

캡스톤 디자인 II 종합설계 프로젝트

프로젝트 명	MYTH(Making Your Things Haptic)	
팀 명	미네르바	
문서 제목	결과보고서	

Version	1.5
Date	2015-NOV-26

	이광헌 (조장)	
	전새미	
팀원	정하엽	
	최여진	
	최준환	

캡스톤 디자인 Ⅱ Page 1 of 38 **결과보고서**



결과보고서			
프로젝트 명	MYTH(Making Your	Things Haptic)	
팀명	미네르바		
Confidential Restricted	Version 1.5	2015-NOV-26	

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "MYTH(Making Your Things Haptic)"를 수행하는 팀 "미네르바"의 팀 원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "미네르바"의 팀원들의 서면 허락 없이 사용되거나, 재 가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	결과보고서_미네르바.doc
원안작성자	이광헌,최준환,전새미,최여진,정하엽
수정작업자	이광헌,최준환,전새미,최여진,정하엽

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2015-11-25	전새미	1.0	최초 작성	최초 내용 작성
2015-11-25	정하엽	1.1	내용 수정	수정된 다이어그램 작성
2015-11-26	최준환	1.2	내용 수정	테스트 케이스 작성
2015-11-26	최여진	1.3	내용 수정	프로젝트 목표 수정
2015-11-26	전새미	1.4	문서 편집	서식 편집
2015-11-26	이광헌	1.5	최종 수정	최종 수정



결과보고서			
프로젝트 명	MYTH(Making You	Things Haptic)	
팀 명	미네르바		
Confidential Restricted	Version 1.5	2015-NOV-26	

목 차

1	개요		4
	1.1 프로	젝트 개요	4
	1.2 추진	배경 및 필요성	5
	1.2.1	문제점 1	5
	1.2.2	문제점 2	6
	1.2.3	문제점 3	7
	1.3 현재	있는 터치 스크린들의 한계	8
	1.3.1	빅 노트	8
	1.3.2	Surface Hub	9
2	개발 내용 및	및 결과물	10
	2.1 목표.		10
	2.2 연구/	/개발 내용 및 결과물	11
	2.2.1	연구/개발 내용	11
	2.2.2	시스템 기능 요구사항	16
	하드웨이	ሃ	16
	기능 요	.구사항	18
	2.2.3	시스템 비 기능(품질) 요구사항	18
	2.2.4	시스템 구조 및 설계도	19
	2.2.5	실제 툴 바의 UI	20
	2.2.6	UI 실행 창	21
	2.2.7	이벤트 전송 과정	23
	2.2.8	활용/개발된 기술	24
	Touch A	Algorithm	25
	Catmul	ll-rom splines	25
	센서 각	도에 따른 터치 범위 보정	26
	화면 캘	리브레이션의 간편화	26
	2.2.9	현실적 제한 요소 및 그 해결 방안	26
	하드웨이		26
	소프트웨	웨어	27
	2.2.10	결과물 목록	29
3	자기평가		33
	3.1 사용	가능성	33
4	참고 문헌		33
5	부록		34
	5.1 사용	자 매뉴얼	34
	5.2 테스!	트 케이스	35



결과보고서			
프로젝트 명	MYTH(Making Your	Things Haptic)	
팀명	미네르바		
Confidential Restricted	Version 1.5	2015-NOV-26	

1 개요

1.1 프로젝트 개요

이 프로젝트는 사용자가 스크린이 아닌 벽에 빔 프로젝터 이미지를 송출하여 화면을 터치하거나, 터치 스크린이 아닌 화면을 터치할 수 있는 기능을 제공하는 프로젝트이다. 사용자의 편의를 위하여 어느 상황에서나 사용이 가능한 툴 바의 기능을 제공한다. 키넥트를 이용한 터치 알고리즘은 모듈로써 프로그램 을 실행시키면 화면 지정 후 자동적으로 적용이 되고, 프로젝트가 제공하는 툴 바로 여러 가지 기능들을 이 전보다 쉽고 간편하게 사용할 수 있다.

우리 프로젝트는 키넥트의 적외선 센서를 이용해 깊이 정보를 읽는다. 그 후 빔 프로젝터가 쏜 화면 혹은 모니터(통칭 화면 출력 장치)의 깊이 값을 구해 저장한 뒤 사용자 손의 깊이 값과 비교하여 접촉을 판별해 이벤트를 발생시킨다. 또한 프로그램에서 제공하는 툴을 이용하여 실행중인 화면에 간단한 메모 혹은 낙서를 할 수 있고 그 화면을 저장할 수 있다. 키보드를 제공하지 않는 터치스크린을 위해 가상키보드를 제공한다.



결과보고서			
프로젝트 명	MYTH(Making You	Things Haptic)	
팀명	미네르바		
Confidential Restricted	Version 1.5	2015-NOV-26	

1.2 추진 배경 및 필요성

최근 몇 년간 터치패널을 이용한 스마트 폰, 태블릿 pc 등이 활발히 보급되면서 터치 인터페이스를 채용하는 기기들이 많아졌고, 과거 공상 과학 영화에서나 볼 수 있었던 기술들이 개발되고 있다. 그에 따라 Graphical User Interface (이하 GUI)를 넘어서 터치스크린의 동작과 같은 자연스러운 상호작용에 의한 사용자인터페이스를 뜻하는 Natural User Interface (이하 NUI)가 각광받고 있다. 이 프로젝트는 그런 점을 착안하여키넥트를 이용해 기존에 사용하던 모니터나 빔 프로젝터를 터치스크린처럼 사용할 수 있게 한다.

1.2.1 문제점 1



[그림1, 그림2] 다수의 사용자가 사용하기에는 부적합한 스크린 크기

[그림 1], [그림 2]와 같이 다수의 사용자가 하나의 스크린을 사용할 때 스마트 폰, 혹은 태블릿 PC는 고정적으로 작은 스크린 사이즈를 가지고 있어 불편함을 겪을 수 있다. 여러 사용자가 한 화면에 터치를 할 때도 스크린 크기 때문에 입력이 정확히 인식되지 않을 때가 있고, 같은 화면을 보고 싶을 때 너무 작은 스크린의 크기는 부적합하다. 이를 보완하기 위하여 빔 프로젝터를 사용해 가변적인 스크린 사이즈를 제공한다.



결과보고서			
프로젝트 명	MYTH(Making You	Things Haptic)	
팀명	미네르바		
Confidential Restricted	Version 1.5	2015-NOV-26	

1.2.2 문제점 2



[그림3, 그림4] 프레젠테이션 원격 제어의 불편함

[그림 3], [그림 4]와 같이 회사나, 학교의 강연에서 프레젠테이션을 할 때 아래와 같은 불편 사항들이 존재한다.

- 슬라이드를 넘길 때, 발표를 하다가 PC로 이동해서 마우스를 클릭해야 한다.
- 특정한 텍스트를 설명할 때에는 레이저 빔을 쏴서 특정한 텍스트를 가리켜야 한다.
- 필기를 할 때에는 빔 프로젝터의 슬라이드가 아닌, 칠판에 해야 한다.
- 원격 제어를 하려면 프레젠테이션 리모콘과 같은 특별한 장비가 필요하다.

이러한 불편함을 해결하기 위하여 깊이 정보를 제공하는 키넥트 장비를 사용한다. 이 장비를 사용하여 벽의 깊이 정보를 최초에 인식한 후, 그 깊이 값과 사용자의 손가락이 같아지면 터치로 인식해 이벤트를 발생시키는 알고리즘을 구현한다. 이렇게 되면 특별한 장비 없이 프로젝터 슬라이드를 넘길 수 있고, 밑줄 과 같은 펜 기능이 PC로 이동하지 않고 사용이 가능해진다.



