VRUI

# 구조

- Unity Vive Utility 부분을 교체 함으로써 다양한 기기 대응 가능

Unity Vive Utility

모든 VR 기기

대응 모듈

<Button Base>

<Drag Base>

<Display Base>

<Base Class 구조>

이후에 모듈 교체를 위한 warp Class의 역할도 함

**<VR 액션을 기준으로 베이스 클래스 제작>**

# Class – ButtonBase

- HTC.UnityPlugin.ColliderEvent를 Warping 한다.

- 버튼에 대한 기본 이벤트 정의, 가상함수로 제공.

- 버튼에 대한 공동 부분을 정의.

HTC.UnityPlugin.ColliderEvent

- IColliderEventPressDownHandler

- IColliderEventPressUpHandler

- IColliderEventPressEnterHandler

- IColliderEventPressExitHandler

- Define Property

- ControllerButton : Vive 컨트롤러의 버튼 지정

- buttonDownDisplacement : 버튼의 깊이

- UnityEventController : 유니티 이벤트 지정

- OriginPosition : UI 처음 위치 값 저장

- HashSet<ColliderButtonEventData> : 이벤트 해쉬

# Class – dRAGBASE

- HTC.UnityPlugin.ColliderEvent를 Warping 한다.

- Drag Controll에 대한 기본 이벤트를 정의 하고 가상함수 로 제공.- 버튼에 대한 공동 부분을 정의.

- Drag Action과 Property 정의.

- UI Class 제작 시 수정 및 추가

- HTC.UnityPlugin.ColliderEvent

- IColliderEventDragStartHandler

- IColliderEventDragFixedUpdateHandler

- IColliderEventDragUpdateHandler

- IColliderEventDragEndHandler

- Define Property

- ControllerButton : Vive 컨트롤러의 버튼 지정

- buttonDownDisplacement : 버튼의 깊이

- UnityEventController : 유니티 이벤트 지정

- OriginPosition : UI 처음 위치 값 저장

- HashSet<ColliderButtonEventData> : 이벤트 해쉬

- MIN\_FOLLOWING\_DURATION

- DEFAULT\_FOLLOWING\_DURATION

- MAX\_FOLLOWING\_DURATION

- Transform HandleObject

- FollowingDuration

# Class – dISPLAYBASE

- VR(3D)상에의 정보표시 및 VR 장비에 Display관련 된 UI를 위한

BaseClass 정의.

- 이후 HMD에 대한 고려 위해 미리 구조를 잡아보자

- VR상에서 2D Interface에 대한 고려.

- UGUI의 지원 보다 심플하게 지원하도록 고려.

- VR UI 제작 시 다중 상속으로 보다 편하게 표시 기능을 제공.

- Define Property

- TextBox : Display 3d Text.



# push button switch

* Unity Event 정의
  + UnityEventButton Button\_Down
  + UnityEventButton Button\_Up
  + UnityEventButton Button\_Press

ButtonBase

3DVRUI\_PushButton

<Class Diagram>

* 모델링의 컬러 및 재질 수정

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

* Unity Event 정의 Unity Interface

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

# Rocker switch

* Unity Event 정의
  + UnityEventButton Switch\_On
  + UnityEventButton Switch\_Off

- 모델링의 컬러 및 Matrial 수정 가능 (Unity Editor)하게 함.

- 서로 따른 버튼 오브젝트를 하나처럼 보이게 처리

- 중간 정도에 값에서 버튼을 Release 시킬 때 처리  
(많이 기울어 진 쪽으로 처리 함)

ButtonBase

3DVRUI\_Rocker

<Class Diagram>

* Unity Event 정의 Unity Interface

스크린샷이(가) 표시된 사진

매우 높은 신뢰도로 생성된 설명

# slide switch

- On/Off 의 중간 값은 없다 밀다가 놓는 순간 많이 기울어 진 쪽으로 값을 설정한다.

- 중간 값은 버튼 자체가 불량이다.(값 자체는 On/Off 존재하게 만든다.)

