과거로의 여행 - Part 1.

Projects using Image Processing

2018. 08. 13.

Hero Seo

piantic.github.io
joshua227.tistory.com
yeongungseo@gmail.com

"하고 싶은게 너무 많아요. 뭘 해야할지 모르겠어요."

"~~하고 싶은게~~ 너무 많아요. 뭘 해야할지 모르겠어요."

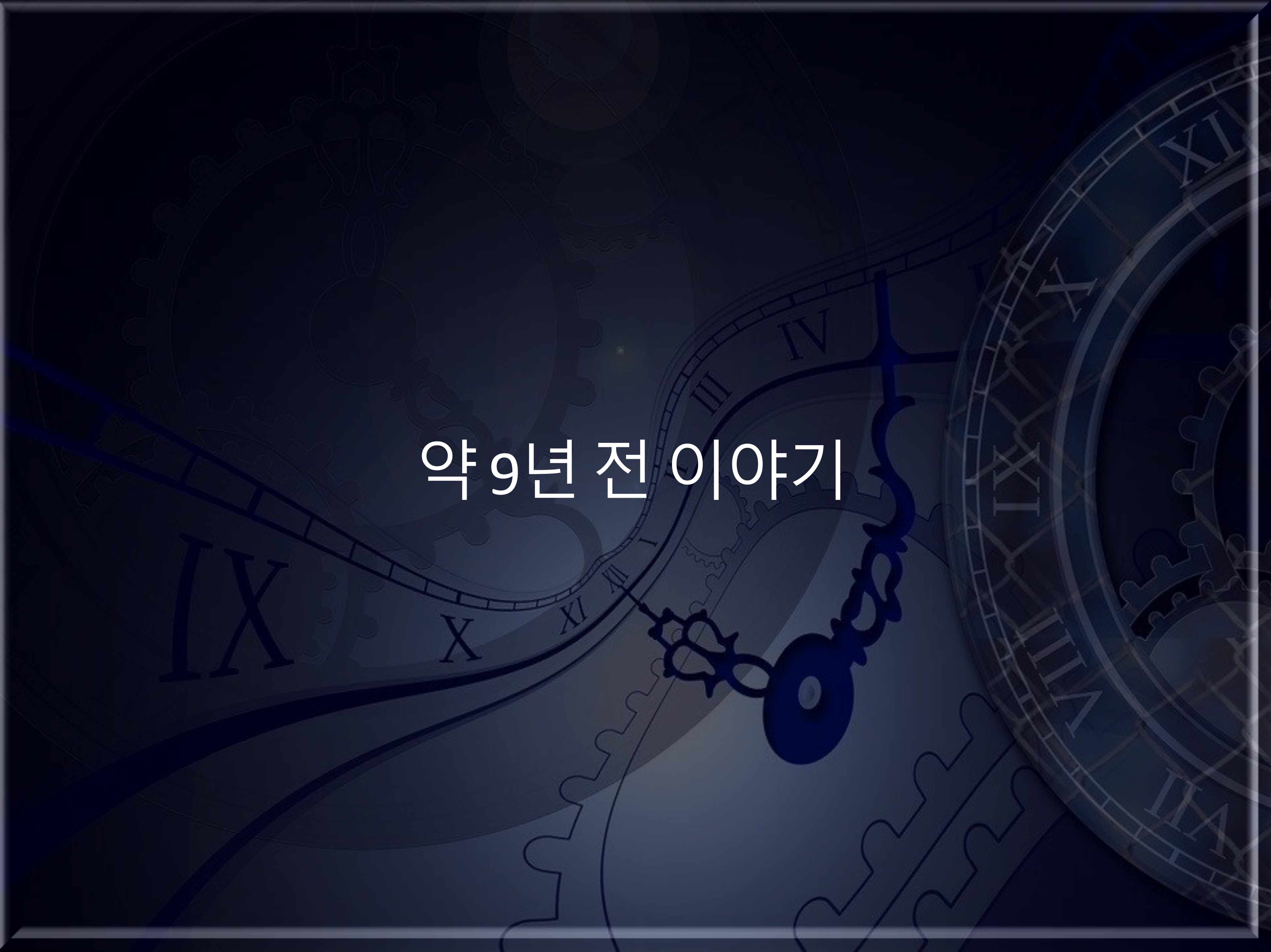
해야 할게

> "~~하고 싶은게~~ 너무 많아요. 뭘 해야할지 모르겠어요."

(무조건) 해야 할게

"~~하고 싶은게~~ 너무 많아요. 뭘 해야할지 모르겠어요."

(무조건, 잘) 해야 할게



약 9년 전 이야기

Lecture

6

Lecture - 6

Project

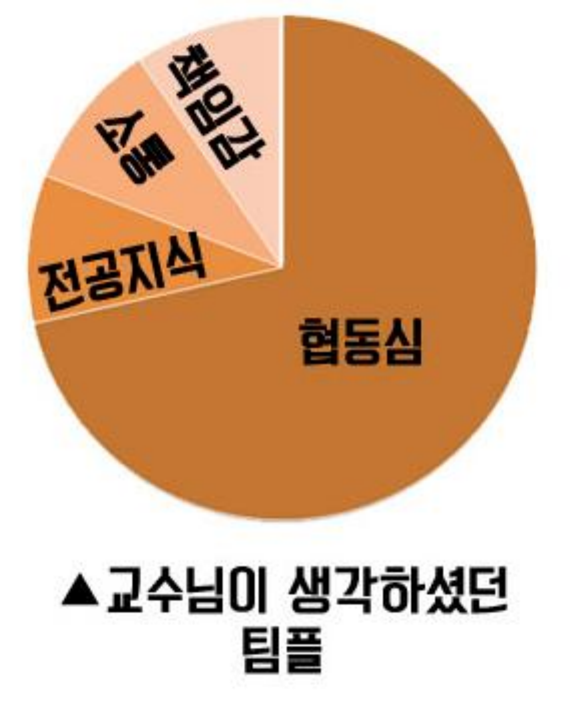
Lecture - 6

6

Lecture - 6
Project - 6

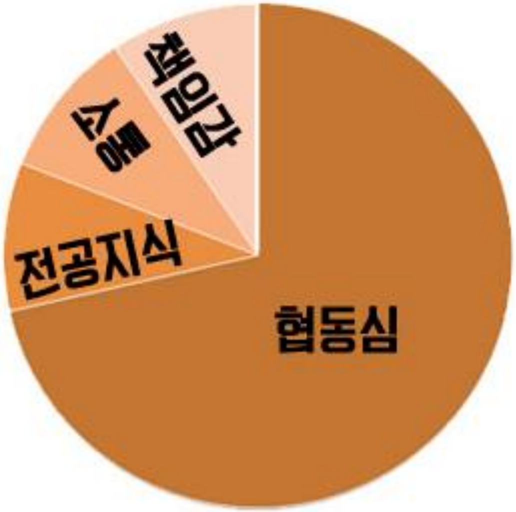
Term Project

Lecture - 6
Project - 6

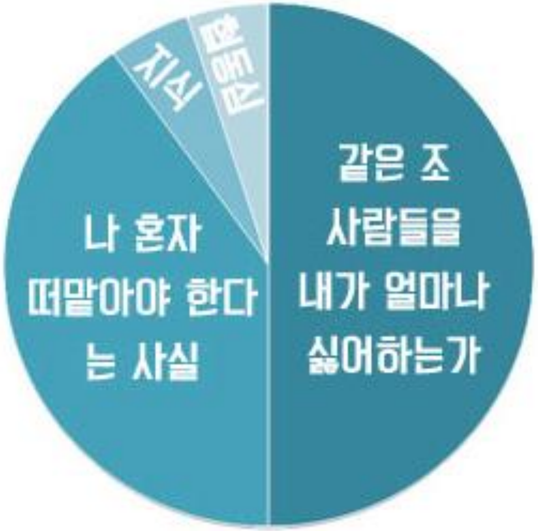


▲교수님이 생각하셨던 팀플

Lecture - 6
Project - 6



▲교수님이 생각하셨던 팀플



▲ 실제로 알게 되는 것

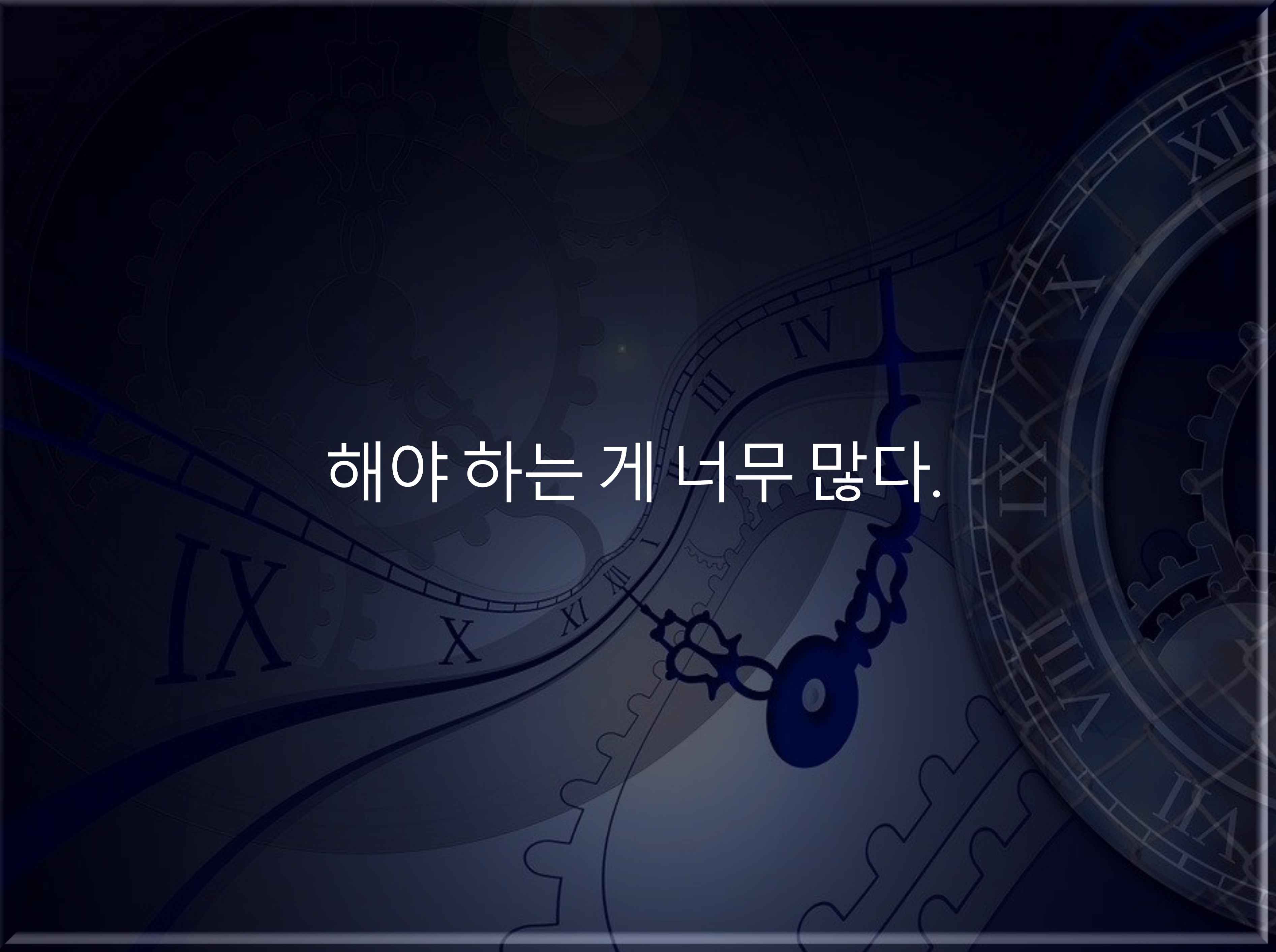
Lecture - 6
Project - 6

~~Term Project~~

Volunteer for others

Lecture - 6
Project - 6

4



해야 하는 게 너무 많다.

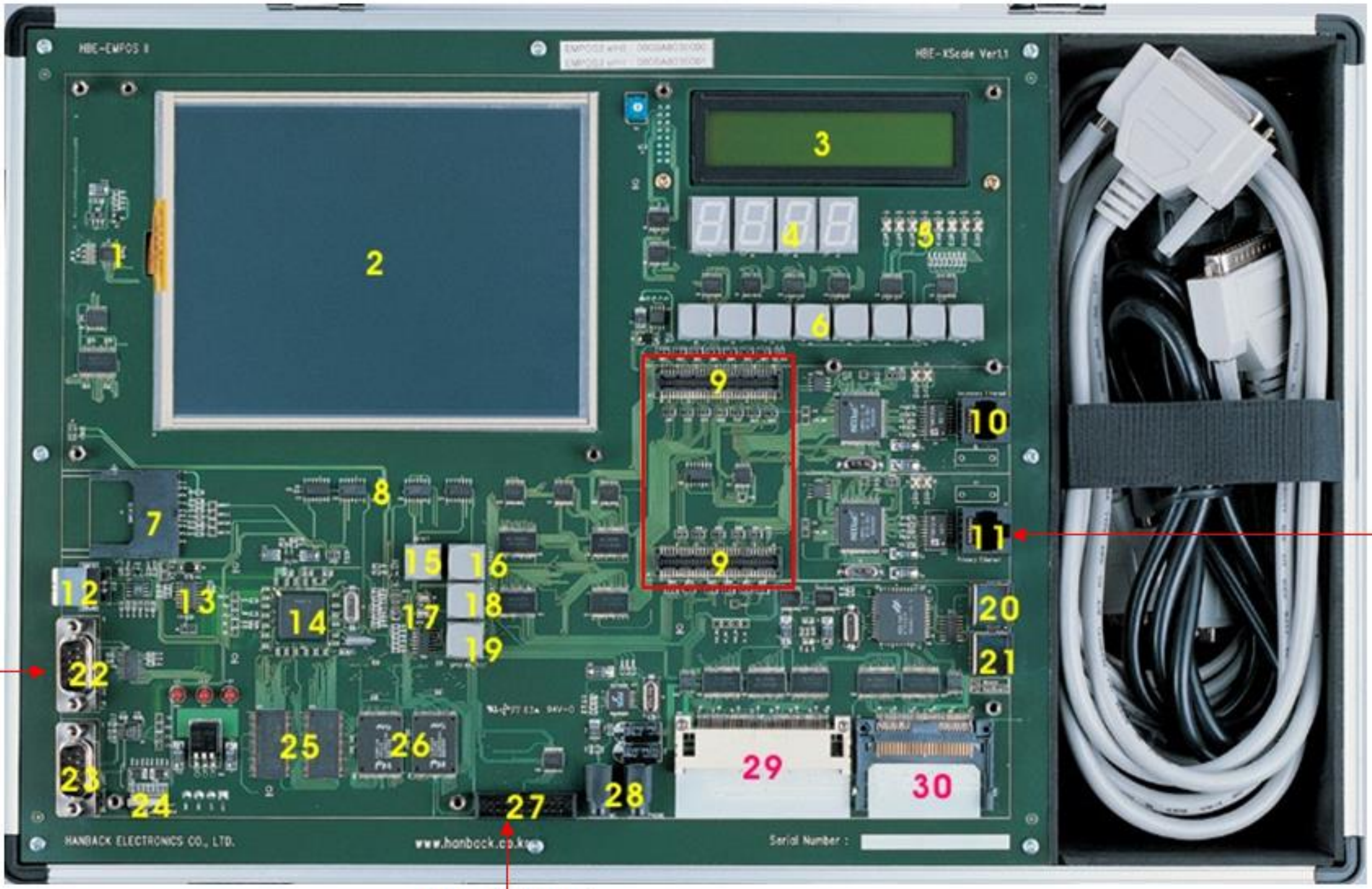
캐리해야 하는 게 너무 많다.