결과보고서			
프로젝트 명	MYTH(Making You	Things Haptic)	
팀명	미네르바		
Confidential Restricted	Version 1.5	2015-NOV-26	

1.2.3 문제점 3



현대아이티 스마트보드 70" 산업PC 테이블스탠드 E70BLHT [터치스크린 / FHD LED패널 / 미팅메 이트 /전면HDMI,USB입력단자]

9,481,000원 가격비교3

디지털/가전 > 모니터주변기기 > 모니터스탠드/암

네이버5%추가할인+신한카드결제시7%할인+카드무이자 3개월(롯데, 신한,KB)

등록일 2014.05 · 🕢 찜하기 0



현대아이티 스마트보드 70" 보급PC 이동스탠드 E70BLMM [터치스크린 / FHD LED패널 / 미팅메이 트 / 전면HDMI,USB입력단자]

9,481,000원 가격비교3

디지털/기전 > 모니터주변기기 > 모니터스탠드/암

네이버5%추가할인+신한카드결제시7%할인+카드무이자 3개윌(롯데, 신한,KB)

등록일 2014.05 · 🕢 찜하기 0

[그림 5] 대형 터치패널의 가격

[그림 5]와 같이 대형 터치패널을 이용하면 "문제점 1"에서 언급한 다양한 사용자들의 불편함과 "문제점 2"에서 언급한 프레젠테이션 원격 사용의 불편함도 어느 정도 해결이 가능하다. 하지만 대형 터치패널은 여러 사용자 혹은 단일 사용자의 상황에 따라 스크린크기의 조절이 불가능하고 크기가 매우 커서 휴대가 거의 불가능하다. 또한, "문제점 2"에서 언급한 슬라이드 넘기기는 가능하지만 필기와 같은 모듈은 사용할 수 없다. 게다가 현재 판매하고 있는 터치 패널의 가격들은 스크린 크기가 커질수록 비싸지므로 실제 일 상 생활에 보급화가 힘들다는 문제가 있다.



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

1.3 현재 있는 터치 스크린들의 한계

1.3.1 빅 노트

LG에서 펜과 수신기를 이용해서 벽면 혹은 모니터에 펜이 접촉했을 때 펜에서 수신기로 접촉이 일어 났다는 정보를 전달해 이벤트를 발생시킨다. 하지만 사용하려면 터치 이벤트를 감지하는 별도의 펜이 필요하며 가격이 매우 비싸다. 강의 기능에 특화되어 있으며 다양한 용도로 쓰이지 못한다.



[그림 6] 빅 노트의 가격



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀 명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

1.3.2 Surface Hub

마이크로소프트에서 84인치 대형 TV 스크린 크기의 서피스(Surface)제품을 2015년 1월 22일 선보였다. LED TV처럼 커다란 이 디바이스는 윈도우10이 돌아가는 터치스크린 머신이다. 마치 화이트보드처럼 전용 펜으로 글씨를 쓰거나 터치UI를 통해 편집을 할 수 있고, 팀 미팅에 활용하기도 적합한 윈도우10 컴퓨터 TV다. 하지만 이 제품도 마찬가지로 해당 기기가 매우 비싸며, 스크린의 크기가 매우 커서 휴대하기 불편하고 스크린의 크기 또한 조절이 불가능하다. 현재 상용화되기에는 시간이 많이 필요한 제품이다.



[그림 7] 서피스 허브



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

2 개발 내용 및 결과물

2.1 목표

- 1) 깊이 정보를 이용하여 벽 혹은 스크린을 인식한다.
- 2) 벽에 터치가 되었을 경우 그 좌표를 INPUT DATA 로 활용한다.
- 3) INPUT DATA 를 이용하여 실제 터치스크린을 사용하는 것 같은 터치 알고리즘을 구현한다.
- 4) 툴 바를 제공하여 터치를 활용할 수 있는 여러 가지 기능들을 제공한다.
- 5) 스마트 폰, 태블릿 pc, pc 등의 디스플레이 기반의 디바이스를 빔 프로젝터를 사용하여 큰 화면으로 터치 가능한 프로그램을 개발한다.



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)		· Things Haptic)	
팀명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용

1) 하드웨어의 사양과 배치

- 키넥트와 빔 프로젝터 사양을 조사한다.
- 위 사양과 사용자의 편의를 고려하여 키넥트와 빔 프로젝터 위치를 결정한다.

2) 키넥트 Depth Data

- 벽이나 스크린의 초기 Depth값을 인식한다.
- 이후 터치가 발생했을 때의 Depth값을 인식하고 좌표를 찾아낸다.

3) Data 처리 및 제스처 인식

- 확인한 좌표를 INPUT DATA로 활용하여 마우스처럼 활용할 수 있도록 Data 처리를 한다.
- 들어오는 여러 개의 좌표를 활용하여 실제 터치스크린과 같은 제스처를 사용할 수 있도록 한다.

4) Toolbar 제공

- 터치를 활용할 수 있는 여러 가지 기능을 가진 Toolbar를 제공한다.
- Toolbar는 펜 기능, 가상키보드 기능 등을 가지고 있다.
- 펜 기능에서 곡선이 부드러워지는 등 성능이 향상되었다.



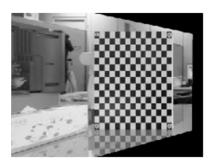
결과보고서		
프로젝트 명	MYTH(Making Your Things Haptic)	
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

5) 센서 각도에 따른 터치 범위보정

- 다양한 각도에서 인식되는 사다리꼴 화면을 직사각형으로 보정한다.
- 각도 범위의 한계를 지정한다.

Perspective Projection의 이미지 변화 예시

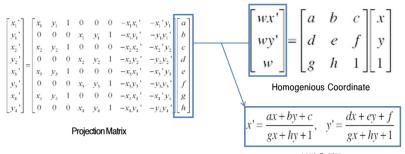




보정전 보정후

[그림 8] 보정 전과 후의 사각형 모양

좌표 보정 계산 방법

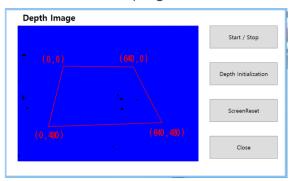


보정 후 좌표

Inverse Matrix를 양쪽 좌변에 곱해 a~h를 구함. Homogenious Coordinate를 사용하여 이동한 좌표를 구함.

[그림 9] 행렬을 이용한 좌표 보정 계산 방법

좌표 보정



유저가 설정한 스크린 좌표를 640X480 크기로 투영하여 보정된 좌표를 구함 .

[그림 10] 좌표를 보정한 모습

6) 리팩토링 및 성능 분석

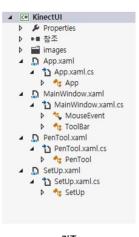
캡스톤디자인 Ⅱ Page 12 of 38 **결과보고서**



결과보고서			
프로젝트 명	프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

- public으로 선언되어 있던 변수들을 private으로 바꾸고, getter, setter 함수를 만들어 접근 가능하도록 바꾸었다.
- 이전에는 링크드 리스트를 직접 선언해서 사용하였으나, 안정성의 문제로 이미 구현되어있는 링크 드 리스트 라이브러리를 사용하고 필요한 함수들을 추가하여 구현하였다.
- 불필요한 if-else문은 삼항 연산자로 대체하고 for문을 줄여서 연산 속도를 향상시켰다.
- 하나의 main 함수에서 개발되었던 코드를 기능에 따라 클래스로 분할하여 가독성을 증가시켰다.