- 버튼의 각종 감도 등 설정 값을 설정.

ButtonBase

3DVRUI\_Slide

<Class Diagram>

* Unity 오브젝트 설정 Unity Interface

전자기기, 도로이(가) 표시된 사진

높은 신뢰도로 생성된 설명

# toggle switch

- On/Off 의 중간 값은 없음 놓는 순간 많이 기울어 진 쪽으로 값을 설정.

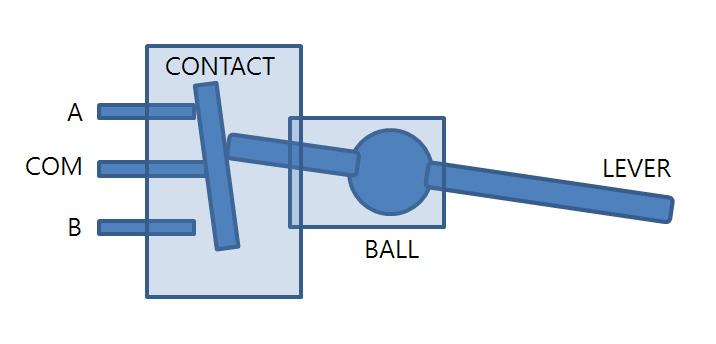
- 값 자체는 On/Off 존재하게 만든다.

- 버튼의 각종 감도 등 설정 값을 설정.

ButtonBase

3DVRUI\_Toggle

<Class Diagram>



동작의 원리

- 밀조작 손잡이 LEVER를 아래위로 제끼면 BALL을 중심으로 시소가 작동되어

CONTACT가 다시 시소동작을 해서 그림과 같이 A와 COM이 붙었다가 반대로

제끼면 B와 COM이 붙었다 하는 구조.

- DPDT는 LEVER의 반대편이 그냥 막대가 아니라 Y구조로 되어서 그림상 앞뒤로 왼쪽 부위가 2벌이 있어서 동일한 2개 회로가 연동하는 구조입니다. 3PDT는 삼지창 형태로 만들고 접점회로를 3벌을 만들어서 3개 회로가 연동

# digital display

- 자동 카운트 기능을 넣어 카운터를 가능.

- 카운트가 끝나면 이벤트 처리.

- 하나를 만들어 조합해서 몇 자리를 표현.

DisplayBase

3DVRUI\_DigitalDisplay

<Class Diagram>

* 숫자 이벤트 설정 Unity Interface



# progress display

- 진행도를 표시한다.

- 0.0 ~1.0 또는 %로 값을 받을 수 있다.

DisplayBase

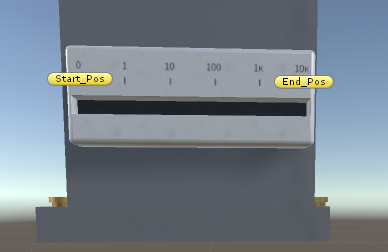
3DVRUI\_Progress

<Class Diagram>

- 값을 비율로 계산 해준다.

하늘이(가) 표시된 사진

높은 신뢰도로 생성된 설명



# cam switch

- 컨트롤러를 잡고 손목 회전으로 작동.

- 실제 동작 할 때와 비슷하게 회전 계산.

- 회전 속도와 손잡이의 민감도 조절 가능.

- 샤프트 회전의 위치에 따른 이벤트 처리.

- 샤프트의 중간에 놓을 때 회전 값에 따라 하나를 선택.

DisplayBase

3DVRUI\_CamSwitch

<Class Diagram>

* 회전 속도와 손잡이 민감도 설정 Unity Interface



# pressure display

- 탄성 압력계 타입.