그래도…

해보자...orz

Embedded Linux

- Topic Lecture
  - Embedded Linux
  - **CPU: PXA 255(Xscale) : Embedded Linux-Based on HW**
  - **Target Board: HBE-EMPOS-II**
  - **Boot Loader**: blob
  - Development Environment: Tool Chains, JTAG
  - **Embedded Linux 개발론**
  - Kernel image
  - **Device Drivers**
  - RAM disk: busybox
  - **Applications: Web Server, mini Browser, Audio/video**
  - **GUI: Qt/E, Qutopia**
- Experiment
- Project

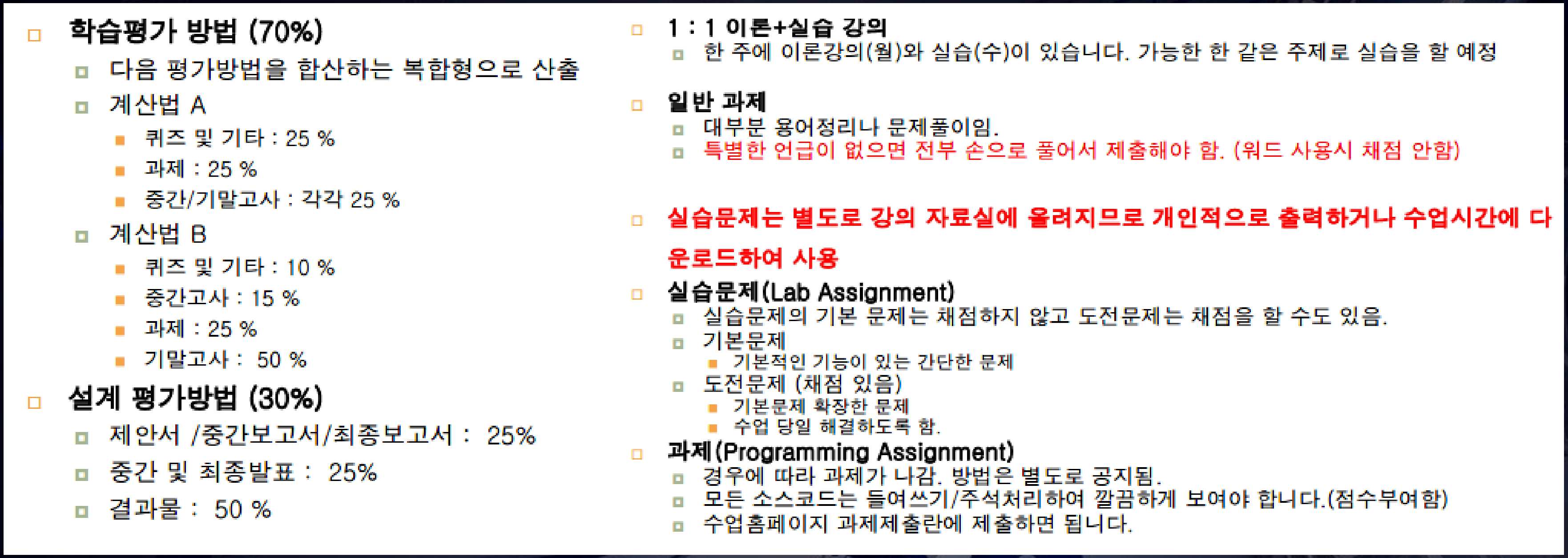


Ethernet
LAN port

Com1
Console
port

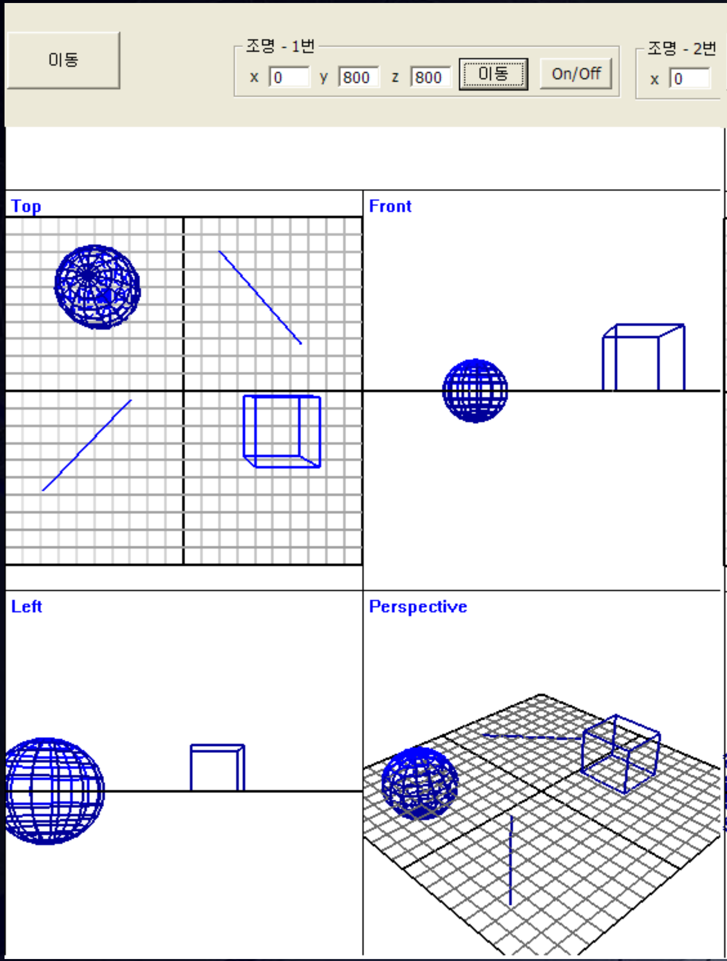
Parallel port
JTAG interface

Graphics



- **학습평가 방법 (70%)**
  - 다음 평가방법을 합산하는 복합형으로 산출
  - 계산법 A
    - 퀴즈 및 기타 : 25 %
    - 과제 : 25 %
    - 중간/기말고사 : 각각 25 %
  - 계산법 B
    - 퀴즈 및 기타 : 10 %
    - 중간고사 : 15 %
    - 과제 : 25 %
    - 기말고사 : 50 %
- **설계 평가방법 (30%)**
  - 제안서 /중간보고서/최종보고서 : 25%
  - 중간 및 최종발표 : 25%
  - 결과물 : 50 %

- **1 : 1 이론+실습 강의**
  - 한 주에 이론강의(월)와 실습(수)이 있습니다. 가능한 한 같은 주제로 실습을 할 예정
- **일반 과제**
  - 대부분 용어정리나 문제풀이임.
  - 특별한 언급이 없으면 전부 손으로 풀어서 제출해야 함. (워드 사용시 채점 안함)
- **실습문제는 별도로 강의 자료실에 올려지므로 개인적으로 출력하거나 수업시간에 다운로드하여 사용**
- **실습문제(Lab Assignment)**
  - 실습문제의 기본 문제는 채점하지 않고 도전문제는 채점을 할 수도 있음.
  - 기본문제
    - 기본적인 기능이 있는 간단한 문제
  - 도전문제 (채점 있음)
    - 기본문제 확장한 문제
    - 수업 당일 해결하도록 함.
- **과제(Programming Assignment)**
  - 경우에 따라 과제가 나감. 방법은 별도로 공지됨.
  - 모든 소스코드는 들여쓰기/주석처리하여 깔끔하게 보여야 합니다.(점수부여함)
  - 수업홈페이지 과제제출란에 제출하면 됩니다.

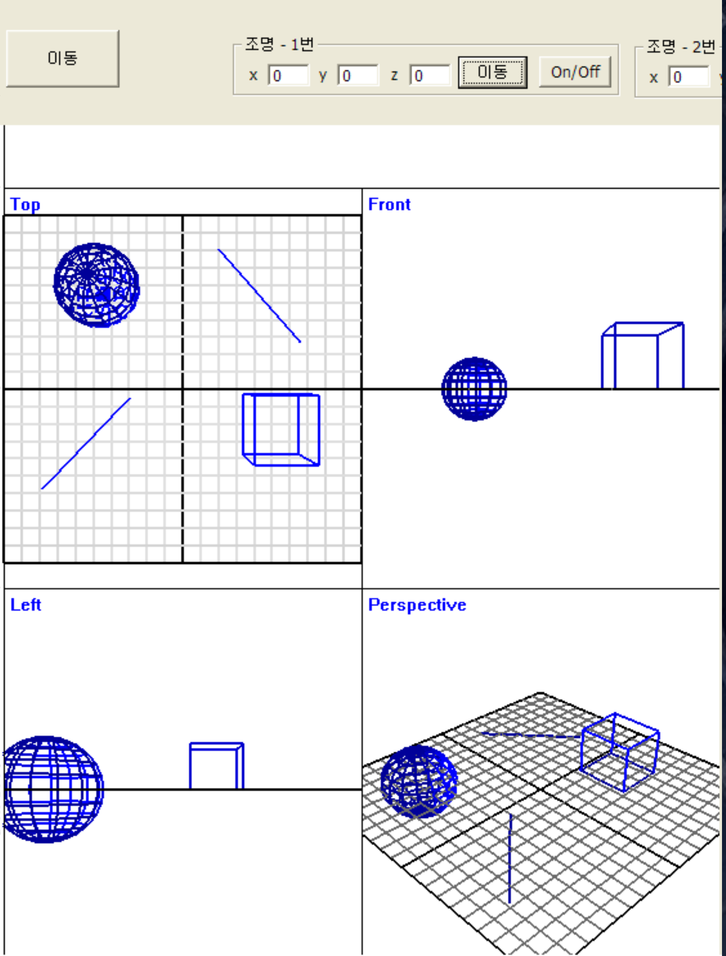


조명 - 1번

x 0 y 800 z 800

조명 - 2번

x 0



조명 - 1번

x 0 y 0 z 0

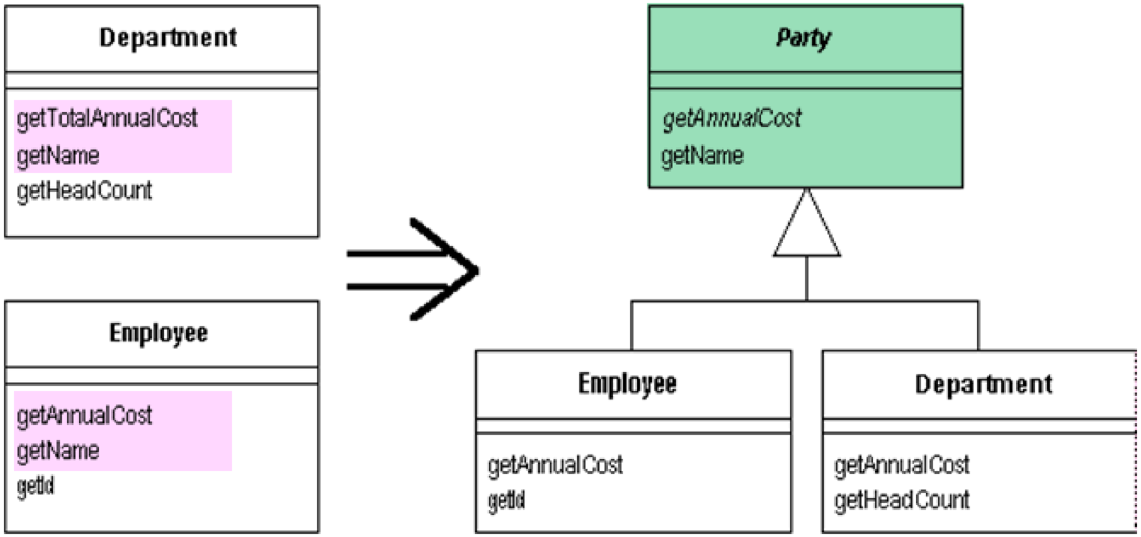
조명 - 2번

x 0

Design for Software System

Extract Superclass

- ***Bad Smell :*** *You have two classes with similar features*
- ***Solution :*** *Create a superclass and move the common features to the superclass*



Department

getTotalAnnualCost
getName
getHeadCount

Employee

getAnnualCost
getName
getId

⇒

Party

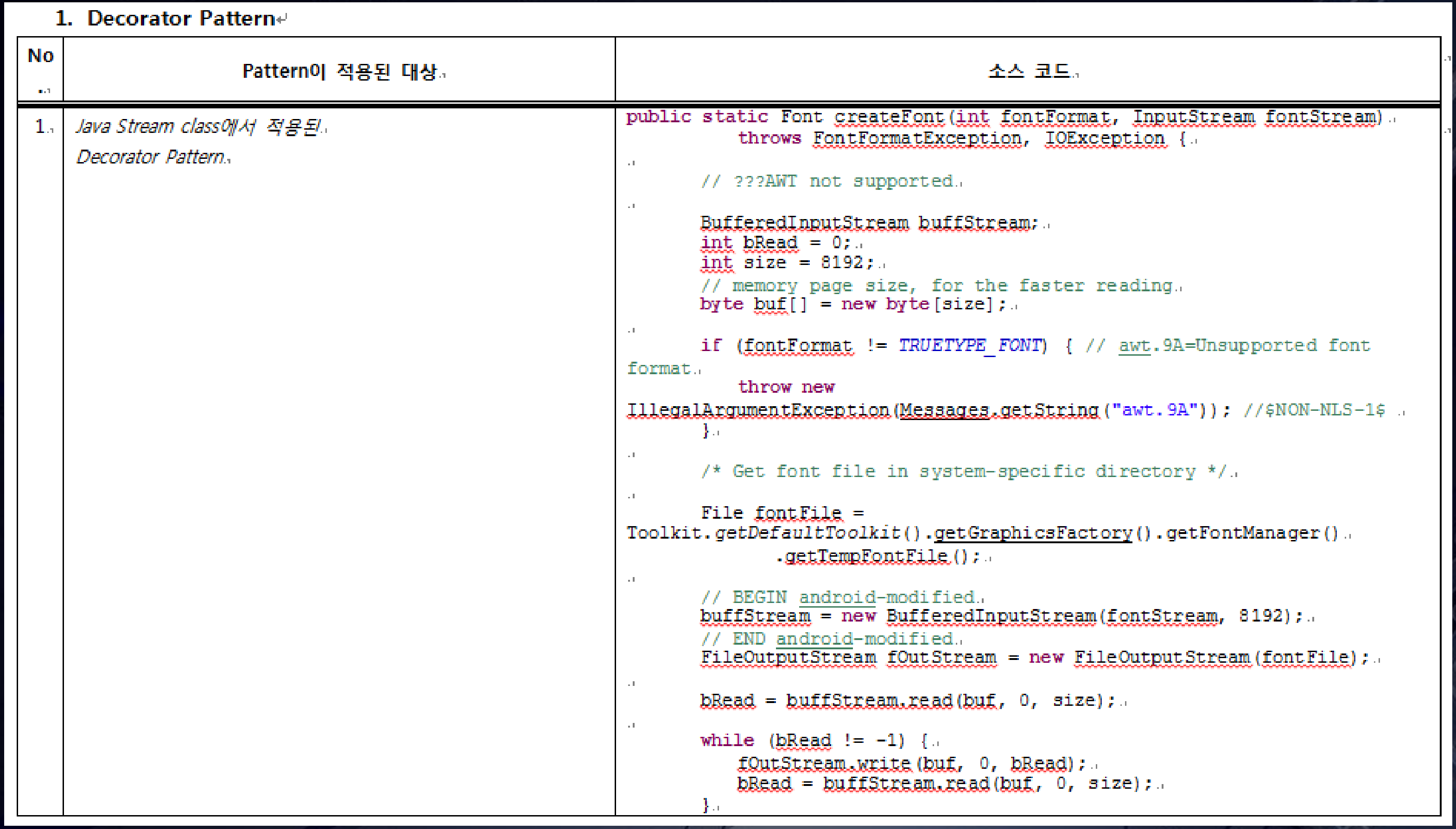
getAnnualCost
getName

Employee

getAnnualCost
getId

Department

getAnnualCost
getHeadCount



1. Decorator Pattern

No	Pattern이 적용된 대상	소스 코드
1	*Java Stream class에서 적용된* *Decorator Pattern*	

```java
public static Font createFont(int fontFormat, InputStream fontStream)
        throws FontFormatException, IOException {

    // ???AWT not supported

    BufferedInputStream buffStream;
    int bRead = 0;
    int size = 8192;
    // memory page size, for the faster reading
    byte buf[] = new byte[size];

    if (fontFormat != TRUETYPE_FONT) { // awt.9A=Unsupported font
format
        throw new
IllegalArgumentException(Messages.getString("awt.9A")); //$NON-NLS-1$
    }

    /* Get font file in system-specific directory */

    File fontFile =
Toolkit.getDefaultToolkit().getGraphicsFactory().getFontManager()
            .getTempFontFile();

    // BEGIN android-modified
    buffStream = new BufferedInputStream(fontStream, 8192);
    // END android-modified
    FileOutputStream fOutStream = new FileOutputStream(fontFile);

    bRead = buffStream.read(buf, 0, size);

    while (bRead != -1) {
        fOutStream.write(buf, 0, bRead);
        bRead = buffStream.read(buf, 0, size);
    }
```

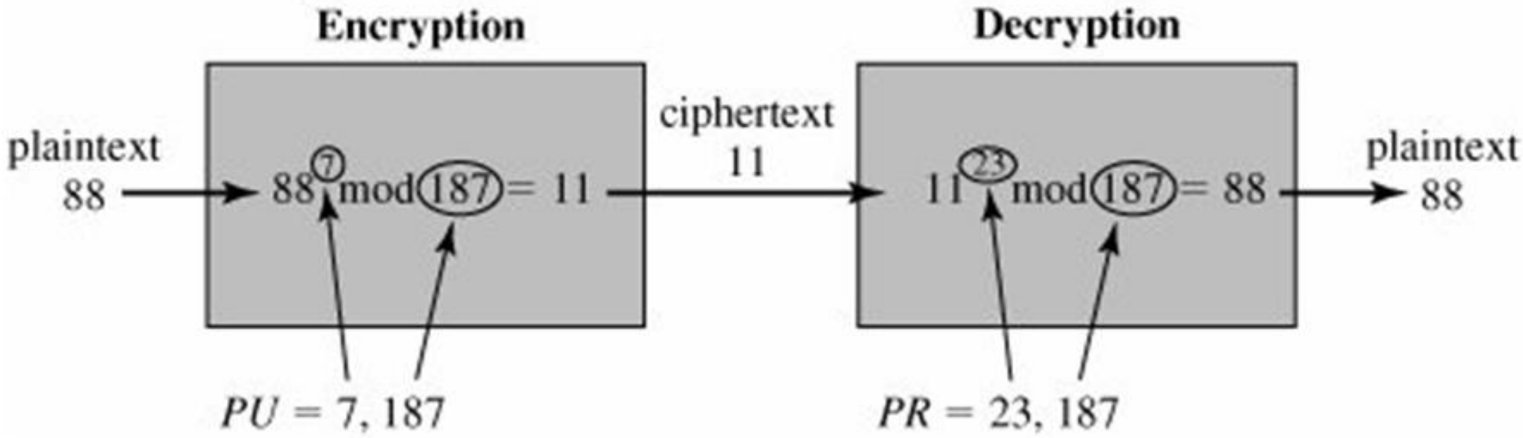
Computer Security

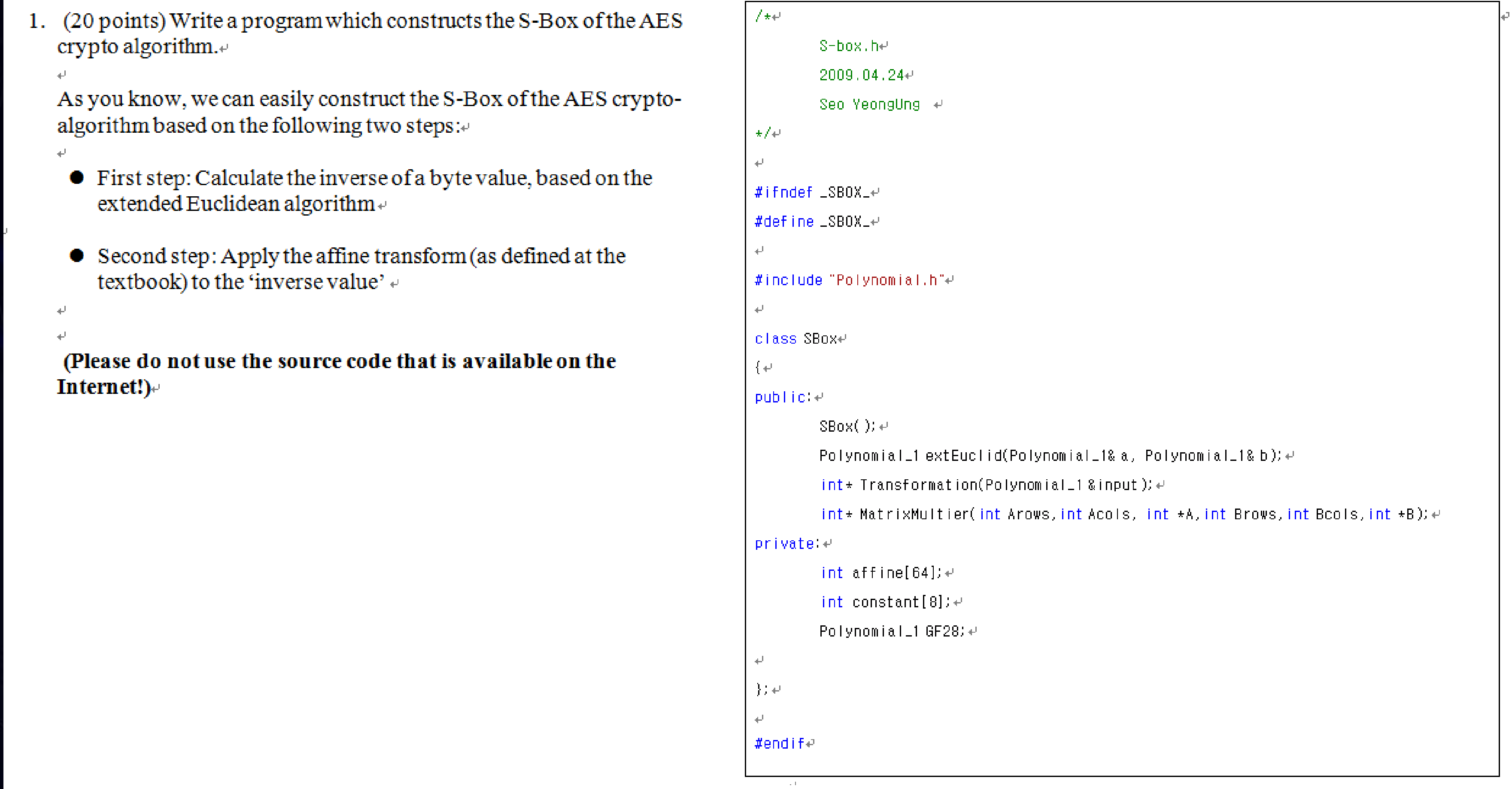
Example of RSA encryption/decryption.