함수의 세분화 및 데이터의 클래스화



▲ C■ MYTH ▶ ► Properties
▶ ■■ 참조 images ▲ D App.xaml 1 App.xaml.cs ▲ C[®] Control.cs **⁴**⊈ ToolBar ▲ MainWindow.xaml MainWindow.xaml.cs MouseEvent
ToolBar ▲ □ PenTool.xaml ▲ ** PenTool.xaml.cs PenTool ▲ D SetUp.xaml SetUp.xaml.cs C# TouchPoint cs ♣ TouchPoint

기존

기존

클린 코딩 이후

클린 코딩 이후

[그림 11] 함수를 세분화하고 데이터를 클래스화한 모습

불필요한 작업의 최소화 및 변수 선언

```
// 좌우 반전으로 Color 값 출력
for (int i = 0; i < 480; i++)
                                                                                      상수들의변수선언
                                                                                                                                  // 좌위 반전으로 Color 값 출력
      for (int j = 639; j >= 0; j--)
                                                                                                                                      r (int i = 0; i ≺ screenPixelHeight i++)
            short currentDepth = depthPixel[i, j];
short initDepth = initdepthPixel[i, j];
                                                                                                                                      for (int j = (int)(screenPixelWidth - 1); j >= 0; j--)
            short subDepth;
            if (currentDepth > 2000 II initDepth > 2000)
                                                                                                                                          short currentDepth = depthPixel[i, j];
                  subDepth = 0;
                                                                                                                                           short initDepth = initdepthPixel[i, i];
                                                                                                                                          short subDepth;
                 subDepth = (short)(initDepth - currentDepth);
           if (subDepth < 0)
    subDepth = 0;
if (0 <= subDepth && subDepth <= 7)</pre>
                                                                                                                                          if (currentDepth > depthMaxLimit || initDepth > depthMaxLimit)
                                                                                                                                               subDepth = 0;
                                                                                                                                          else
subDepth = (short)((initDepth - currentDepth < 0) ? 0 : (initDepth - currentDepth));
                 colorPixels[colorPixelIndex++] = 255;
colorPixels[colorPixelIndex++] = 0;
colorPixels[colorPixelIndex++] = 0;
                                                                                                  조건무의 최소화
                                                                                                                                          colorPixels[colorPixelIndex++] = (byte)(subDepth <= blueColorLimit ? 255 : 0);
                                                                                              삼항연산자로반복되는
                                                                                                                                          colorPixels[colorPixelIndex++] = 0:
colorPixels[colorPixelIndex++] = (byte)[blueColorLimit < subDepth && subDepth <= redColorLimit ? 255 : 0);
                                                                                                      코드처리
                 colorPixels[colorPixelIndex++] = 0;
colorPixels[colorPixelIndex++] = 0;
colorPixels[colorPixelIndex++] = 255;
                                                                                                                                           ++colorPixelIndex
                 colorPixels[colorPixelIndex++] = 0;
```

[그림 12] 불필요한 작업을 최소화 하고 변수를 선언한 모습

캡스**톤디자인 Ⅱ** Page 13 of 38 **결과보고서**



결과보고서			
프로젝트 명	MYTH(Making Your Things Haptic)		
팀	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

기존에 구현되어 있던 Linked List로 변경

```
// 없어진 TouchPoint Linked List에서 삭제
public void removeNoneExist()
                                                                                      직접 구현한 Linked List 를
                                                                                                                                                   LinkedListNode<TouchPoint> R;
for (R = PointList.First; R != null; )
            chfoint T:
(T = head; T.next() != null; T = T.next()) Linked List 라이브러리로 대체
                                                                                                                                                           if (!R.Value.getExist())
             if (!next.getExist())
                                                                                                                                                                 if (R.Value.getrevision() > 5)
                                                                                                                                                                     R.Value.setclick(false);

if (!gestureFlag)

Up(MouseButton.Left);

rightFlag = true;

PointList.Remove(R);

pointCnt--;

R.Value.setrevision(0);

R = PointList.First;
                   if (next.revision > 5)
                        T.next().setclick(false);
Up(MouseButton.Left);
rightFlag = true;
T.removeNext();
T = head;
pointCnt--;
if (T.next() == null)
break;
                         next.revision = 0;
                                                                                                                                                                      R.Value.revisionPlus();
R = R.Next;
      initExist();
                                                                                                                                                     initExist();
                                        기존
                                                                                                                                                 클린 코딩 이후
```

[그림 13] 이미 존재하는 자료의 사용

- setPoint() 라는 함수는 터치할 때마다 호출되는 함수인데, 내부에서 네 번이나 재호출되기 때문에 프레임을 지연시킬 수 있을 것으로 판단되었다. 그러나 알고리즘 상, 정확한 판단을 위해 변경하지 않는 편이 좋을 것이라는 의견으로 모아졌고, 성능에 악영향을 미치고 있지는 않은지 판단 할 필요가 생겼다. 따라서 한 프레임 전체를 처리하는 시간과 이 함수만을 처리하는 시간을 타임스탬프로 찍어 비교해보았다. 성능 테스트 결과는 다음과 같았고, 성능에 악영향을 미치지 않는 것으로 결론지었다.

	전체	의심 부분	차지 비율(%)
평균 소요 시간	0.034064	0.000023	0.067520
최대 소요 시간	0.053640	0.000201	0.374720
최소 소요 시간	0.027160	0.000006	0.022091

- 프레임 지연 현상을 알아보고자 타임스탬프를 찍어 테스트해 보았다. 프레임을 표현하는 부분은 크게 두 부분으로, 붉은 점을 파악하여 터치포인트를 판단하는 부분과 depth 값에 따라 색을 지정, 저장해주는 부분이다. 전체 프레임과 해당 부분을 각각 테스트 하였다. 어느 부분에서 지연이 일어나고 있는지를 알았으나, 수정 불가능한 부분이라고 판단되어, 이 프로그램의 성능은 한 프레임 당 약 0.03 초로 마무리하게 되었다.

	전체	터치포인트 판단 부분	색 지정, 저장 부분
평균 소요 시간	0.034064	0.005889	0.027706
차지 비율 (%)	100	17.29	81.33

캡스**톤디자인 II** Page 14 of 38 **결과보고서**

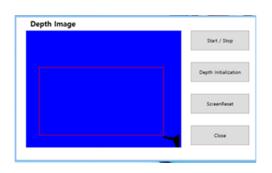


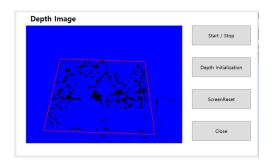
결과보고서			
프로젝트 명	르젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

7) 화면 캘리브레이션의 간편화

- 터치 범위 설정을 손으로 정하는 방법에서 마우스 클릭으로 가능하게 한다.
- 프로그램이 보여주는 화면을 Depth 값에서 RGB값을 추가한 비트맵으로 교체한다.