- 부가적으로 수치를 표시

- 압력이 내려 갈 때도 자연스럽게 보이게 하자

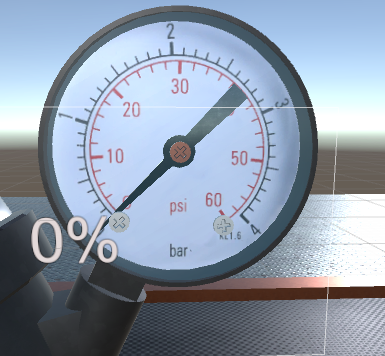
- 압력이 내려가는 도중에 다시 올라 갈수 있게 처리.

- 수치의 증가는 이벤트에서 처리.

DisplayBase

3DVRUI\_PressureGauge

<Class Diagram>



# micro switch

- 레버를 눌렀다 때는 순간 on 됨

- 사용 시에는 몸체를 원하는 곳에 배치하고 충돌 오브젝트를 이용하여 레버에 압력을 가하는 형태에 실제 사용과 비슷하게 VR상에서도 사용

- 실제로도 많이 사용되지만 3D공간 상에도 많은 곳에서 동작을 시키거나 동작센서로 사용가능.

DisplayBase

3DVRUI\_Micro

<Class Diagram>



# monolever switch

- 실제 레버의 3가지 운동을 구현(계속, 자동복귀,봉쇄 시행) 이벤트로 처리하자.

- 레버의 각종 설정을 할 수 있게 Property를 설정..

- 실제와 같은 조작을 중심으로 놓으면 가운데로 가고, 방향 유지를 하면 계속 실행.

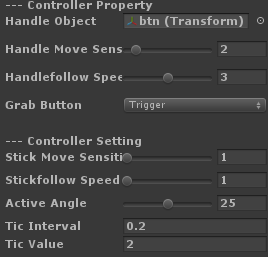
- 고무 부싱도 모델링으로 제작.

DragBase

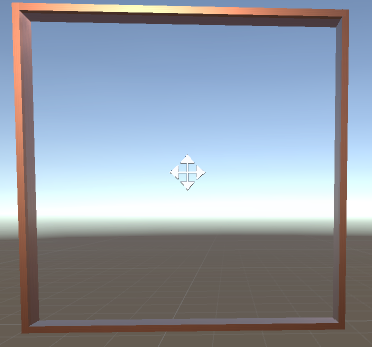
3DVRUI\_MonoLever

<Class Diagram>

* 속도 조절 설정 Unity Interface



테스트용 오브젝트 방향에 따라 화살표 이동하게 제작



# knife switch

- 퓨저도 제작하자 (프로그램에서는 플레그로) 조건으로 사용 가능.

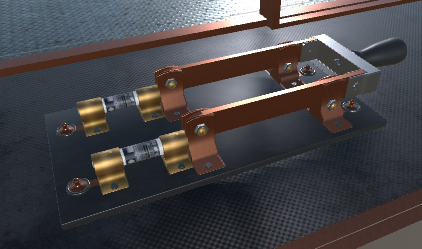
- 가상에서는 많이 사용 될 듯하다. 실제에는 감전이나 기타 위험 때문에 사용하지 않지만,

가상에서는 정확도와 동작으로 제어 가능한 컨트롤러로서 유용.

DragBase

3DVRUI\_Knife

<Class Diagram>



2개의 조건으로 퓨저 사용 가능  
특정 값이 넘어가면 퓨저가 끊어 동작 못하게 할 수 있음

# Timer

- 순시동작 순시복귀: 전원이 들어가면 즉시 동작하고 전원이 내려가면 즉시 복귀하는 접점

- 순시동작 한시복귀 : 전원이 들어가면 즉시 동작하고 전원이 내려가면 세팅 된 시간후에 복귀하는

접점

- 한시동작 순시복귀: 전원이 들어가면 세팅한 시간만큼 지연한 후 동작하고 전원이 내려가면

즉시 복귀하는 접점

- 한시동작 한시복귀: 전원이 들어가면 지연 동작하고 전원이 내려가도 지연 복귀하는 접점

DragBase

3DVRUI\_Timer

<Class Diagram>



- 시간을 자유롭게 인자값으로 설정.