- Key : `PU={7,187}, PR={23,17,11}`
- Given message `M = 88` (note, `88<187`)
- Encryption:

  $C = 88^{7} \bmod 187 = 11$
- Decryption:

  $M = 11^{23} \bmod 187 = 88$





1. (20 points) Write a program which constructs the S-Box of the AES crypto algorithm.

   As you know, we can easily construct the S-Box of the AES crypto-algorithm based on the following two steps:

   - First step: Calculate the inverse of a byte value, based on the extended Euclidean algorithm
   - Second step: Apply the affine transform (as defined at the textbook) to the 'inverse value'

   (Please do not use the source code that is available on the Internet!)

```
/*
	S-box.h
	2009.04.24
	Seo YeongUng
*/

#ifndef _SBOX_
#define _SBOX_

#include "Polynomial.h"

class SBox
{
public:
	SBox( );
	Polynomial_1 extEuclid(Polynomial_1& a, Polynomial_1& b);
	int* Transformation(Polynomial_1 &input );
	int* MatrixMultier( int Arows, int Acols, int *A, int Brows, int Bcols, int *B);
private:
	int affine[64];
	int constant[8];
	Polynomial_1 GF28;

};

#endif
```

Embedded Linux
Graphics
Design for Software System
Computer Security

Embedded Linux
Graphics
Design for Software System
Computer Security

+

Out of class project (robot)

Embedded Linux
Graphics
~~Design for Software System~~
~~Computer Security~~
Out of class project (robot)

Using Image Processing

침입자 감시 시스템
(for Embedded Linux)

침입자 감시 시스템

- 원거리에서 효과적으로 침입자를 감시하기 위한 시스템
- 감시를 위한 시스템은 고성능의 HW가 필요하지 않음
- Embeded System에 적합

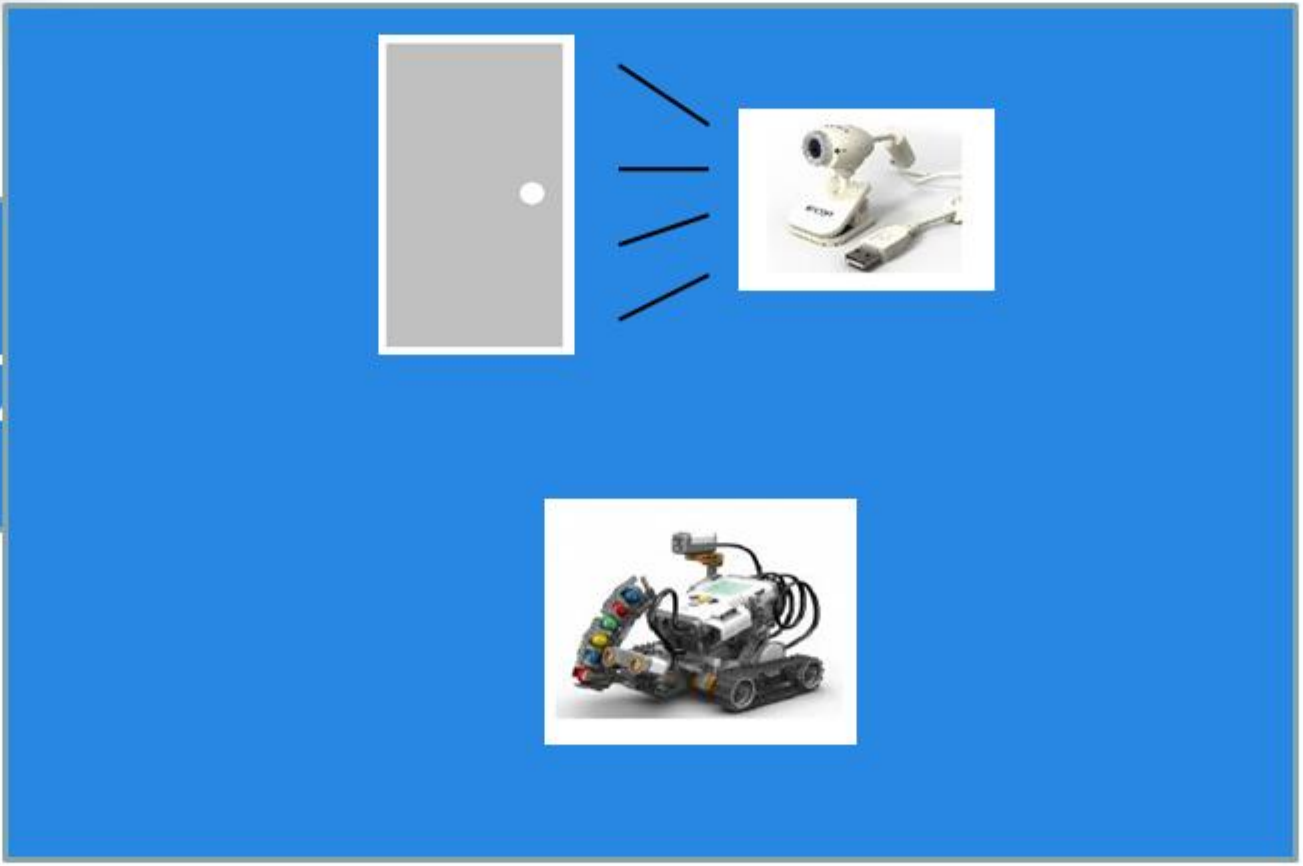
침입자 감시 시스템

경비실

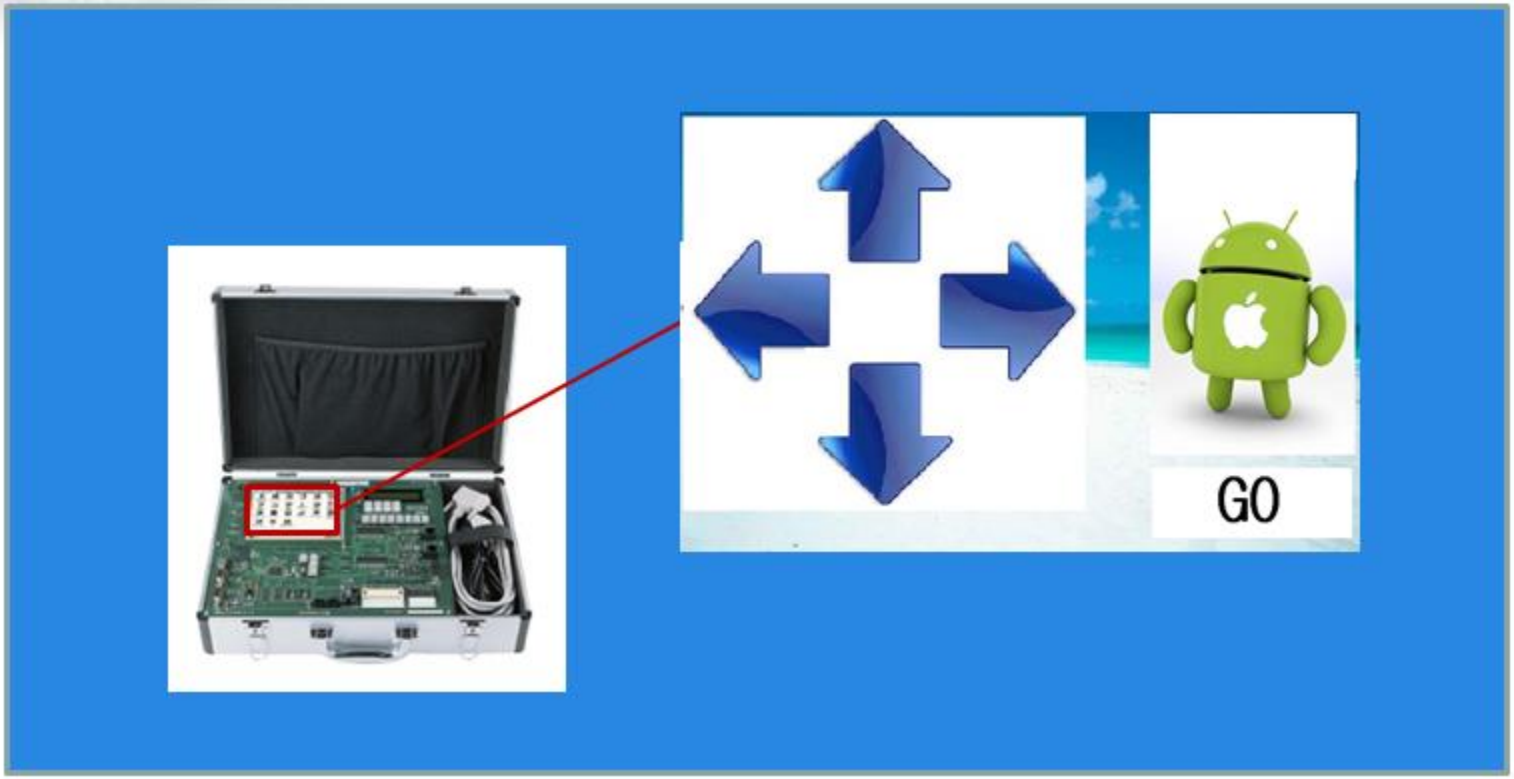
창고

침입자 감시 시스템

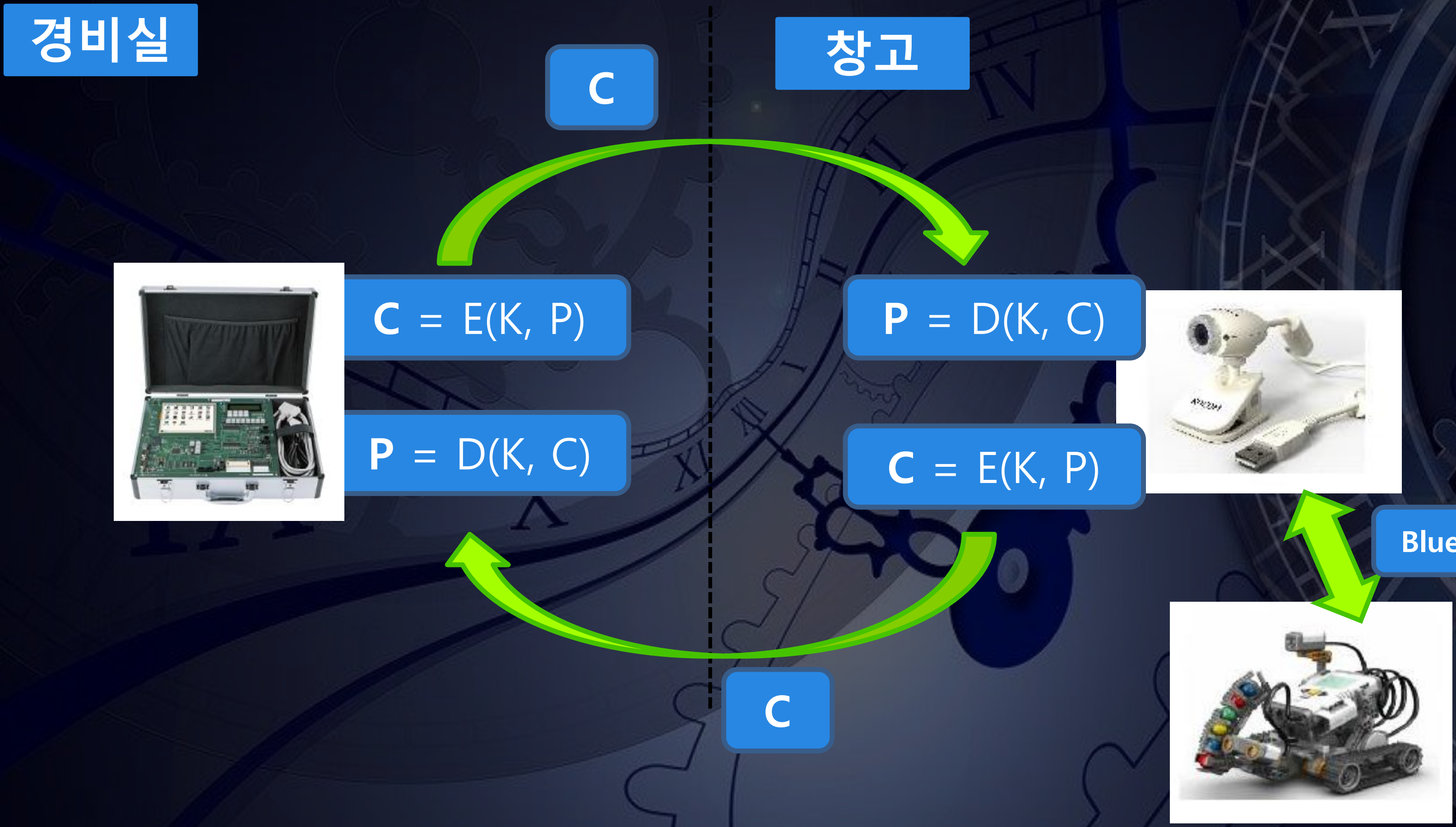
경



침입자 감시 시스템



침입자 감시 시스템



경비실

창고

C

C = E(K, P)

P = D(K, C)

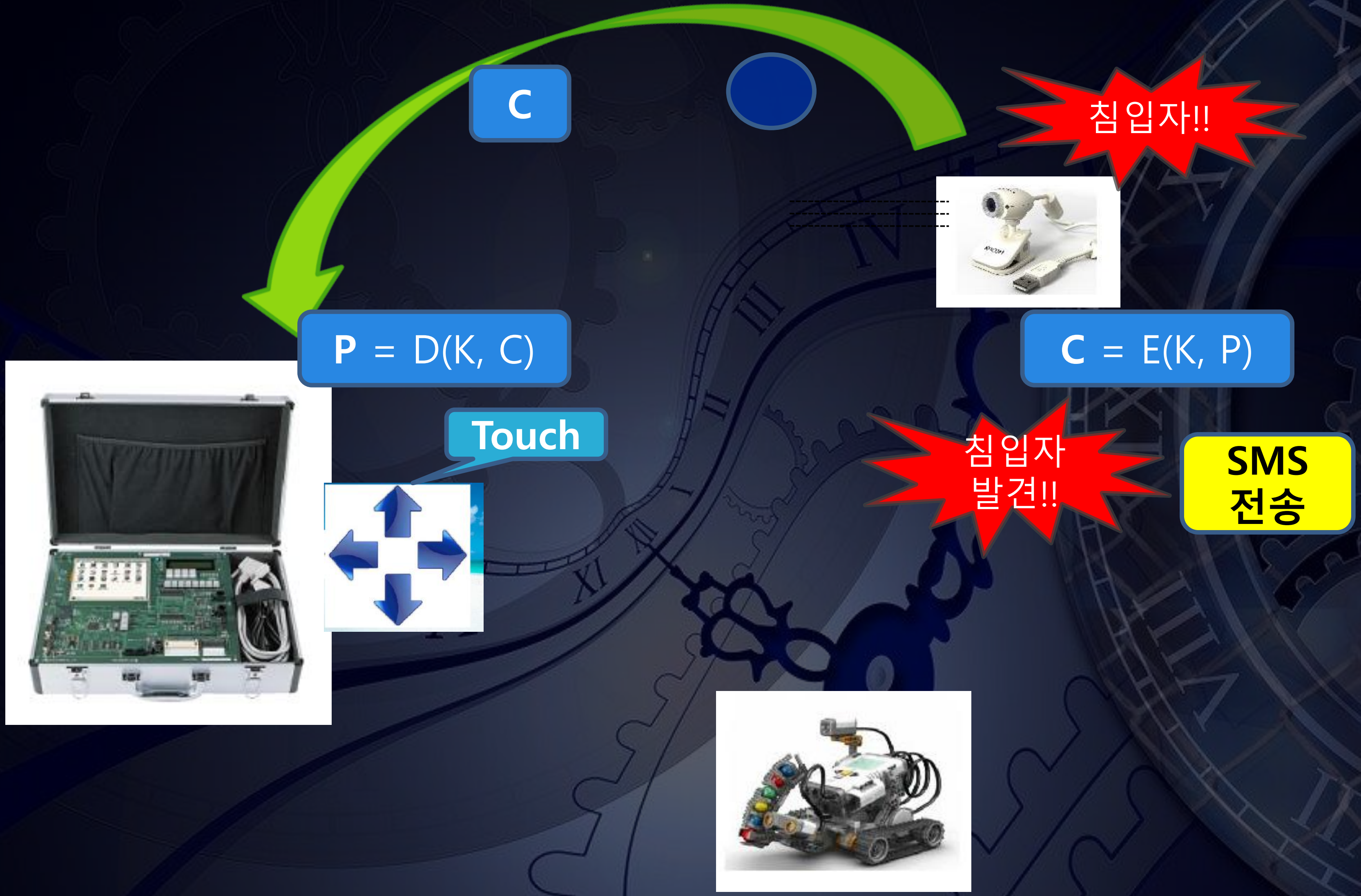
P = D(K, C)

C = E(K, P)

C

Bluetooth

시나리오



침입자!!