Calibration 방법 추가





기존 마우스클릭추가

[그림 14] Calibration을 간편화한 모습

Depth Camera 와 Color Camera 시점 문제





기존의문제 해결시

[그림 15] depth camera와 color camera를 합한 모습



결과보고서		
프로젝트 명	MYTH(Making Your Things Haptic)	
며	미네르바	
Confidential Restricted	Version 1.5 2015-NOV-26	

2.2.2 시스템 기능 요구사항

하드웨어

1) Kinect



[그림 16] 키넥트 v1

- 키넥트의 깊이 정보 추출 원리는 적외선 카메라의 중심점을 원점으로 하여 객체를 3차원으로 표시한다. Z축은 영상영역(image plane)에 수직이고, X축은 Z축에 대하여 수직이며, 적외선 카메라에서레이저 프로젝터로 향하는 방향이다. Y축은 Z축과 X축에 대하여 수직이다.
- 키넥트 RGB카메라 오른쪽에 부착 된 것은 적외선 송출 프로젝터이다. 이 적외선 프로젝터는 전면의 물체에 픽셀단위의 적외선을 송출한다. RGB카메라의 왼쪽에 부착된 것은 적외선 카메라이다. 적외선 프로젝터에서 송출된 점들이 반사되는 것을 받아들여 물체를 인식하게 된다.
- Kincet는 X box를 사용할 때 1.2~3.5m까지 지원되고, 확장 모드는 0.7~6m까지 지원된다. 센서의 각도는 y축으로 ±57°까지 센서를 인식하고, x축으로는 ±43°까지 인식한다. Kinect자체의 각도를 위아래로 27°까지 조정할 수 있다. Kinect가 제대로 동작하려면 Kinect와 물체의 거리가 최소 87cm 이상 떨어져야 한다.



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

2) 빔 프로젝터



[그림 17] 빔 프로젝터

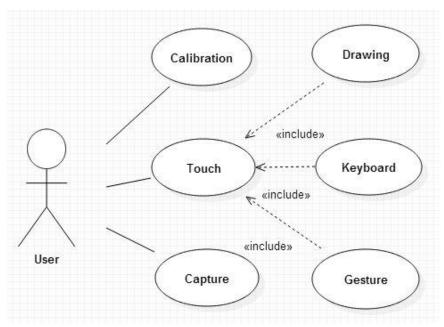
빔 프로젝터의 사양은 아래의 조건이 충족되어야 한다.

화면 크기	16:10
명암비	15,000:1
밝기	500 Ansi Lumens
해상도	와이드 HD-WXGA(1280x800)
투사렌즈	수동 조정 초점, 고정 줌
표준 투사거리	0.8m~3.25m
지원 화면 크기	20인치~100인치
입출력 단자	HDMI(Ver 1.4)



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

기능 요구사항



[그림 18] 기능 요구사항

- 1) User: 이 기능을 사용하게 될 사용자
- 2) Calibration : 초기에 키넥트 이벤트 발생 범위를 빔 프로젝터 화면으로 지정
- 3) Touch : 터치를 기반으로 이루어진 행동
 - Drawing : 낙서나 그림과 같이 펜을 이용하는 기능
 - Keyboard : 키보드가 없는 환경에서 타자를 치기 위한 가상 키보드 기능
 - Gesture : 사용자의 모션에 따른 기능 수행
- 4) Capture : 현재의 스크린을 캡쳐하여 저장할 수 있는 기능

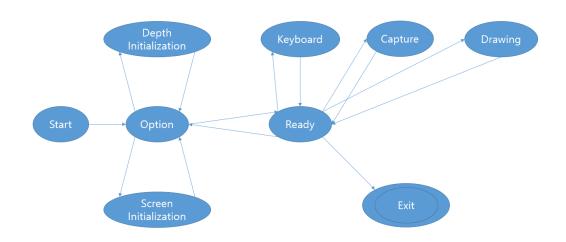
2.2.3 시스템 비 기능(품질) 요구사항

- 1) 사용자가 발생시키는 이벤트인 터치를 손으로 하기 때문에 손을 몸으로 가린다면 인식할 수 없다.
- 2) 빔 프로젝터의 화면을 사용자가 가릴 수 있다.
- 3) PC를 통해 터치 디스플레이를 표현하기 때문에 멀티 터치가 가능해야 한다.
- 4) 키보드가 없기 때문에 가상키보드를 지원한다.



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

2.2.4 시스템 구조 및 설계도



[그림 19] 상태 다이어그램

- Start : 프로젝트를 시작한다.
- Option : 프로젝트 시작 시 옵션을 실행해 Depth Initialization과 Screen Initialization을 실행한다.
- Depth Initialization : 화면의 깊이 값을 갱신한다.
- Screen Initialization : 화면의 크기를 갱신한다.
- Ready: 부 프로그램이 실행되지 않은 대기 상태.
- Keyboard : 키보드를 실행한다.
- Capture : 현재 화면을 캡처해 바탕화면에 저장한다.
- Drawing : 그림 판을 실행한다.
- Exit : 종료



결과보고서		
프로젝트 명	MYTH(Making Your	Things Haptic)
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

2.2.5 실제 툴 바의 UI

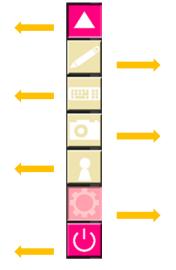
아래 [그림 12]와 [그림 13]은 MYTH프로그램을 실행 시켰을 때 켜지는 툴 바의 전개도와 실제 실행 화면에서 나타나는 UI이다.

하위 항목으로 설정, 펜, 키보드, 캡쳐, 전원 모드 버튼이 나타난다. 한 번 더 클릭할 시에 아래의 모드 들이 접히면서 첫 실행 시와 같이 수행된다.

가상 키보드가 호출되면서 사용자의 touch입력으로 키보드 입력을 하게 된다. 인터넷 창, 윈도우 화면에서 모두 키보드 입력이 가능하다.

1인 모드/다인 모드의 선택이 가능하다.1인 모드 선택 시 터치, 마우스 컨트롤이 모두 가능하다. 다인 모드 선택 시 터치기능만 가능하다.

선택 시 프로그램이 종료된다.

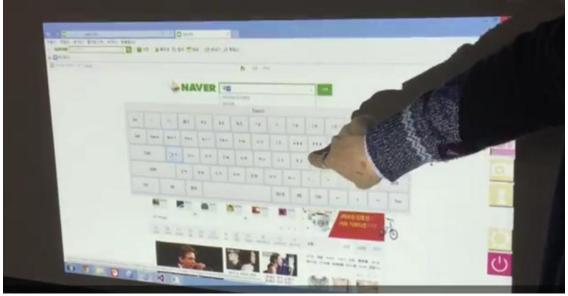


[그림 20] UI 설명

사용자가 원하는 그림을 그리거나 글씨를 쓸 수 있다. 길게 누르면 펜 색상 툴의 실 행으로 펜의 색상을 선택할 수 있고, 부분 지우기, 전체 지우기 등이 가능하다.

현재 보이는 창을 캡쳐하여 이미지 파일로 저장하게 된다. 저장은 바탕화면에 저장되면 저장되는 파일 명은 캡쳐한 시각과 번호로 저장된다.

자동으로 호출 되면서, depth 값을 초기화하고, 사용자가 화면 인식 범위를 지정한다. 후에는 사용자가 클릭을 하면 실행이 되고 그 때에도 depth값 초기화와 화면 인식 범위를 지정할 수 있다.



