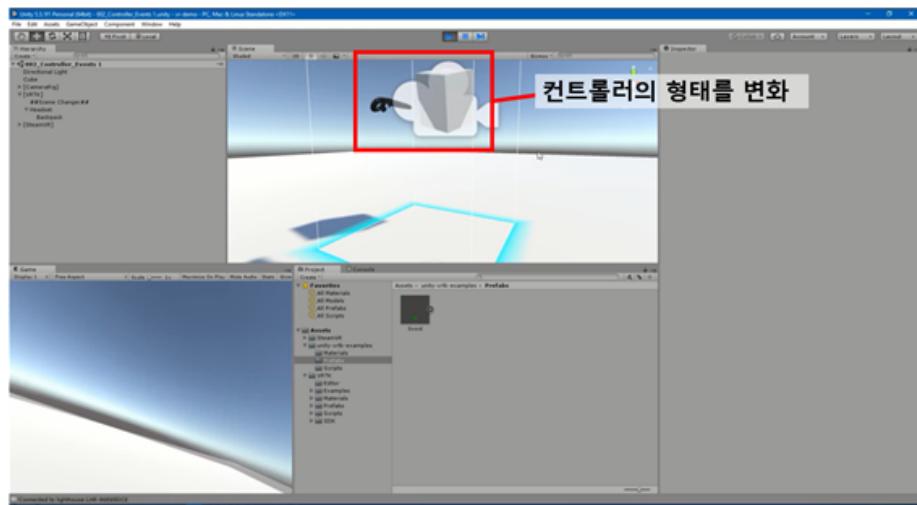


Figure 2.9: SwordGrab 예시

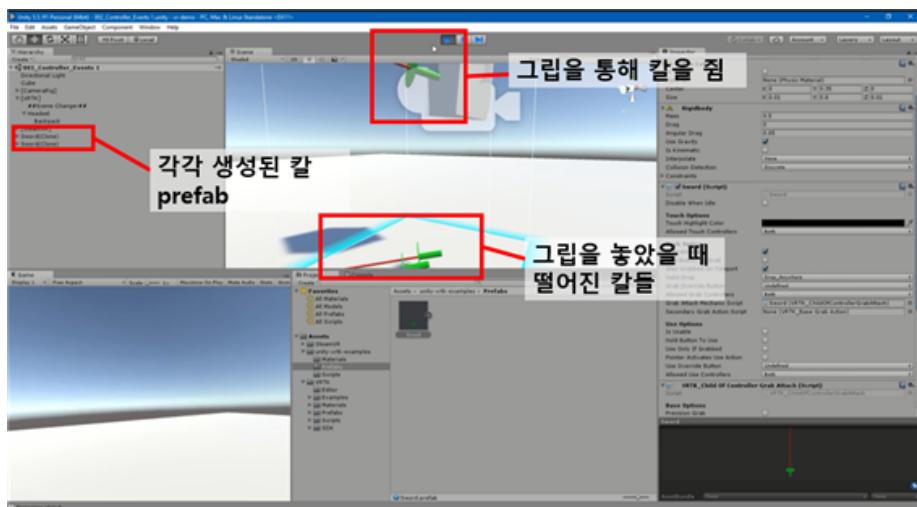


Figure 2.10: SwordGrab 예시

Chapter 3

오픈소스SW의 버그 및 원인 또는 기능 향상을 위한 제안

3.1 후기

- 20104834 이창수(조장ID)
 - 바이브 컨트롤러를 사용한 모델링 작업을 수행할 프로그램을 제작가능할 수 있다.
 - 이를 통해 멀티플레이어 작업을 연동한다면 다양한 분야의 모델링 작업을 협업할 수 있다.
 - ex) 의료분야 자동차 디자인분야 건축분야 등등
- 20124859 박지환(GitHub:genari7)
 - VIVE를 사용하여 프로젝트를 만드는데 있어서 유용한 기능들을 모아놓아서 유용하다.
 - 연구실을 찾아가서 따라 만든 파일을 직접 시연해보니 좋았다.
 - 협업을 위한 GITHUB와 레포트를 편하게 만들수 있는 LATEX의 사용법을 공부해보니 앞으로 유용하게 쓰일거 같다.
- 20134833 김진혁(branch:hoxfix2)
 - SteamVR에서 걸어놓은 제약조건을 쉽게 해결할 수 있었다.
 - 다양한 프리팹과 예시들이 있어서, 원하는 것을 조금 더 쉽게 제작할 수 있었다.
 - VR에서 우리가 생각하는 것보다 다양한 기능을 구현할 수 있었으며, 이와 관련된 여러 작품들이 많이 나와있다는 것을 알 수 있었다.