C = E(K, P)

C

P = D(K, C)

Touch

침입자 발견!!

SMS 전송

Warehouse

- 1. Object Recognition with WebCam
- 2. Robot Control using Bluetooth

Security Office

- 1. Real-Time Confirmation for Warehouse
- 2. Remote Robot Control

Common

- 1. Block Cipher(DES or AES) using CCM mode of operation
- 2. SMS

제약 사항

1. 카메라와 로봇 제어를 위한 PC

2. 카메라 인식을 위한 침입자 정의

3. Robot의 위치는 미리 안다고 가정

4. Key는 미리 분배되어 있다고 가정

추가 구현

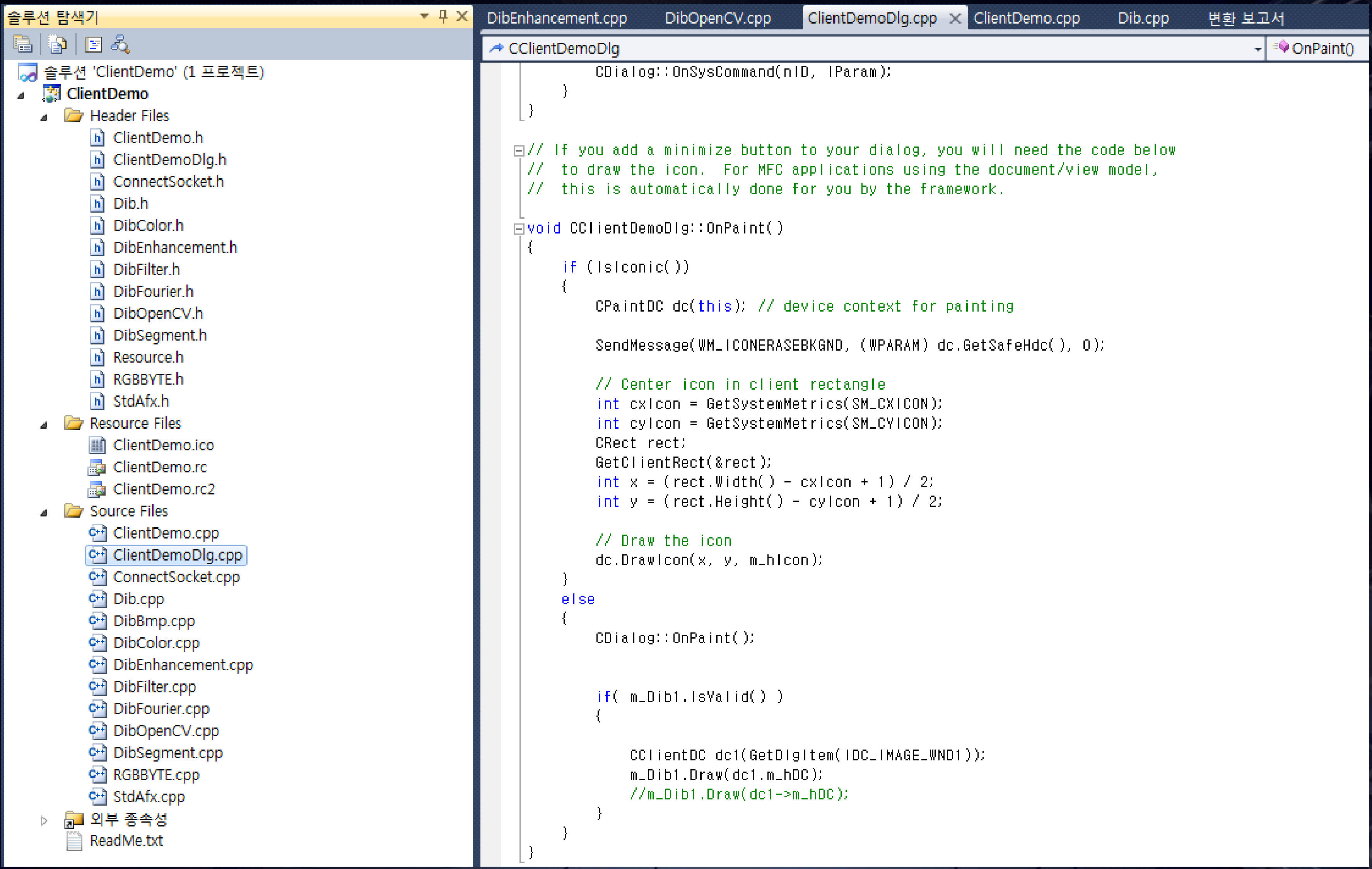
1. 실시간 창고 확인

WebCam의 영상을 EMPOS-2에서 확인

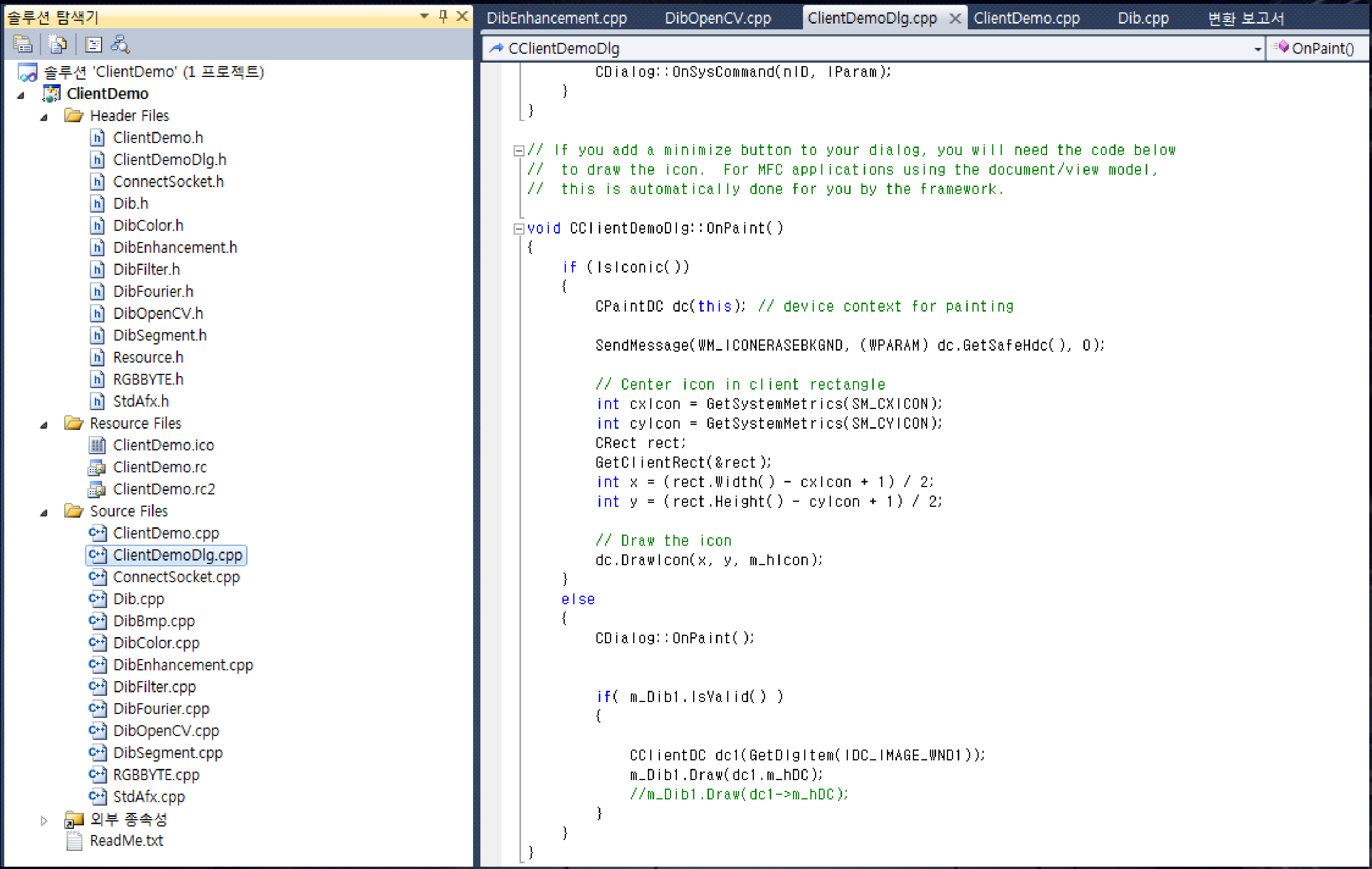
2. Public-Key Cryptography를 이용한 key distribution

RSA or ECC

코드

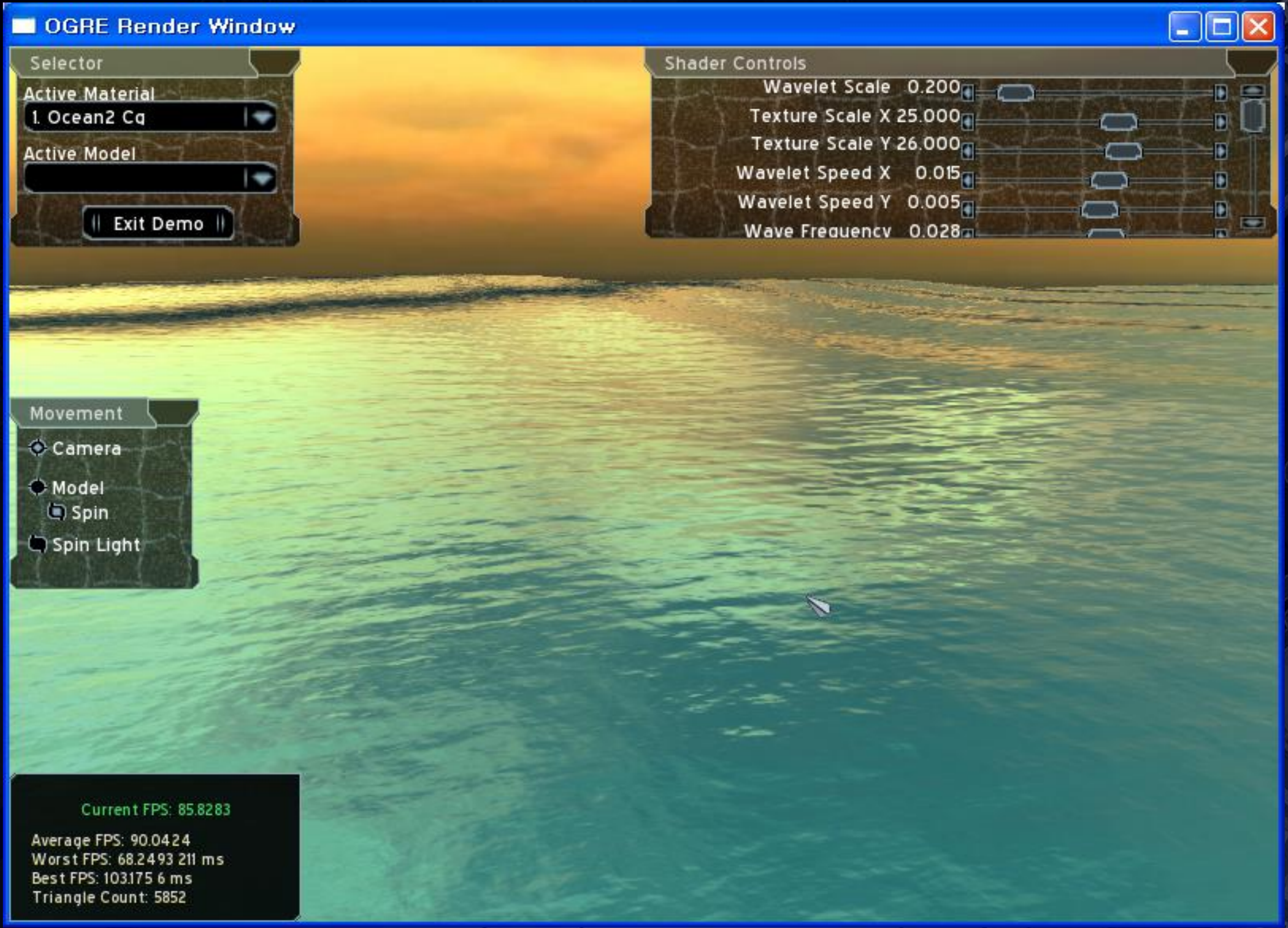


```
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
//  to draw the icon.  For MFC applications using the document/view model,
//  this is automatically done for you by the framework.

void CClientDemoDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();


        if( m_Dib1.IsValid() )
        {

            CClientDC dc1(GetDlgItem(IDC_IMAGE_WND1));
            m_Dib1.Draw(dc1.m_hDC);
            //m_Dib1.Draw(dc1->m_hDC);
        }
    }
}
```



```
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
//  to draw the icon.  For MFC applications using the document/view model,
//  this is automatically done for you by the framework.

void CClientDemoDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();


        if( m_Dib1.IsValid() )
        {

            CClientDC dc1(GetDlgItem(IDC_IMAGE_WND1));
            m_Dib1.Draw(dc1.m_hDC);
            //m_Dib1.Draw(dc1->m_hDC);
        }
    }
}
```

Orge3D를 이용한 AR - 증강현실
(for Graphics)