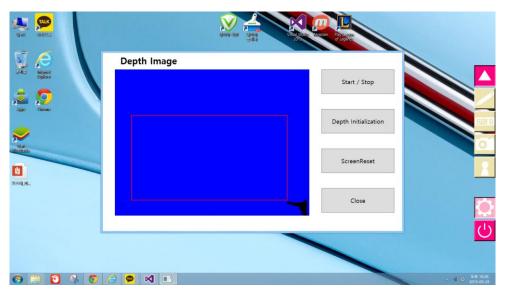
[그림 21] 실제 실행 화면의 UI



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

2.2.6 UI 실행 창

- 설정 모드(〇)



[그림 22] 화면 범위 조정

- 펜 모드()

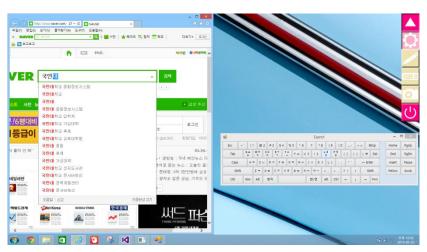


[그림 23] 펜 기능의 사용



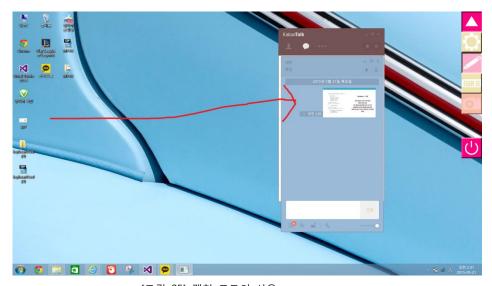
결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

- 키보드 모드(



[그림 24] 가상키보드의 호출

- 캡쳐 모드(**조**)



[그림 25] 캡쳐 모드의 사용

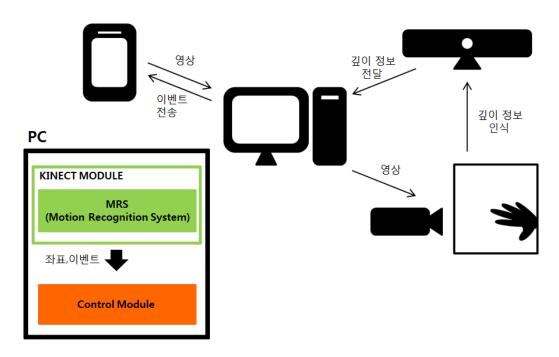
현재 화면을 저장하여 바탕화면에 그림 파일이 생성된다. 드래그 이벤트로 카카오 톡이나 메일에 첨부하여 파일을 보낼 수 있다.

캡스톤디자인 Ⅱ Page 22 of 38 **결과보고서**



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

2.2.7 이벤트 전송 과정



[그림 26] 이벤트 전송 다이어그램

시스템의 전체적인 과정은 위의 다이어그램과 같이 디스플레이 기반의 디바이스의 화면을 빔 프로젝터를 사용하여 벽에 출력하게 하고 키넥트를 이용하여 Depth 정보를 PC로 전달한다. MRS는 Depth 정보를 통해 사용자의 동작을 인식하는 Motion Recognition System 이다. MRS는 그에 맞는 좌표와 이벤트를 Control Module 로 넘겨주고 Control Module 에서 원하는 동작의 이벤트를 실행 한다.



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

2.2.8 활용/개발된 기술

MYTH 프로그램은 터치 스크린이 아닌 곳을 터치 스크린처럼 사용하는 것이므로 터치 알고리즘이 가장 중요하다. 직관적인 제스처를 제공해야 하므로 실제 스마트 폰이나 컴퓨터에서 사용하는 확대, 축소 제스처와 같게 알고리즘을 만들었다.

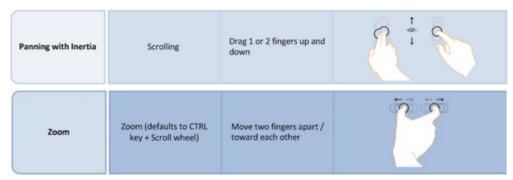
1) 단일터치



[그림 27] 단일터치

터치포인트가 한 개 일 경우 단일터치로 인식을 하고 마우스 왼쪽 버튼 클릭이벤트를 사용한다. 그리고 터치포인트 생성시 클릭상태를 유지하여 드래그 이벤트를 가능하게 하고 한 좌표에서 클릭상태를 유지 할 경우 마우스 오른쪽 버튼 클릭이벤트를 발생시킨다.

2) 제스처



[그림 28] 제스처 이벤트

터치포인트가 두 개 일 경우 제스처로 인식한다. 제스처 인식에는 확대와 축소, 스크롤이 있고 입력된 두 개의 터치포인트가 발생시키는 제스처 인식 방법은 다음과 같다.

- 스크롤 다운 : 두 점이 이동하는 방향의 각도의 차가 30도 이하이고 -> 방향을 0도를 기 준으로 하여 30~150도 일 때 이벤트 발생.
- 스크롤 업 : 두 점이 이동하는 방향의 각도의 차가 30도 이하이고 -> 방향을 0도를 기준으로 하여 210~330도 일 때 이벤트 발생
- 확대 : 두 점이 이동하는 방향의 각도의 차가 90도 이상이고 이전 프레임과 비교하여 일 정거리 이상 멀어지면 이벤트 발생
- 축소 : 두 점이 이동하는 방향의 각도의 차가 90도 이상이고 이전 프레임과 비교하여 일

캡스**톤디자인 II** Page 24 of 38 **결과보고서**



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

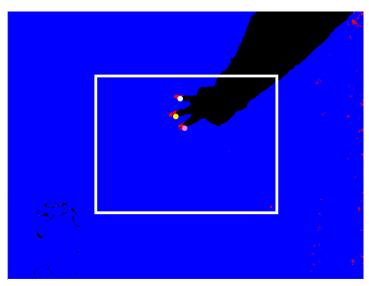
정거리 이상 가까워지면 이벤트 발생

3) 마우스 좌표

사용자가 빔 프로젝터로 출력된 영상의 보고 터치를 하게 되면 Kinect를 통해 터치이벤트를 인식하고 사용자가 출력 영상을 보고 클릭한 좌표와 실제 PC의 좌표의 맞추기 위해 Kinect 카메라가 인식하는 영상의 좌 상단 좌표를 이용하여 다음과 같은 계산 방식을 사용한다.

(현재 좌표 - 화면 좌 상단 좌표) / 화면의 크기 * PC해상도 위와 같은 계산 방식으로 실제 터치 패널 같이 사용자가 클릭한 좌표를 PC의 좌표로 사용 가능하게 한다.

Touch Algorithm



[그림 29] 키넥트 터치 알고리즘

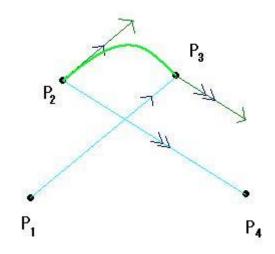
Kinect를 시작 시 초기 Depth값을 픽셀 단위로 저장하고 현재 프레임과 비교하여 차이가 없는 부분의 픽셀 컬러 값을 파란색으로 지정하고 손가락이라고 인식 할 수 있는 Depth값의 차이를 갖는 픽셀의 컬러 값을 빨간색으로 지정 이외의 것들을 검은색으로 지정한다. 이 컬러 비트맵을 이용하여 픽셀의 컬러가 빨간색인 픽셀을 재귀함수를 통해 연결되어 있는 픽셀의 수를 이용하여 일정 이하 또는 일정 이상의 픽셀 수를 갖는 점은 오류로 판단하고 터치포인트로 인식된 점의 중점을 이용하여 좌표로 사용한다.

Catmull-rom splines



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

점을 지나가는 곡선을 표현하는 알고리즘이다. 펜 기능을 사용 할 때, 곡선 및 선이 매끄럽지 않아 이를 보정하기 위해 사용하였다. 점을 지나간다는 점에서 점을 지나지 않는 베지어 곡선 알고리즘과 큰 차이가 있다.



[그림 30] Catmull-rom splines

[그림 22]의 곡선을 표현하는 매개 변수 방정식은 다음과 같다.