- 시간 설정은 회전값으로 설정

- 시간을 표시하기에 모델링으로 직접하기에는 부족하여 보조적으로 텍스로 표시.

# toggleonoff switch

- 일반 푸쉬 버튼을 토글 기능을 가지는 경우

- 자동차, 세탁기 버튼 등 다양향 형태의 다양재질로 제작.

- LED를 이용한 직관성이 높은 형태로 제작.

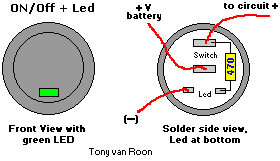
- 토클 버튼 푸쉬 버튼형식으로 변형한 형태.

- On/Off표시가 자체적으로 된다는 것.

ButtonBase

3DVRUI\_ToggleOnOff

<Class Diagram>



# foot switch

- 커버속에 압력 입력 버튼이 핵심

- 압력의 강도 조절 입력

- 압력이 해제 되면 다시 처음 상태로 돌아옴

- 압력이 최대치면 더 이상 이상의 입력은 없음.

- 회전, 리벳팅 같은 어플리케이션에서 기계 작동에 이상적

- 더블 페달은 순간 또는 유지 된 자유 동작 또는 트립 스타일을 포함한 다양한 동작을 제공

DragBase

3DVRUI\_FootSwitch

<Class Diagram>



- 풋 컨트롤러의 입력을 받는 VR 장비를 가지고 있지 않기 때문에 입력은 버튼 입력 받는다.

- 입력의 단계를 숫자로 표시 해주자

# Hoist switch

- 보통 레일을 올리거나 내릴 때 많이 사용하는 스위치, 비상 버튼을 포함한다.

- 버튼은 기능에 따라 여러 개가 존재하기도 한다.

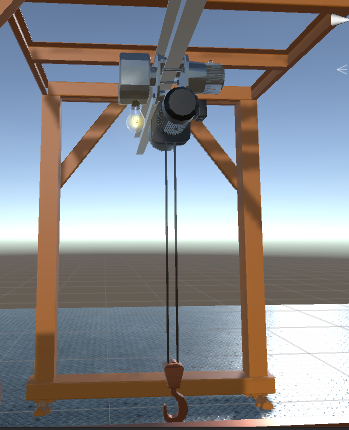
- 버튼은 레일의 모터와 연결되어 동작한다.

- 과부하 및 전압 차기가 내장 된 임베디드 타입 및 노출타입

DragBase

3DVRUI\_Hoist

<Class Diagram>



- 테스트 할 수 있는 모터형의 레일을 제작

- 버튼에 입력 방향을 알 수 있게 문자를 표기.

- 버튼을 누르고 있으면 값이 계속 전달되고 때면 정지.

- 비상 버튼은 그냥 경고 메시지를 출력.

# magneticreader / card

- 카드의 자성 재료 밴드 위의 작은 철계 자성 입자의 자력을 수정하여 데이터를 저장 할 수 있는

유형

- 스와이프카드 또는 마크네 스파라이프라고 한다.

- 판독은 자기판독 헤드를 지나쳐 읽음 트래스폰더 장치 및 마이크로 칩을 포함 할 수 있다.

DragBase

3DVRUI\_MagenticReader

<Class Diagram>

DragBase

3DVRUI\_MagneticCard

<Class Diagram>

****

- 카드와 리더기에 각각 값을 입력 할 수 있게 한다.

- 리더기와 카드에 값이 일치하면 True 아니면 False 이벤트를 발생

- 클래스 구성은 리더기와 카드 각각으로 구성 한다.

- 실감을 주기 위해 리더기와 카드의 충동 체크 한다.

# handwheel

- 회전의 속도 조절을 가능.

- 방향에 따른 이벤트 처리

- 이벤트에 값을 넣어 증가시키거나 감소 가능.

- 다양하게 활용하게 이벤트에서 주로 처리.