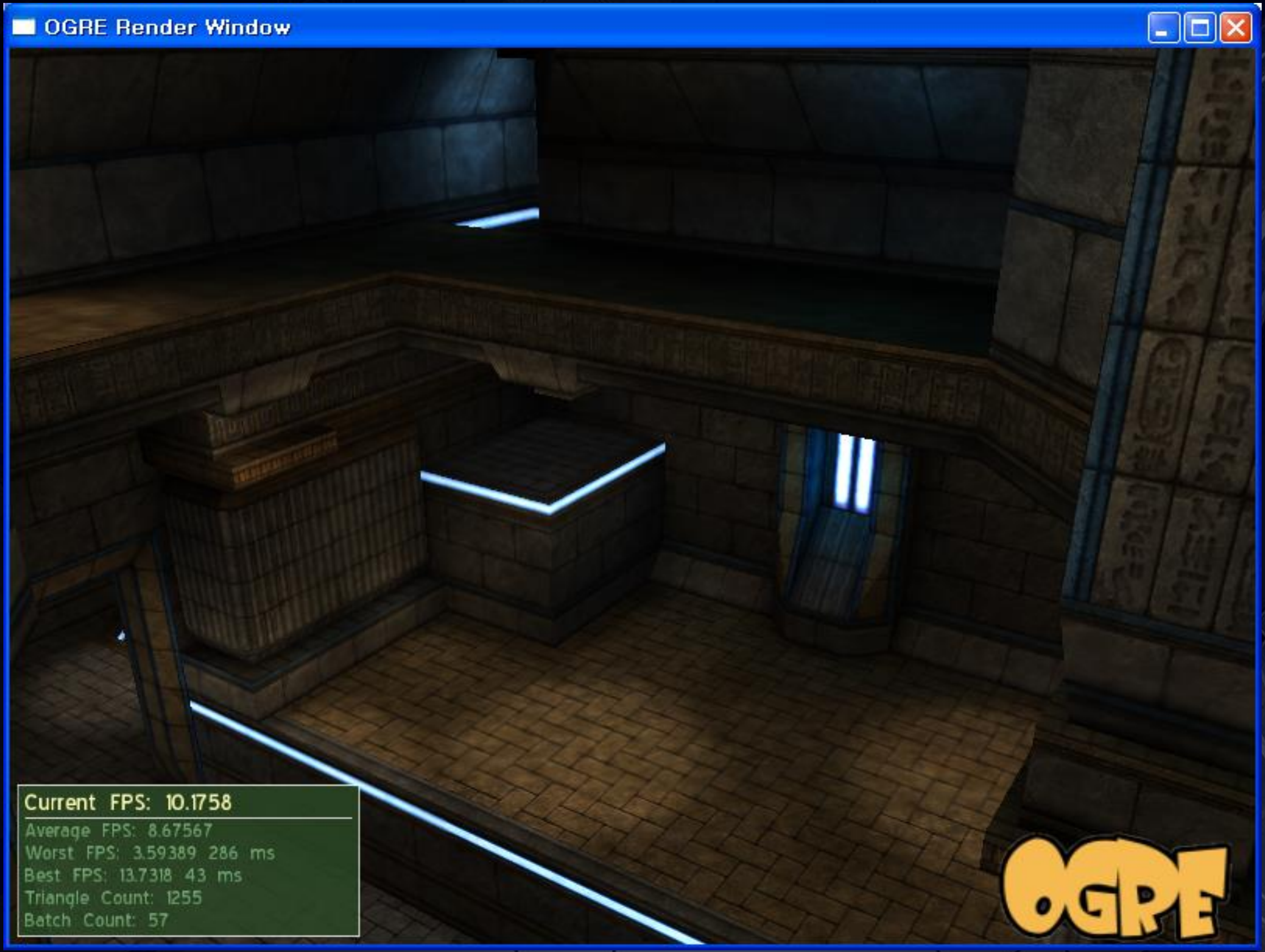


Active Material: 1. Ocean2 Cg

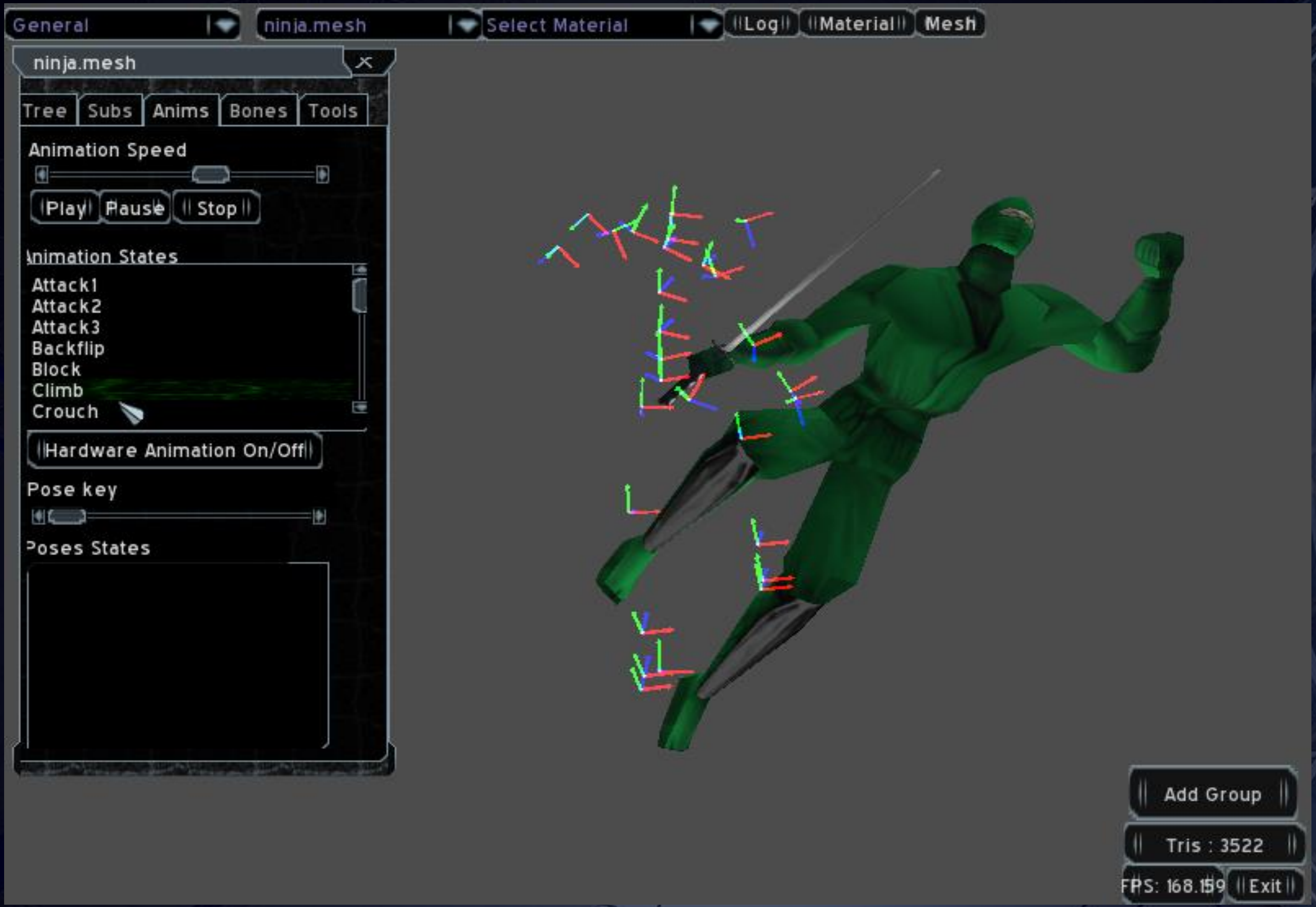
Active Model:

Wavelet Scale 0.200
Texture Scale X 25.000
Texture Scale Y 26.000
Wavelet Speed X 0.015
Wavelet Speed Y 0.005
Wave Frequency 0.028

```
Current FPS: 85.8283

Average FPS: 90.0424
Worst FPS: 68.2493 211 ms
Best FPS: 103.175 6 ms
Triangle Count: 5852
```



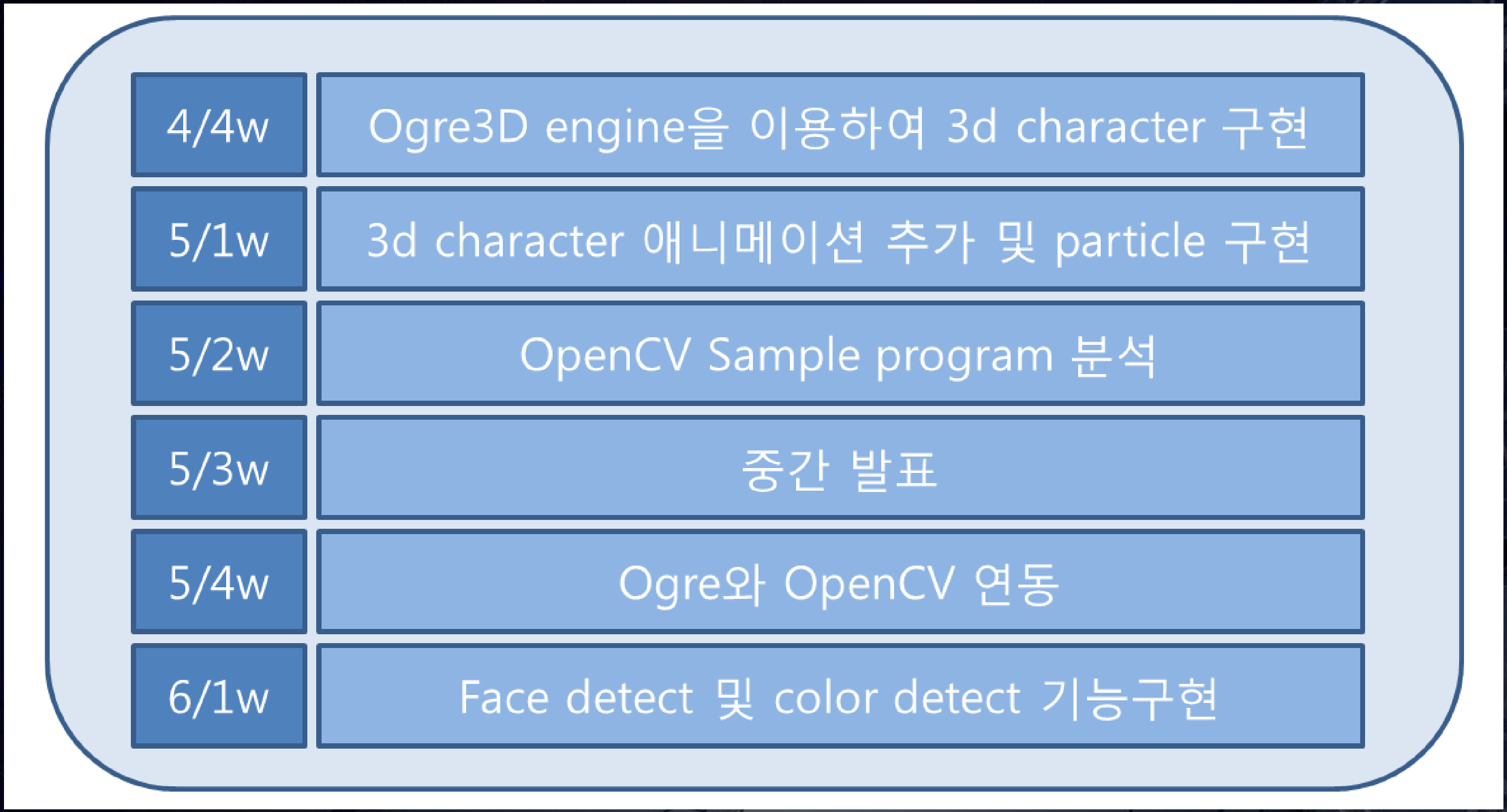
```
Current FPS: 10.1758
Average FPS: 8.67567
Worst FPS: 3.59389 286 ms
Best FPS: 13.7318 43 ms
Triangle Count: 1255
Batch Count: 57
```


General
ninja.mesh
Select Material
Log
Material
Mesh
ninja.mesh
Tree
Subs
Anims
Bones
Tools
Animation Speed
Play
Pause
Stop
Animation States
Attack1
Attack2
Attack3
Backflip
Block
Climb
Crouch
Hardware Animation On/Off
Pose key
Poses States
Add Group
Tris : 3522
FPS: 168.159
Exit

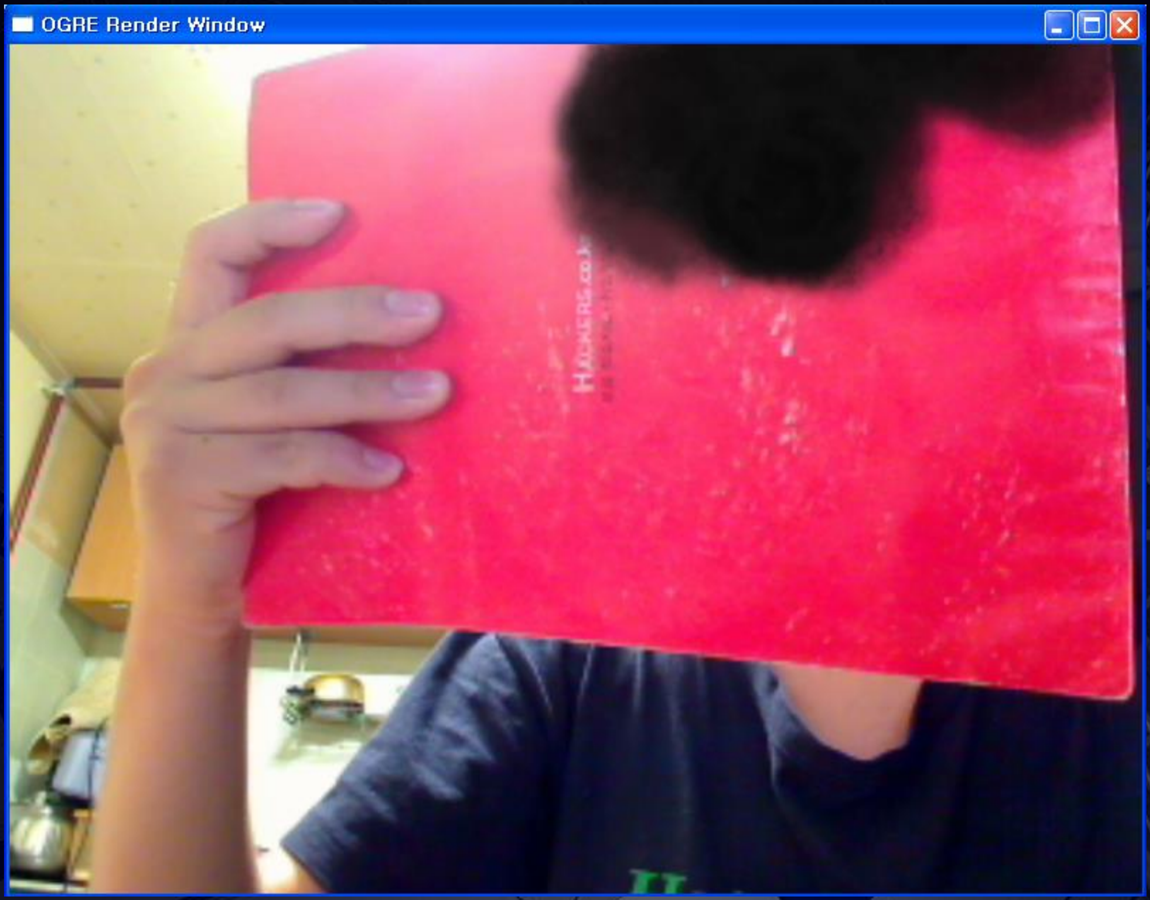
닌자의 모험 Using AR

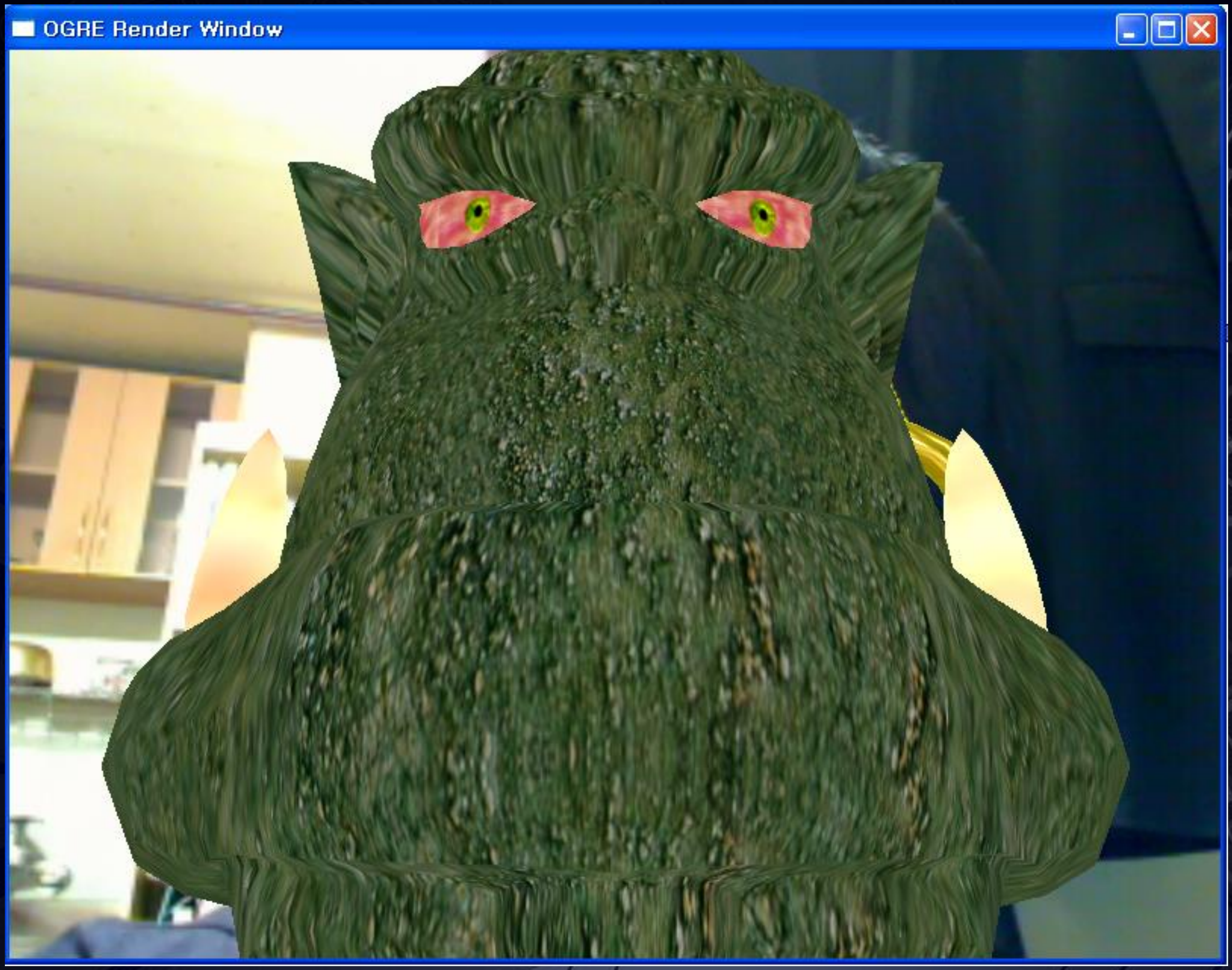
~~닌자의 모험 Using AR~~
영상처리를 이용한 AR 구현

혼자 게임 언제 만들... 읍읍

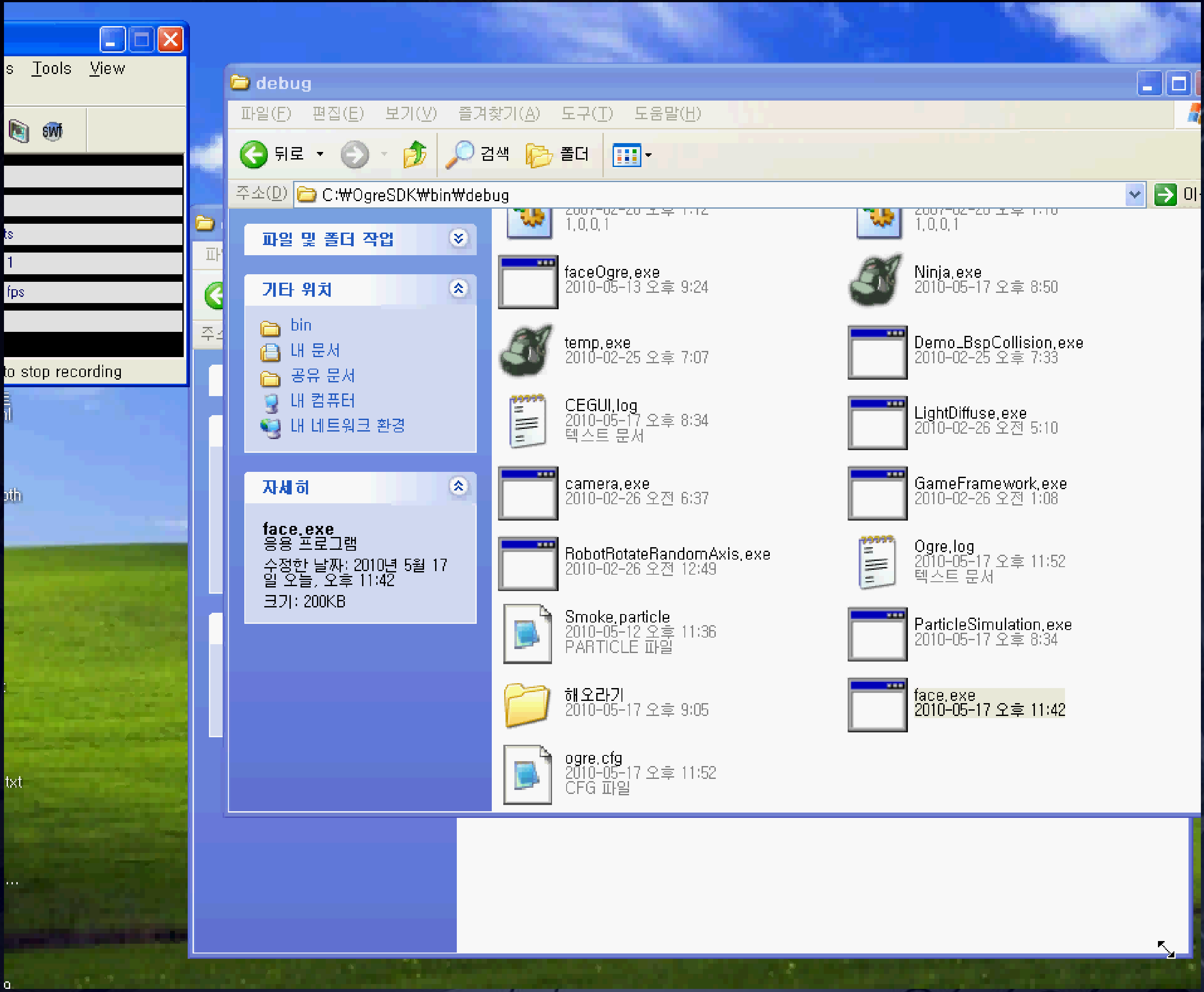


4/4w	Ogre3D engine을 이용하여 3d character 구현
5/1w	3d character 애니메이션 추가 및 particle 구현
5/2w	OpenCV Sample program 분석
5/3w	중간 발표
5/4w	Ogre와 OpenCV 연동
6/1w	Face detect 및 color detect 기능구현