 $P(t) = 0.5 * ((2.0 * P1) + (-P0 + P2) * t + (2.0 * P0 - 5.0 * P1 + 4.0 * P2 - P3) * t^2 + (-P0 + 3.0 * P1 - 3.0 * P2 + P3) * t^3);$

보정 결과, 펜 기능의 선이 부드러워졌음을 확인할 수 있었다.



[그림 31] 펜 보정 전과 후

센서 각도에 따른 터치 범위 보정

화면 캘리브레이션의 간편화

2.2.9 현실적 제한 요소 및 그 해결 방안

하드웨어

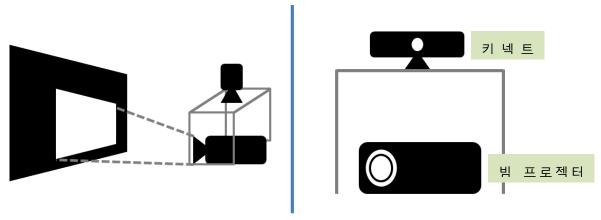
1) 하드웨어의 위치 고정이 필요하다. 위치가 이동되면 캘리브레이션 함수가 제대로 발동되지 않을 수



결과보고서		
프로젝트 명 MYTH(Making Your Things Haptic)		
팀 명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

있다. 키넥트와 빔과의 위치를 잘 고정해야 한다.

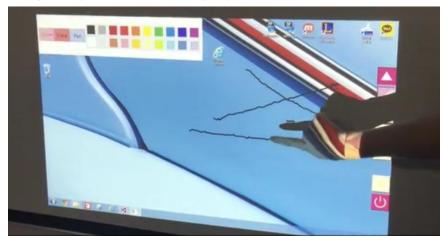
→ 현재는 아래의 [그림 17]과 같이 빔 프로젝터와 키넥트를 배치해야 한다. 키넥트는 벽과 평행하 게 놓아야 하고 빔 프로젝터는 사용자가 원하는 위치에 두어 화면의 크기를 조절할 수 있다.



[그림 32] 하드웨어 설계도

2) 빔 프로젝터가 화면을 송출할 때 사람이 가리고 있으면 그 부분은 화면이 제대로 출력되지 않는다.

→ 화면이 제대로 출력이 되지 않아도 사용자가 MYTH 프로그램을 사용하는 데에는 불편함이 없다.
아래의 [그림 18]은 실제 MYTH 프로그램을 사용하는 모습이다.



[그림 33] 실제 MYTH 사용 모습

소프트웨어

- 1) Kincet가 제공하는 library의 skeleton 인식이 정확하지 않았다. Skeleton의 인식이 오래 걸린다는 것과 skeleton의 손목과 손인식이 부정확하고 제대로 된 좌표 값이 출력되지 않는 단점이 있다. 이것을 보완하기 위해 상반신만 인식하는 방법을 고려하였으나, 이 방법도 정확하지 않았다.
 - → 현재는 순수한 depth이미지의 변화만으로 이벤트들을 인식한다.
- 2) 화면이 송출될 벽에서 실제로 화면이 비춰지는 영역을 자동으로 설정하지 못한다. 초기의 아이디어는

캡스**톤디자인 Ⅱ** Page 27 of 38 **결과보고서**



결과보고서		
프로젝트 명	MYTH(Making Your	· Things Haptic)
팀명	미네르바	
Confidential Restricted	Version 1.5	2015-NOV-26

빔 프로젝터로 특정한 색을 가진 사각형 모양의 화면을 송출하여 키넥트의 RGB캠으로 그 색들이 나타내는 범위를 유효한 화면으로 설정하려고 하였다. 그러나 흰색 벽의 경우에는 특정한 색을 특정한 영역에만 송출하더라도 빛의 반사에 의하여 주변도 송출하는 것과 같은 색을 띠게 되는 현상이 생겼다. 그 결과 키넥트가 색을 인식하여 유효한 화면을 감지할 때 그 주변도 같이 인식이 되어버린다. 이것을 해결하기 위해서 다양한 조도에서 다양한 색으로 테스트를 하였으나 주변 환경에 따라 차이가매우 심하기 때문에 다른 방법을 고안할 예정이다.

→ 사용자가 직접적으로 벽에 원하는 화면 위치와 크기의 네 꼭지점을 터치하게 하여 그 부분을 유효 한 화면으로 인식한다.



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

2.2.10 결과물 목록

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과	비고
	<i>마우스</i> 좌클릭	마우스를 사용했을 때 클릭한 것과 같은 효과를 준다.		마우스 클릭이 동작한다.	성공	mouseEvents()
단일 터치	<i>마우스</i> 우클릭	마우스를 사용했을 때 우클릭 한 것 같은 효과를 준다.	바탕화면에서 손가락을 화면에 1 초정도 대고 있으면 1) 아이콘이나 버튼이 위치한 곳에 손가락을 대고 있었다면 아이콘이나 버튼에 대한 단축메뉴가 열린다. 2) 아이콘이나 버튼이 없는 곳에 손가락을 대고 있었다면 바탕화면에 대한 단축메뉴가 열린다.	마우스 우클릭이 동작한다.	성공	mouseEvents()
	드래그	마우스 드래그를 사용하는 것과 같은 효과를 준다.	화면에 손가락을 대고 손가락을 이동하면 선택 블록이 생기면서 블록 안쪽에 아이콘이 있으면 아이콘이 선택된다.	마우스 드래그가 동작한다.	성공	mouseEvents()
멀티 터치	<i>스크롤</i>	마우스 휠을 굴렸을 때 의 효과를 준다.	스크롤이 긴 창에서 손가락으로 두점을 찍고 손가락을 위로 올렸을 경우 스크롤을 아래로 내리는 효과가나타나고 손가락을 아래로 내렸을 경우 스크롤을 위로 올리는 효과가나타난다.	마우스 휠이 작동한다	성공	mouseEvents()



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀 명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

	줌	기능을 한다.	손가락으로 두 점을 찍고 양쪽으로 펼쳤을 경우 화면의 확대가 되고, 펼쳐진 상태에서 손가락을 한쪽으로 모았을 경우 축소가 된다.	확대, 축소가 가능한 화면에서 확대 혹은 축소를 한다.	성공	MouseEvents()
Toolbar	<i>옵션버튼</i>	Kinect 동작 의 시작, 중지, Depth 값의 초기화, Depth Image 를 초기화하는 창을 연다.	Toolbar 에서 톱니바퀴 모양의 버튼을 터치하면 옵션 창이 열린다.	옵션 창이 열림.	성공	Open_MouseLef tButtonDown()
옵	Screen Reset 버튼	Screen 에 생긴 포인트나 사각형을 초기화 시켜준다.	Depth Image 에 나타나는 한 점을 계속 터치하고 있으면 초록색 점이나타난다. 그 초록색 점은 Screen Reset을 통해 지울 수 있다. 다양한 사각형 모양으로 인식된 스크린도 직사각형으로 보정가능하며 마우스 클릭으로 스크린범위를 지정할 수 있다.	Depth Image 에 있던 점이나 사각형이 지워진다.	성공	ScreenSize_Click() SetPerspective() GetInverseMatrix() GetAdjMatrix() GetTransposeMatrix() GetCoFactorMatrix() GetMinorMatrix() Get2x2Determinant() GetDeterminant() SetPointOrder() ResetScreenPoint() ResetSearchArea()
	Depth Initialization 버튼	Kinect 가저 장한 Depth 값을 초기화 한다.	· · · · · · · · · · · · · · · · · · ·	Depth Image 가 파란 화면으로 변한다.	성공	DepthInit_Click()
	Start/Stop	Kinect 동작을	처음에 Depth Image 에 물체가 인식되는 것을 확인하고 Start /	Start->Stop 시 Depth Image 가	성공	StartStop_Click()