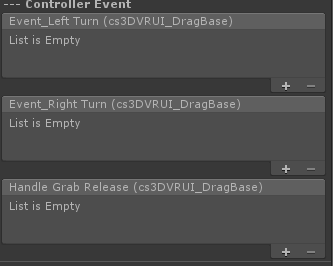
- 무한 회전.

DragBase

3DVRUI\_HandWheel

<Class Diagram>

* 상태 별 이벤트 설정 Unity Interface



# TApe measure

- 줄자 오브젝트는 누르면 줄자 모드로 설정되는 형태

- 줄자 오브젝트는 그 자리에 있는다. 버튼 같은 역할로 쓴다  
- 오브젝트 클릭 후 빈 공간에 클릭을 하면 측정을 시작한다.

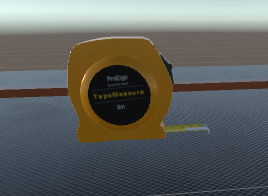
- 클릭 후 길이를 늘리면 숫자로 표시하고 측정이 완료 된 끝에 값을 표시한다.

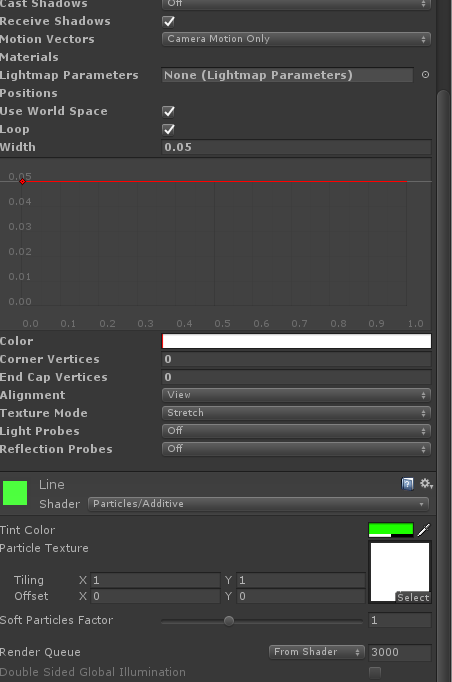
- 줄자의 선의 모양을 편집 가능(유니티 에서)

DragBase

3DVRUI\_TapeMeasur

<Class Diagram>





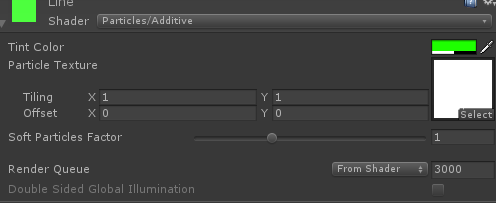
# laser switch

- 레이저 포인터와 오브젝트 충돌 시에 On 충돌 이 없으면 Off처리

- On/Off 이벤트 처리

- 실제와 같은 재질로 만든다.

- 레이저를 수정 할 수 있게 재질과 색을 수정 가능



DragBase

3DVRUI\_LaserSwitch

<Class Diagram>

# gntry crane control box

- 거대한 크레인을 컨트롤 할 수 있는 크레인 컨트롤 박스를 제작.

- 좌우 다리로 구성 되어있다 고톤수와 고양정 장 스판의 크레인의 경우는 한지 레그

고정레그로 구성

- 크레인은 동력을 사용하여 하물을 달아 올려 수평으로 운반한다

- 구조에 따라 호이스트식, 트래브식, 로프 트롤리식 등이 있다

- 권상장치는 크레인에서 하물을 들어 오리거나 내리 위한 장치

(권상전동기, 축 이음부, 브레이크, 치차, 트럼, 시브, 훅크 블록 및 와이어로프 구성)

- 컨트롤 박스를 (컨트롤 타워에) 설치 한 형태로 구성한다

- 초기는 컨트롤로 박스 형태는 기능 중심으로 배치 후 수정 하자

- 박스는 인접하면 접근하게 하자



**Test Model Data**

* 실제와 같은 모델로 구성
* 애니메이션 일부 구현