OGRE Render Window
HACKERS.co.kr


OGRE Render Window

시연 (동영상)

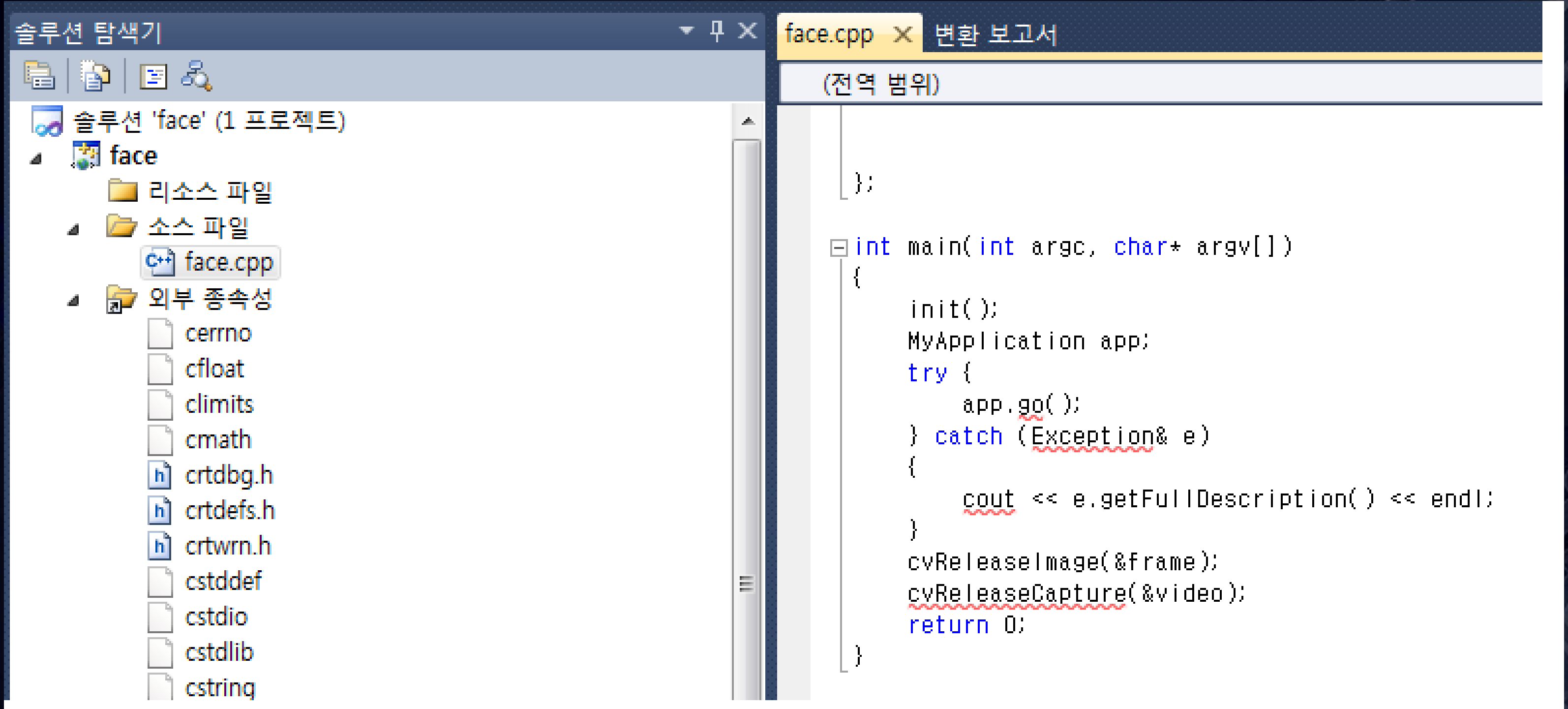


C:\OgreSDK\bin\debug

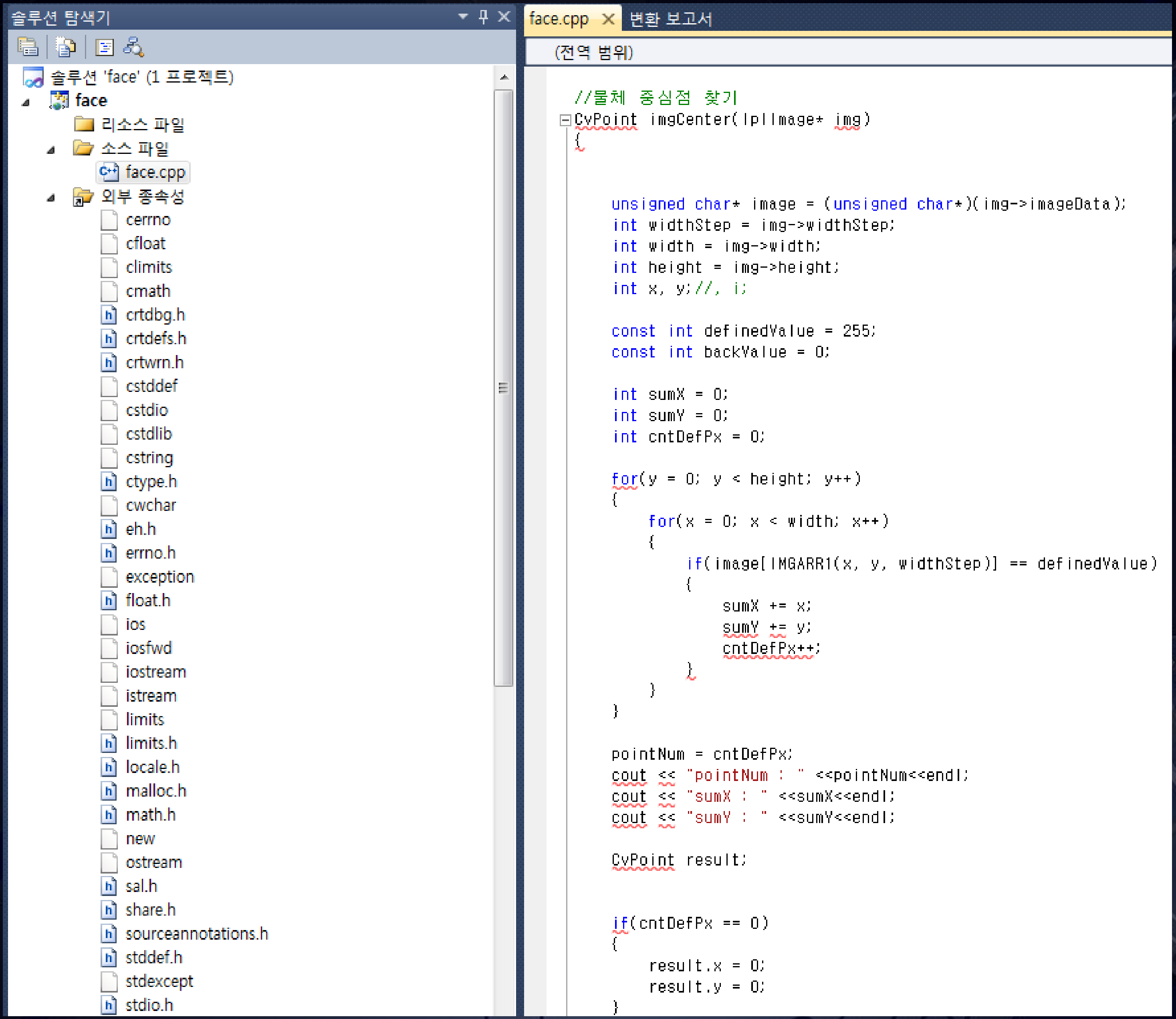
face.exe
응용 프로그램
수정한 날짜: 2010년 5월 17일 오늘, 오후 11:42
크기: 200KB

```
1,0,0,1                                   1,0,0,1
faceOgre.exe                              Ninja.exe
2010-05-13 오후 9:24                       2010-05-17 오후 8:50
temp.exe                                  Demo_BspCollision.exe
2010-02-25 오후 7:07                       2010-02-25 오후 7:33
CEGUI.log                                 LightDiffuse.exe
2010-05-17 오후 8:34                       2010-02-26 오전 5:10
텍스트 문서
camera.exe                                GameFramework.exe
2010-02-26 오전 6:37                       2010-02-26 오전 1:08
RobotRotateRandomAxis.exe                 Ogre.log
2010-02-26 오전 12:49                      2010-05-17 오후 11:52
                                          텍스트 문서
Smoke.particle                            ParticleSimulation.exe
2010-05-12 오후 11:36                      2010-05-17 오후 8:34
PARTICLE 파일
해오라기                                    face.exe
2010-05-17 오후 9:05                       2010-05-17 오후 11:42
ogre.cfg
2010-05-17 오후 11:52
CFG 파일
```

코드



```
};

int main( int argc, char* argv[] )
{
    init( );
    MyApplication app;
    try {
        app.go( );
    } catch (Exception& e)
    {
        cout << e.getFullDescription( ) << endl;
    }
    cvReleaseImage( &frame );
    cvReleaseCapture( &video );
    return 0;
}
```



```
//물체 중심점 찾기
CvPoint imgCenter(IplImage* img)
{


    unsigned char* image = (unsigned char*)(img->imageData);
    int widthStep = img->widthStep;
    int width = img->width;
    int height = img->height;
    int x, y;//, i;

    const int definedValue = 255;
    const int backValue = 0;

    int sumX = 0;
    int sumY = 0;
    int cntDefPx = 0;

    for(y = 0; y < height; y++)
    {
        for(x = 0; x < width; x++)
        {
            if(image[IMGARR1(x, y, widthStep)] == definedValue)
            {
                sumX += x;
                sumY += y;
                cntDefPx++;
            }
        }
    }

    pointNum = cntDefPx;
    cout << "pointNum : " <<pointNum<<endl;
    cout << "sumX : " <<sumX<<endl;
    cout << "sumY : " <<sumY<<endl;

    CvPoint result;


    if(cntDefPx == 0)
    {
        result.x = 0;
        result.y = 0;
    }
```

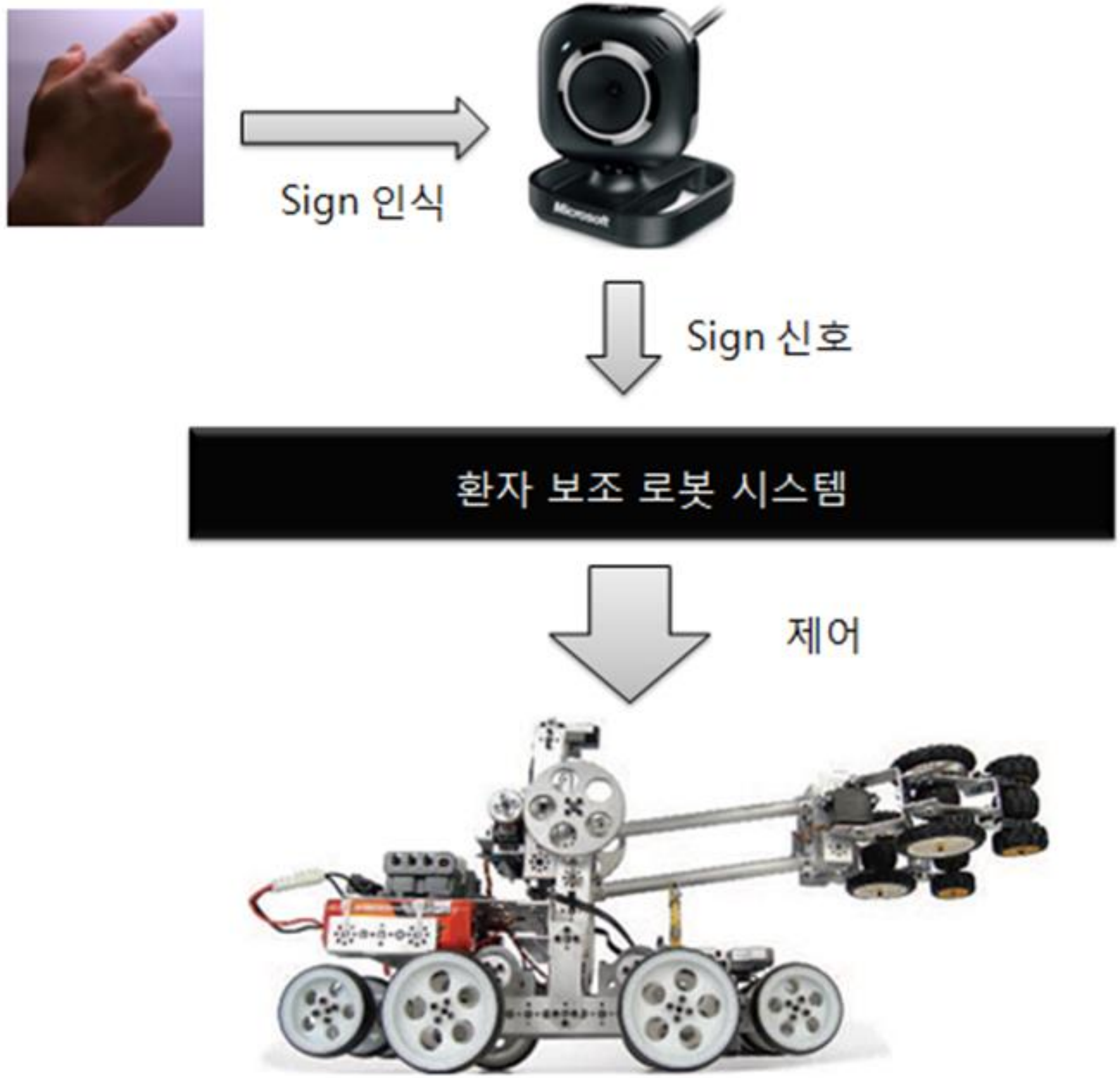
영상처리를 이용한 환자보조로봇
(for Out of class project - robot)