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
며	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

Toolbar	키보드	버튼을 눌러	검색사이트에 접속 한 후 키보드	키보드가 화면에	성공	Image_MouseD
	색 버튼	펜의 색을 변경할 수 있다.	그림을 그리다가 현재 펜과 다른 색의 색 버튼을 누르고 다시 그림을 그리면 이전에 썻던 펜과 다른 색으로 그림을 그린다.	펜의 색을 바꿀 수 있다.	성공	Btn_Pen()
팔레트	Erase 버튼	지금까지 그렸던 그림을 지우개로 부분 지운다.	화면에 그림이 그려져 있는 상태에서 Erase 버튼을 누르면 포인트가 지우개로 바뀌면서 그림을 부분적으로 지울 수 있다.	드래그를 하여 그림의 부분을 지울 수 있다.	성공	Btn_Erase()
	All Erase 버튼	지금까지 그렸던 그림을 모두 지운다.	화면에 그림이 그려져 있는 상태에서 All Erase 버튼을 누르면 그림이 모두 지워지는 것을 확인할 수 있다.	여태까지 그려져 있던 그림이 모두 지워진다.	성공	Btn_AllErase()
그림판	팔레트	그림을 그리는 펜의 색을 바꿔주거나 지우개, 혹은 그림을 모두 지울 수 있다.	그림판이 켜진 상태에서 화면을 터치한 채 1초 정도 기다리면 팔레트가 열린다.	팔레트가 열린다.	성공	Preview_Mouse_ Right_Button_Do wn()
	그리기	그림판을 이용해 그림을 그릴 수 있게 해준다.	그림판이 켜진 상태에서 화면을 터치하고 그 상태로 드래그하면 드래그하는 형태로 그림이 그려진다. 선의 형태가 비교적 더 부드러워지고 인식률이 높아졌다.	드래그하는 모양으로 그림이 그려진다.	성공	mouseEvents() catmull_rom_spli nes() calculateCurve()
Toolbar	펜 버튼	그림판 기능을 수행할 수 있게 해준다.	Toolbar 에서 펜 모양의 버튼을 터치한다. 마우스 포인터가 사라지고 점 형태로 나타난다.	마우스 포인터가 초기 펜 색의 점 형태로 변한다.	성공	Pen_MouseLeftB uttonDown()
		시작/중지한 다.	Stop 버튼을 누른다. 다시 물체를 인식시키면 물체가 인식되지 않는다. 다시 Start / Stop 버튼을 누르면 물체가 인식되는 것을 확인할 수 있다.	멈춘다. Stop -> Start 시 Depth Image 에 변화가 생긴다.		



결과보고서				
프로젝트 명 MYTH(Making Your Things Haptic)				
팀 명	팀 명 미네르바			
Confidential Restricted	Version 1.5 2015-NOV-26			

	<u> </u>	키보드의 키를 입력할 수 있다.	버튼을 눌러 키보드를 부른다. 검색어 란에 한번 터치한 후 키보드의 버튼을 누르면 해당하는 키보드 버튼이 눌린다.	나타난다.		own()
	캡처 버튼	보고 있는 화면의 스크린 샷을 찍을 수 있다.	그림판으로 그림을 그린 후에 캡처 버튼을 눌러 스크린 샷을 찍으면 바탕화면에 그림을 그렸던 화면이 캡처되어 바탕화면에 저장된다.	바탕화면에 현재 화면의 스크린 샷 파일이 나타난다.	성공	Keyboard_Mous eLeftButtonDow n()
	Multi mode 버튼	다인 모드를 사용할 수 있다.	터치에 반응하는 프로그램을 실행한 후에 Single mode 로 터치하면 여러 포인트를 동시에 터치할 수 없지만 Multi mode 에서는 터치가 가능하다.	드래그, 제스처 기능이 동작하지 않고 동시에 여러 개의 터치가 들어왔을 때 터치로 인식한다.	성공	Capture_MouseL eftButtonDown()
	종료 버튼	MYTH 를 종료한다.	여태껏 사용했던 기능을 사용할 수	Toolbar 가사라진 다. 제공되는 모든 기능을 사용할 수 없다.	성공	Usermode_Mous eLeftButtonDow n()



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

3 자기평가

3.1 사용가능성

최근 터치스크린 기반의 디스플레이가 각광을 받고 있다. 왜냐하면 사용하는데 어려움 없이 직관적으로 알 수 있기 때문이다. 우리 프로젝트 MYTH는 그 점을 착안하여 컴퓨터나 휴대폰 등 여러 가지 기기들을 터치스크린으로 사용할 수 있게 해준다. 터치스크린을 사용하게 될 경우 키보드나 마우스 같은 별도의 입력 장치가 필요하지 않아 편리하다.

그리고 프로젝트 MYTH는 프로젝터를 활용하기 때문에 화면의 크기가 유동적이다. 큰 화면으로 사용할 경우 고정된 크기의 디스플레이보다 MYTH가 휴대하기 편리하다. 또 화면의 크기를 유동적으로 조정할수 있기 때문에 혼자서 혹은 여럿이서 사용할 수 있다.

마지막으로 대형 터치 패널은 가격이 매우 비싼데 MYTH를 사용할 경우 대형 터치패널을 사용하는 것 보다 훨씬 저렴한 가격에 터치스크린을 사용할 수 있다. 그러므로 편리성, 휴대성, 경제성 면에서 봤을 때 이전에 있던 제품들보다 경쟁력이 있어 충분히 사용 가능성이 있다고 생각한다.

4 참고 문헌

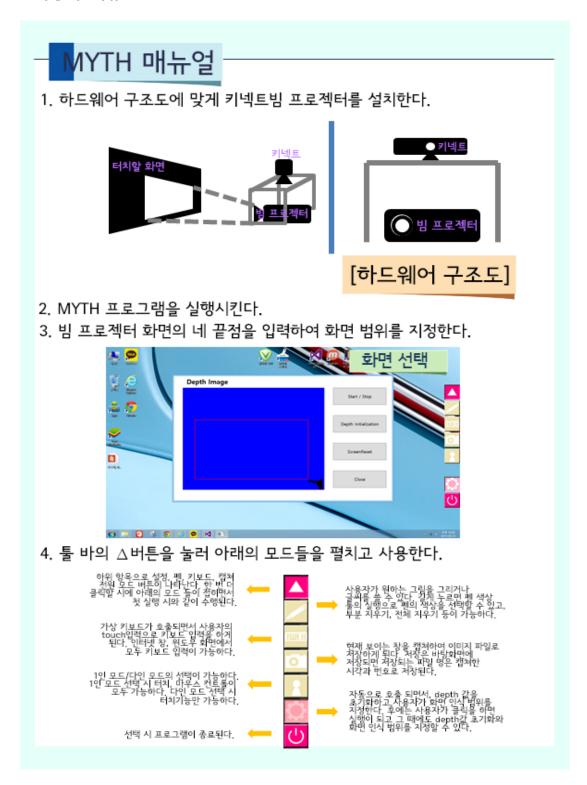
번호	종류	제목	출처	발행 년도	저자	기타
01	서적	키넥트 프로그래밍		2012.12.11	자렛 웹,	
					제임스	
					애쉴리	
02	서적	Hello 키넥트		2013.07.30	이혁수	
					저자	
03	서적	유니티 게임 개발 스타트업		2014	지국환	
					저자	
04	웹 페이지	https://msdn.microsoft.com/ko-	마이크로			코드
		kr/library/vstudio	소프트사			참고