환자보조로봇

거동이 불편한 환자를 보조하기 위한 로봇 시스템

Part

1. 환자의 손동작을 카메라로 인식하여 메시지를 전달하는 부분
2. 로봇이 물체를 판별하는 부분
3. 실제 로봇을 제어하는 부분



Sign 인식

Sign 신호

환자 보조 로봇 시스템

제어

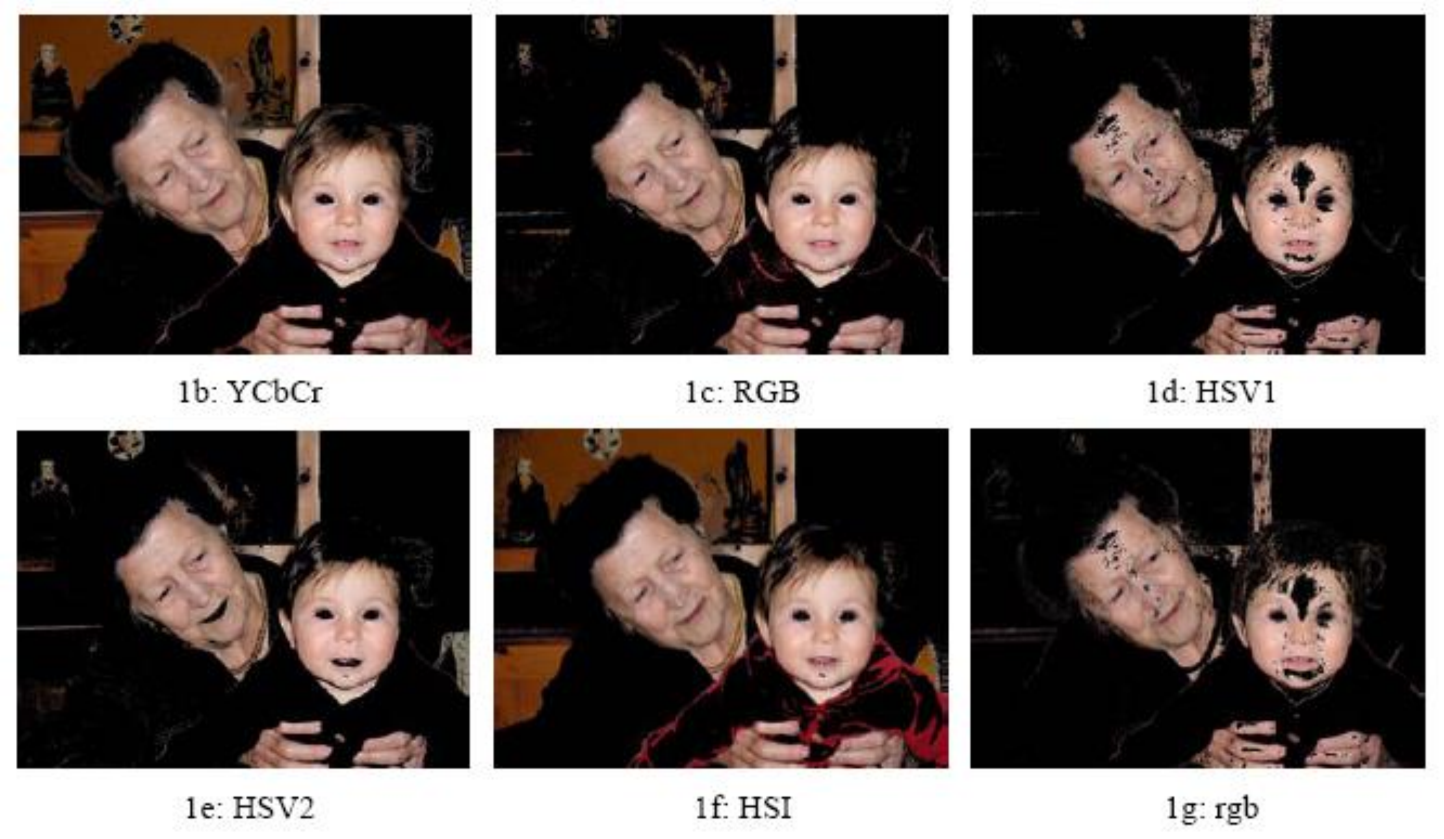
Part1 - 손 동작 인식

살색 검출

1a

F. Gasparini, R. Schettini, "Skin segmentation using multiple thresholding,"SPIE proceedings, pp. 6061-18, 2006

살색 검출

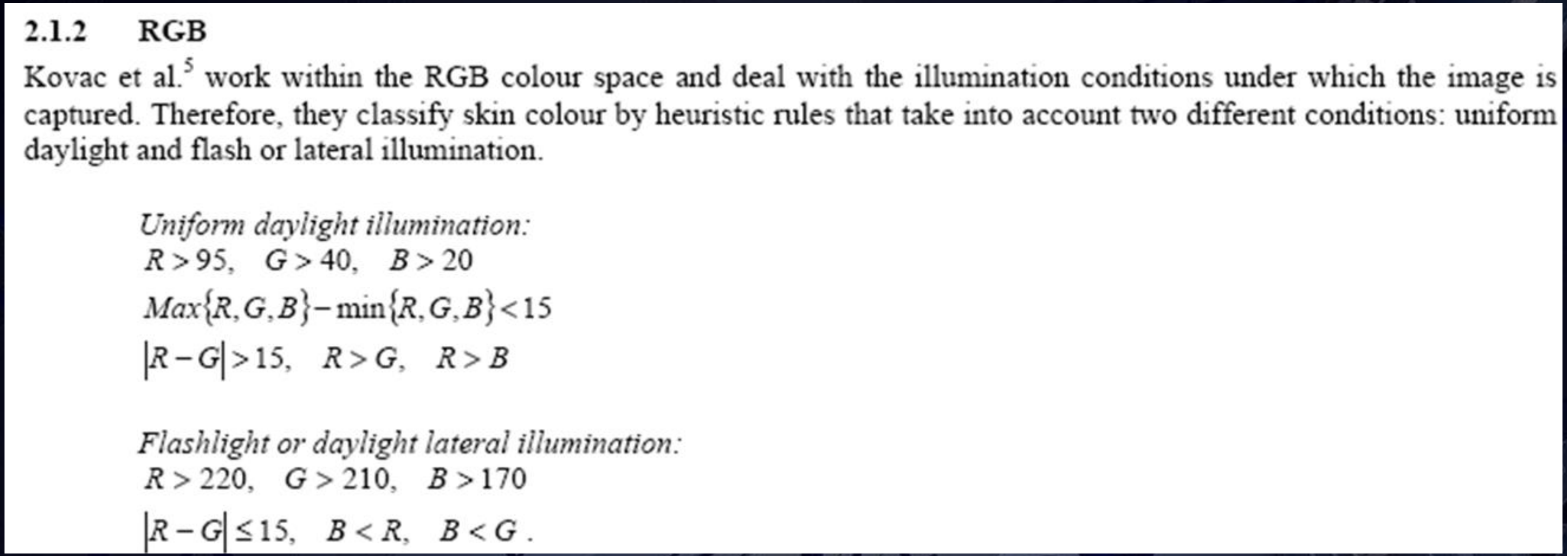


1b: YCbCr 1c: RGB 1d: HSV1

1e: HSV2 1f: HSI 1g: rgb

F. Gasparini, R. Schettini, "Skin segmentation using multiple thresholding,"SPIE proceedings, pp. 6061-18, 2006

살색 검출 - RGB



2.1.2 RGB

Kovac et al.[5] work within the RGB colour space and deal with the illumination conditions under which the image is captured. Therefore, they classify skin colour by heuristic rules that take into account two different conditions: uniform daylight and flash or lateral illumination.

Uniform daylight illumination:

$R > 95, \quad G > 40, \quad B > 20$

$Max\{R, G, B\} - \min\{R, G, B\} < 15$

$|R - G| > 15, \quad R > G, \quad R > B$

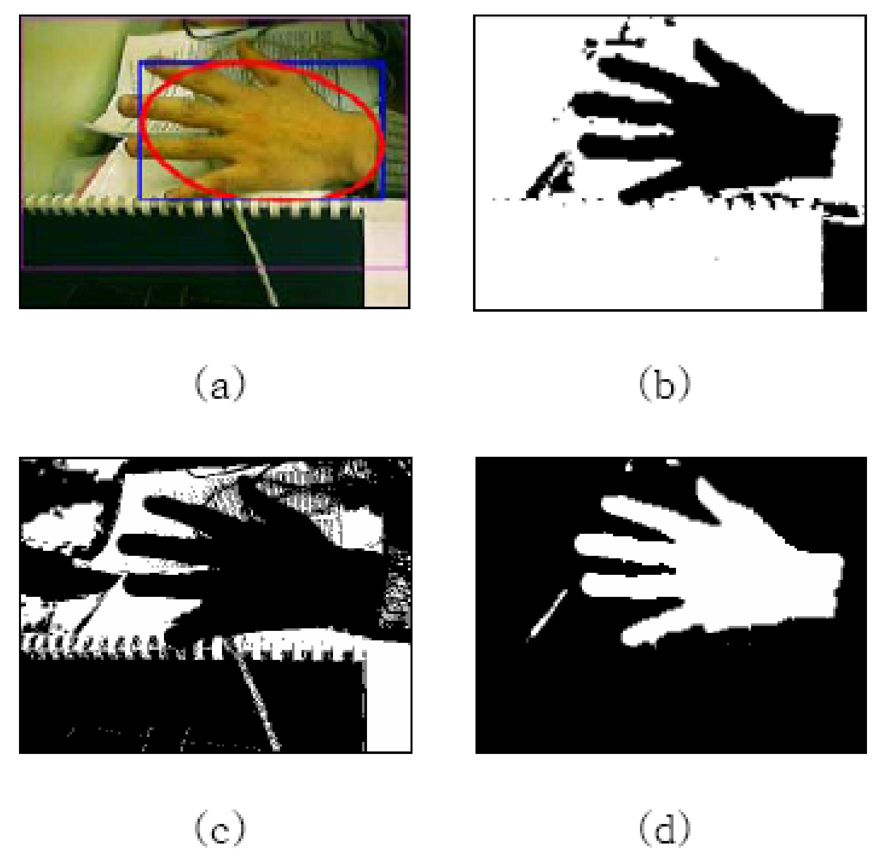
Flashlight or daylight lateral illumination:

$R > 220, \quad G > 210, \quad B > 170$

$|R - G| \leq 15, \quad B < R, \quad B < G$.

F. Gasparini, R. Schettini, "Skin segmentation using multiple thresholding,"SPIE proceedings, pp. 6061-18, 2006

살색 검출 - HSI



Taejin Ha, Woontack Woo, "Video see-through HMD based Hand Interface for Augmented Reality," HCI 2006, vol. 1, pp.169-174, 2006

살색 검출

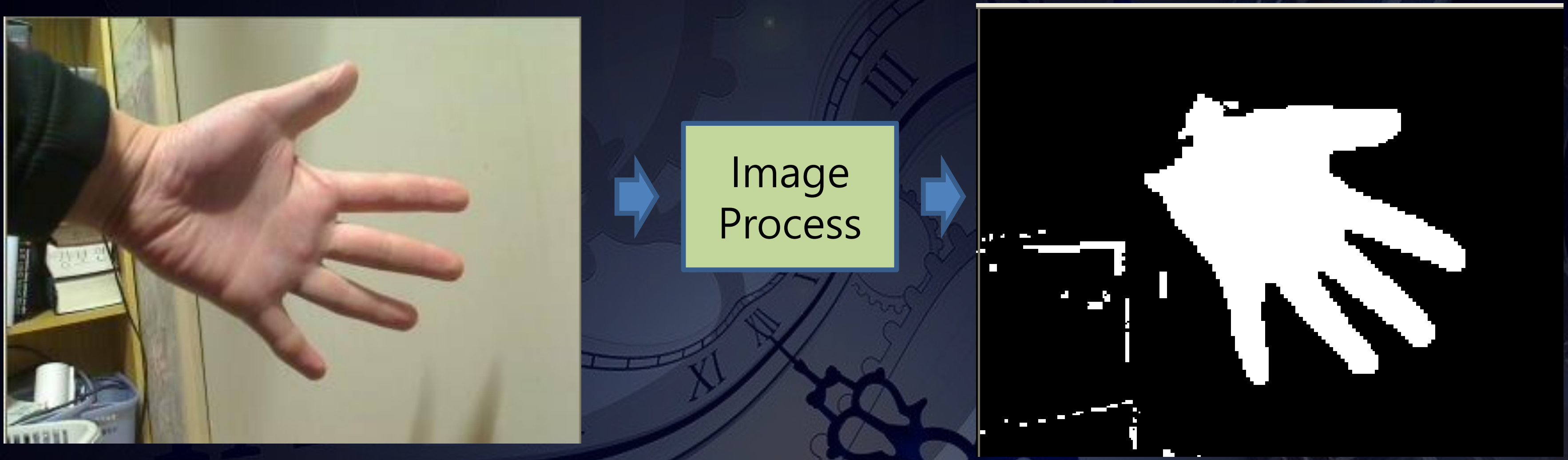


Image Process

무게중심 및 끝점 찾기

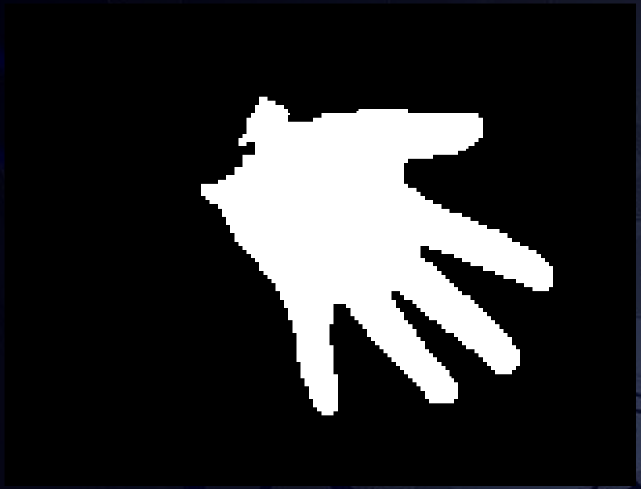
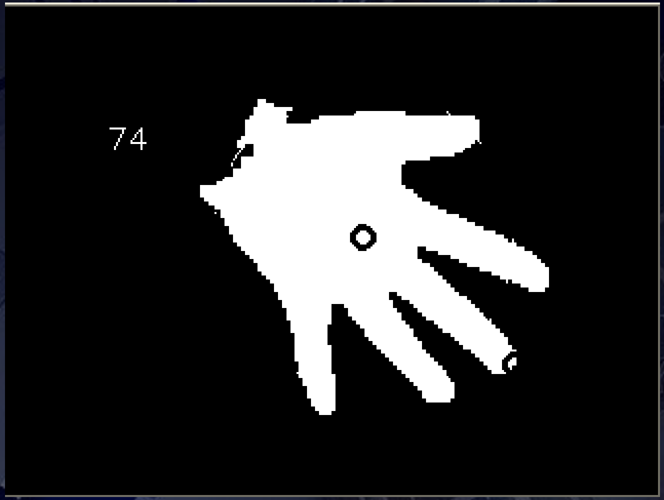