결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

5 부록

5.1 사용자 매뉴얼





결과보고서			
프로젝트 명 MYTH(Making Your Things Haptic)			
팀 명	미네르바		
Confidential Restricted	Version 1.5 2015-NOV-26		

5.2 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
	<i>마우스</i> 좌클릭	마우스를 사용했을 때 클릭한 것과 같은 효과를 준다.		마우스 클릭이 동작한다.	성공
단일 터치	마우스를 가용했을 때 소가락을 대고 있으면 사용했을 때 소가락을 대고 있 어이콘이나 버튼의 가우스 우클릭 한 구클릭 한 구클릭 건 같은 효과를 준다. 4) 아이콘이나 버튼의 소가락을 대고 있 바탕화면에 대한	3) 아이콘이나 버튼이 위치한 곳에	마우스 우클릭이 동작한다.	성공	
	드래그	마우스 드래그를 사용하는 것과 같은 효과를 준다.	화면에 손가락을 대고 손가락을 이동하면 선택 블록이 생기면서 블록 안쪽에 아이콘이 있으면 아이콘이 선택된다.	마우스 드래그가 동작한다.	성공
멀티 터치	<i>스크롤</i>	마우스 휠을 굴렸을 때 의 효과를 준다.	스크롤이 긴 창에서 손가락으로 두점을 찍고 손가락을 위로 올렸을 경우 스크롤을 아래로 내리는 효과가나타나고 손가락을 아래로 내렸을 경우 스크롤을 위로 올리는 효과가나타난다.	마우스 휠이 작동한다	성공



결과보고서				
프로젝트 명 MYTH(Making Your Things Haptic)				
팀 명 미네르바		발바		
Confidential Restricted	Version 1.5	2015-NOV-26		

	줌	│ 기능을	손가락으로 두 점을 찍고 양쪽으로 펼쳤을 경우 화면의 확대가 되고, 펼쳐진 상태에서 손가락을 한쪽으로 모았을 경우 축소가 된다.	확대, 축소가 가능한 화면에서 확대 혹은 축소를 한다.	성공
Toolbar	옵션버튼	Kinect 동작 의 시작, 중지, Depth 값의 초기화, Depth Image 를 초기화하는 창을 연다.	Toolbar 에서 톱니바퀴 모양의 버튼을 터치하면 옵션 창이 열린다.	옵션 창이 열림.	성공
	Screen Reset 버튼	Screen 에 생긴 포인트나 사각형을 초기화 시켜준다.	Depth Image 에 나타나는 한 점을 계속 터치하고 있으면 초록색 점이 나타난다. 그 초록색 점은 Screen Reset 을 통해 지울 수 있다. 다양한 사각형 모양으로 인식된 스크린도 직사각형으로 보정 가능하며 마우스 클릭으로 스크린 범위를 지정할 수 있다.	Depth Image 에 있던 점이나 사각형이 지워진다.	성공
옵션	Depth Initialization 버튼	Kinect 가저 장한 Depth 값을 초기화 한다.	MYTH 가 작동하고 있는 상태에서 키넥트를 움직이면 초기 Depth 값과 차이를 보이면서 Depth Image 가 튄다. 그 때 Depth Initialization 버튼을 누르면 그 상태로 바라본 Depth 값이 초기값으로 지정된다.	Depth Image 가 파란 화면으로 변한다.	성공
	Start/Stop	Kinect 동작을 시작/중지한 다.	처음에 Depth Image 에 물체가 인식되는 것을 확인하고 Start / Stop 버튼을 누른다. 다시 물체를 인식시키면 물체가 인식되지 않는다. 다시 Start / Stop 버튼을 누르면 물체가 인식되는 것을 확인할 수 있다.	Start->Stop 시 Depth Image 가 멈춘다. Stop -> Start 시 Depth Image 에 변화가 생긴다.	성공
Toolbar	펜 버튼	그림판 기능을 수행할 수 있게	Toolbar 에서 펜 모양의 버튼을 터치한다. 마우스 포인터가 사라지고 점 형태로 나타난다.	마우스 포인터가 초기 펜 색의 점 형태로 변한다.	성공

캡스톤디자인 Ⅱ Page 36 of 38 **결과보고서**



결과보고서				
프로젝트 명 MYTH(Making Your Things Haptic)				
팀	미네르바			
Confidential Restricted	Version 1.5	2015-NOV-26		

		해준다.			
그림판	그리기	그림판을 이용해 그림을 그릴 수 있게 해준다.	그림판이 켜진 상태에서 화면을 터치하고 그 상태로 드래그하면 드래그하는 형태로 그림이 그려진다. 선의 형태가 비교적 더 부드러워지고 인식률이 높아졌다.	드래그하는 모양으로 그림이 그려진다.	성공
	팔레트	그림을 그리는 펜의 색을 바꿔주거나 지우개, 혹은 그림을 모두 지울 수 있다.	그림판이 켜진 상태에서 화면을 터치한 채 1초 정도 기다리면 팔레트가 열린다.	팔레트가 열린다.	성공
버튼 팔레트 H튼	All Erase 버튼	지금까지 그렸던 그림을 모두 지운다.	화면에 그림이 그려져 있는 상태에서 All Erase 버튼을 누르면 그림이 모두 지워지는 것을 확인할 수 있다.	여태까지 그려져 있던 그림이 모두 지워진다.	성공
		지금까지 그렸던 그림을 지우개로 부분 지운다.	화면에 그림이 그려져 있는 상태에서 Erase 버튼을 누르면 포인트가 지우개로 바뀌면서 그림을 부분적으로 지울 수 있다.	드래그를 하여 그림의 부분을 지울 수 있다.	성공
	색 버튼	펜의 색을 변경할 수 있다.	그림을 그리다가 현재 펜과 다른 색의 색 버튼을 누르고 다시 그림을 그리면 이전에 썻던 펜과 다른 색으로 그림을 그린다.	펜의 색을 바꿀 수 있다.	성공
Toolbar	키보 <u>드</u> 버튼	버튼을 눌러 키보드의 키를 입력할 수 있다.	검색사이트에 접속 한 후 키보드 버튼을 눌러 키보드를 부른다. 검색어 란에 한번 터치한 후 키보드의 버튼을 누르면 해당하는 키보드 버튼이 눌린다.	키보드가 화면에 나타난다.	성공
	캡처 버튼	보고 있는 화면의 스크린 샷을 찍을 수 있다.	그림판으로 그림을 그린 후에 캡처 버튼을 눌러 스크린 샷을 찍으면 바탕화면에 그림을 그렸던 화면이 캡처되어 바탕화면에 저장된다.	바탕화면에 현재 화면의 스크린 샷 파일이 나타난다.	성공



결과보고서				
프로젝트 명 MYTH(Making Your Things Haptic)				
팀 명	미네르바			
Confidential Restricted	Version 1.5	2015-NOV-26		

Multi mode 버튼	다인 모드를 사용할 수 있다.	터치에 반응하는 프로그램을 실행한 후에 Single mode 로 터치하면 여러 포인트를 동시에 터치할 수 없지만 Multi mode 에서는 터치가 가능하다.	드래그, 제스처 기능이 동작하지 않고 동시에 여러 개의 터치가 들어왔을 때 터치로 인식한다.	성공
종료 버튼	MYTH 를 종료한다.		Toolbar 가사라진다 . 제공되는 모든 기능을 사용할 수 없다.	성공