Image Process



손 영상처리 과정

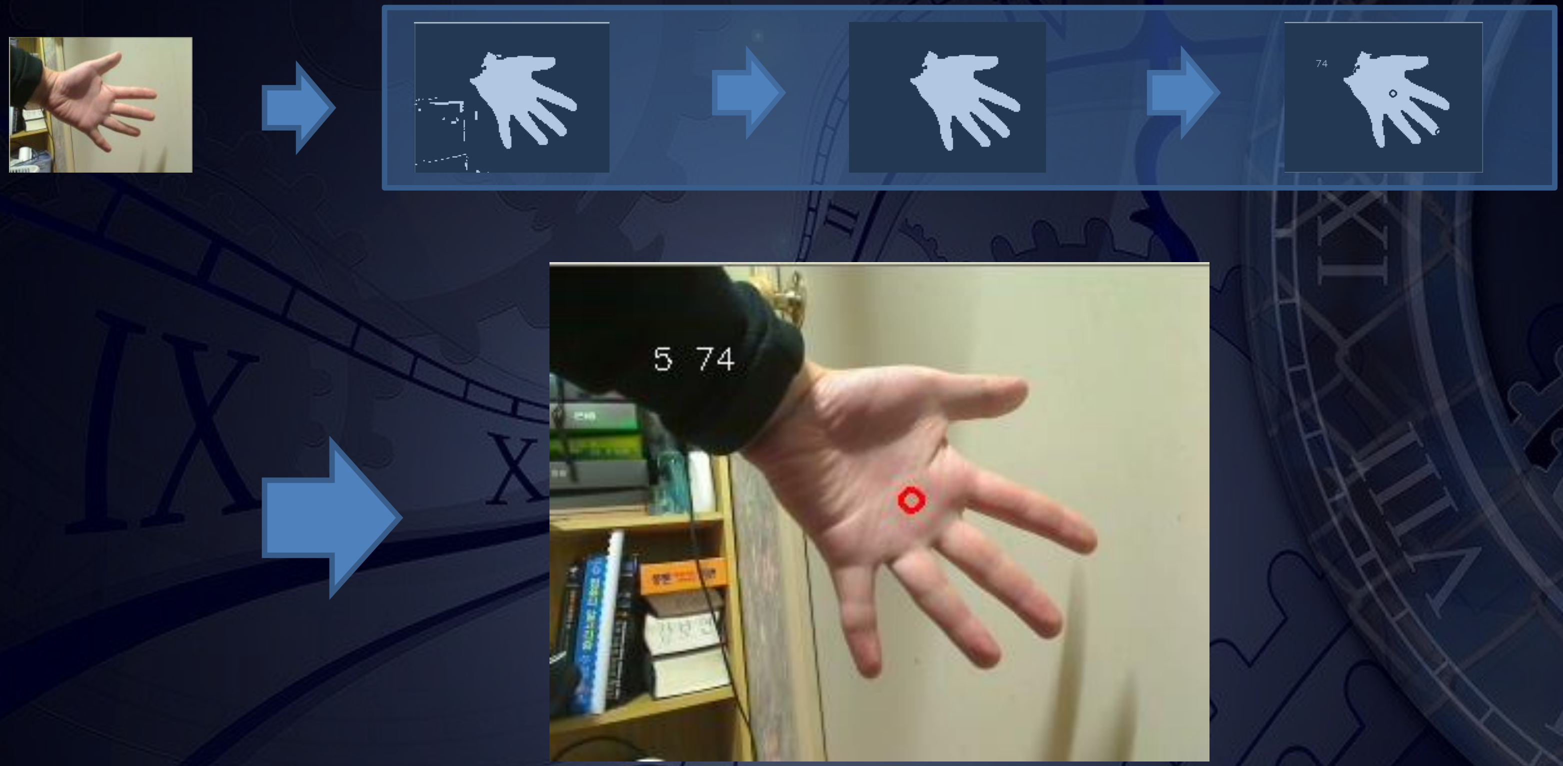


Image Processing

손 동작 인식

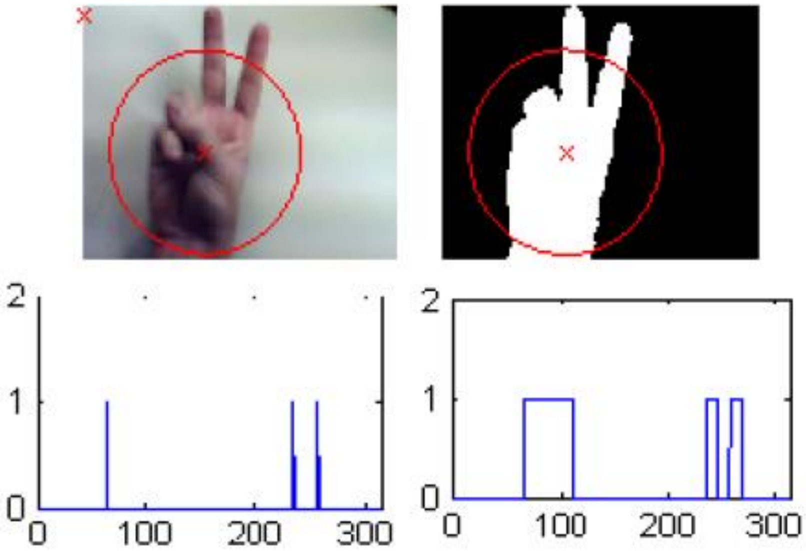
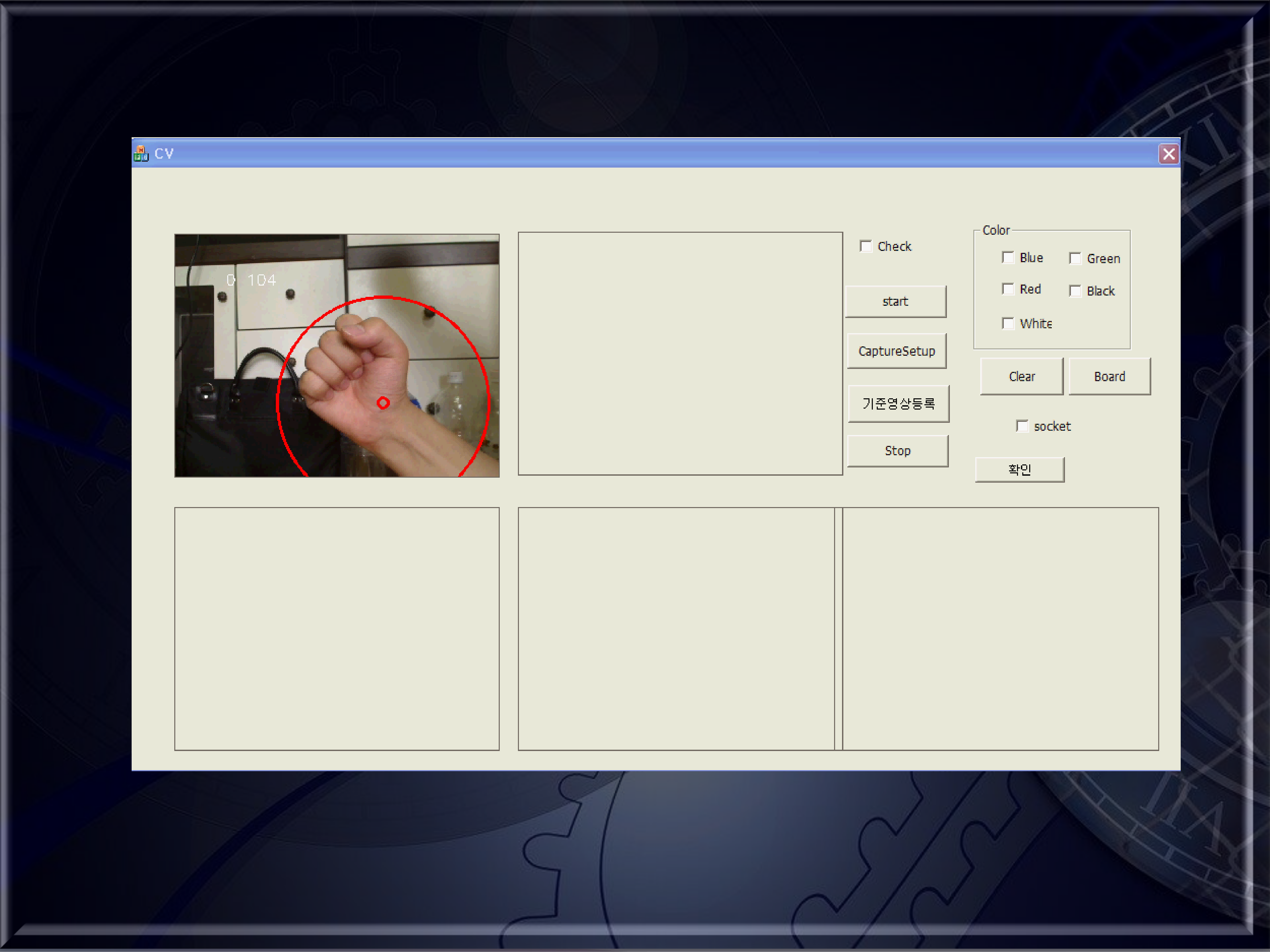


Figure 6: Sample result for a "two" count. Top left: Original image and the circle constructed. Top right: Segmented hand and the circle constructed. Bottom left: Zero-to-one transitions in the 1D signal extracted. Bottom right: The 1D signal itself.

A Malima, E Ozgur, M Cetin, "**A fast algorithm for Vision based hand gesture recognition for robot control,**" 14th IEEE conference on Signal and Image Processing

손 동작 인식

시연 (동영상)


CV
0 104
☐ Check
start
CaptureSetup
기준영상등록
Stop
Color
☐ Blue
☐ Green
☐ Red
☐ Black
☐ White
Clear
Board
☐ socket
확인

Part2 - 물체 인식

물체 인식

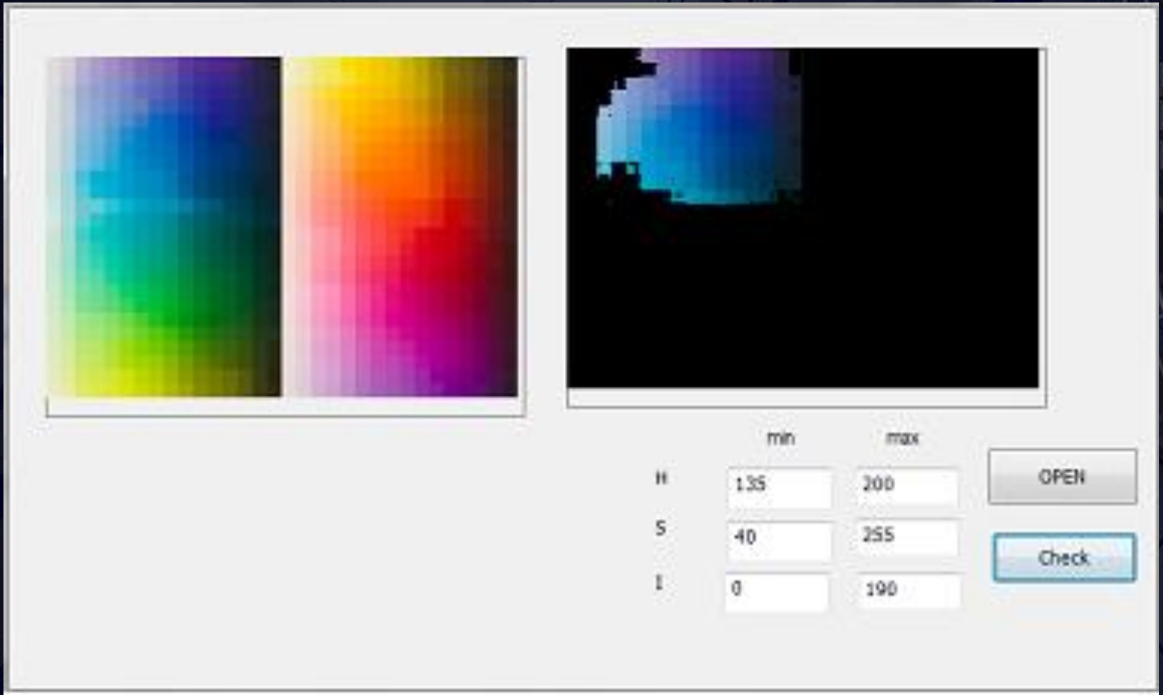
1. Image Capture with webcam
2. HSI, YUV color model을 이용한 색 인식
    - 영상 보정
    - 물체 크기 파악(레이블링)
    - 물체 위치 판별(물체 중심점)
3. SIFT algorithm 적용

색 인식

- HSI
  - 색상, 채도, 명도로 색을 표현하는 색 모델
    - H 색상 : 색의 종류
    - S 채도 : 색의 탁하고 선명한 정도
    - I 명도 : 빛의 밝기
- YUV
  - 컬러 비디오 표준에 사용되는 색 모델
    - Y 성분 : 빛의 밝기
    - U 성분 : 파란색에서 밝기 성분을 뺀 정보
    - V 성분 : 빨간색에서 밝기 성분을 뺀 정보

색 인식

- RGB색 모델 -> HSI, YUV 색 모델



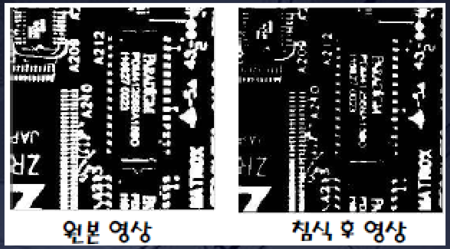
	min	max
H	135	200
S	40	255
I	0	190

색 인식

- Red
  - 0< H < 5, 235 < H < 255
  - 0 < S < 255
  - 0 < I < 210
  - Y = 0
  - 80 < U < 150
  - 160 < V < 255
- Green
  - 60< H < 135
  - 45 < S < 255
  - 10 < I < 210
  - Y = 0
  - 40 < U < 145
  - 0 < V < 125
- Blue
  - 135 < H < 200
  - 40 < S < 255
  - 0 < I < 190
  - Y = 0
  - 130 < U < 255
  - 0 < V < 125
- Yellow
  - 33 < H < 50
  - 20 < S < 255
  - 50 < I < 200
  - Y = 0
  - 0 < U < 95
  - 140 < V < 185

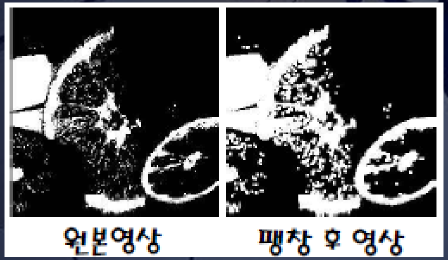
색 인식

- 침식 연산
  - 물체에 대해 배경을 확장시키고 물체의 크기를 축소



원본 영상 침식 후 영상

- 팽창 연산
  - 물체의 크기를 확장시키고 배경을 축소



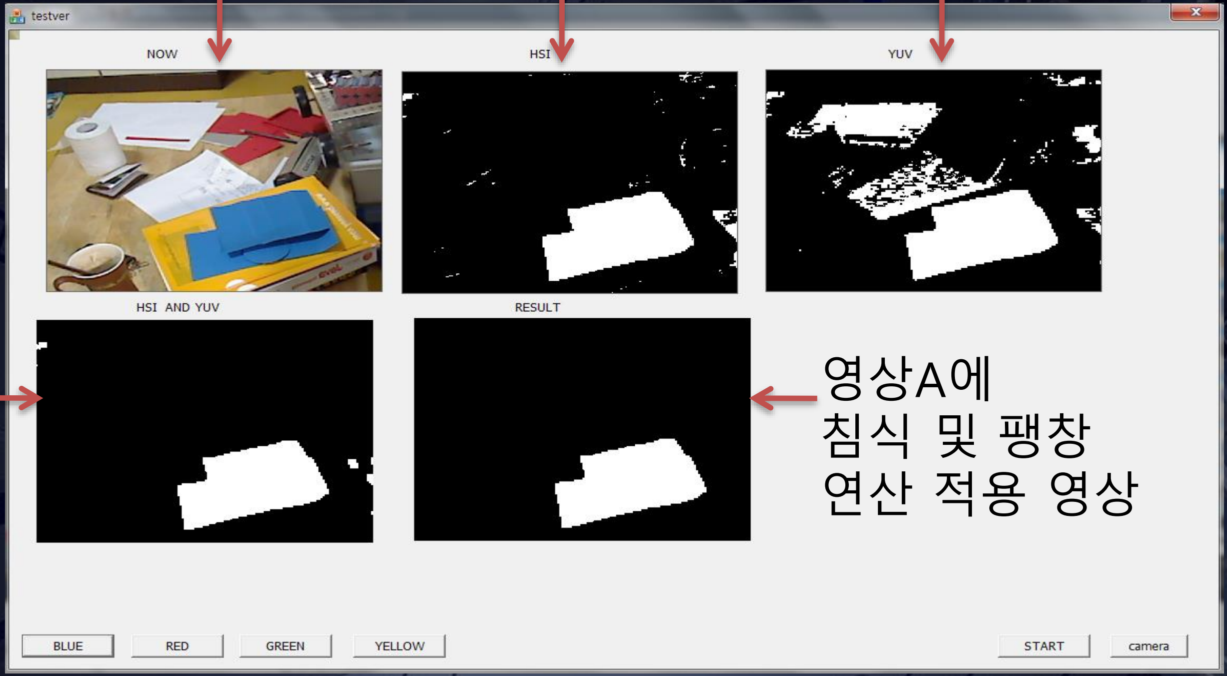
원본영상 팽창 후 영상

물체 인식

원래 영상

HSI색 모델 영상

YUV색 모델 영상

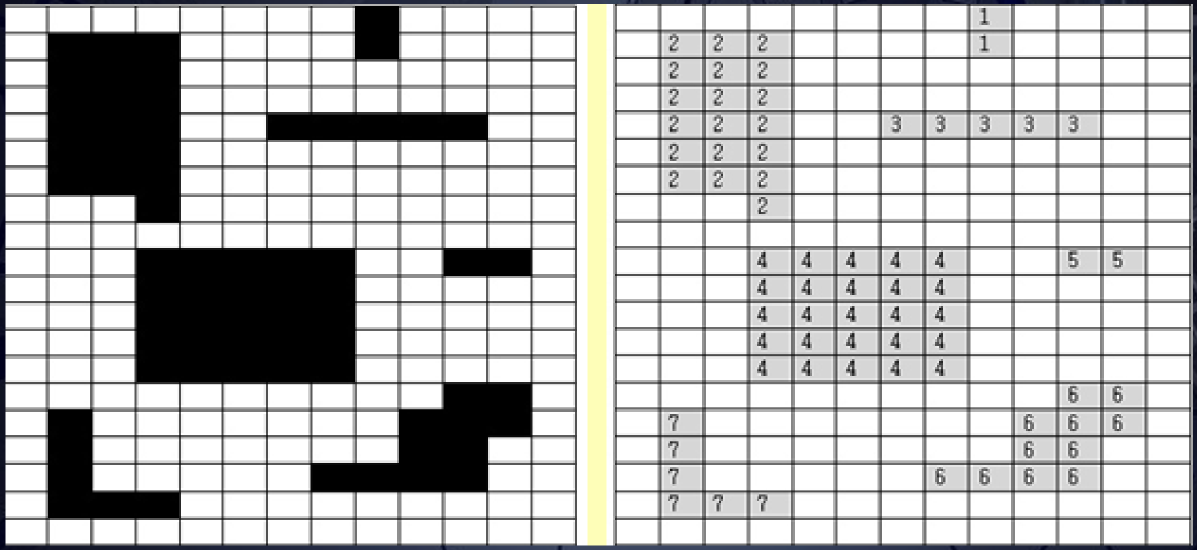


HSI와 YUV를
AND시킨 영상
영상(A)

영상A에
침식 및 팽창
연산 적용 영상

물체 인식 - 영상 보정

- 레이블링
  - 인접한 화소에 모두 같은 번호를 붙이고 연결되지 않은 다른 성분에는 다른 번호를 붙이는 작업



								1				
	2	2	2					1				
	2	2	2									
	2	2	2									
	2	2	2			3	3	3	3	3		
	2	2	2									
	2	2	2									
			2									
			4	4	4	4	4			5	5	
			4	4	4	4	4					
			4	4	4	4	4					
			4	4	4	4	4					
			4	4	4	4	4					
										6	6	
	7								6	6	6	
	7								6	6		
	7						6	6	6	6		
	7	7	7									

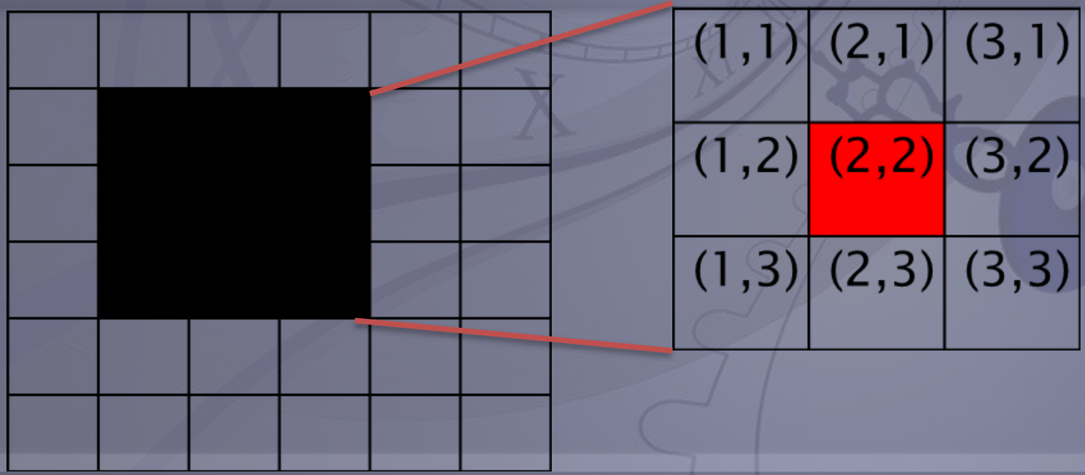
물체 인식 - 영상 보정

- 제일 큰 물체 찾기
  - 레이블링에서 주어진 label번호를 이용하여 물체의 사이즈를 잰다.
  - 그 사이즈가 제일 큰 물체를 선택하여 화면에 표시한다.
  - 제일 큰 물체의 사이즈가 너무 작을 경우 무시한다.

- 이점
  - 노이즈를 제거할 수 있다.
  - 한 화면에 여러 개 물체가 있어도 그 물체들 중에서 하나만 확실히 선택된다.

물체 인식

- 물체 중심 구하기
  - 물체의 위치를 파악하기 위해 물체의 중심을 구한다.
    - 물체가 있는 pixel를 찾을 때마다 count를 1증가
    - 그 pixel의 x좌표 y좌표 값을 각각 더하여 저장한다.
    - Count로 위에서 구한 총 x좌표 y좌표 값을 나눈다.



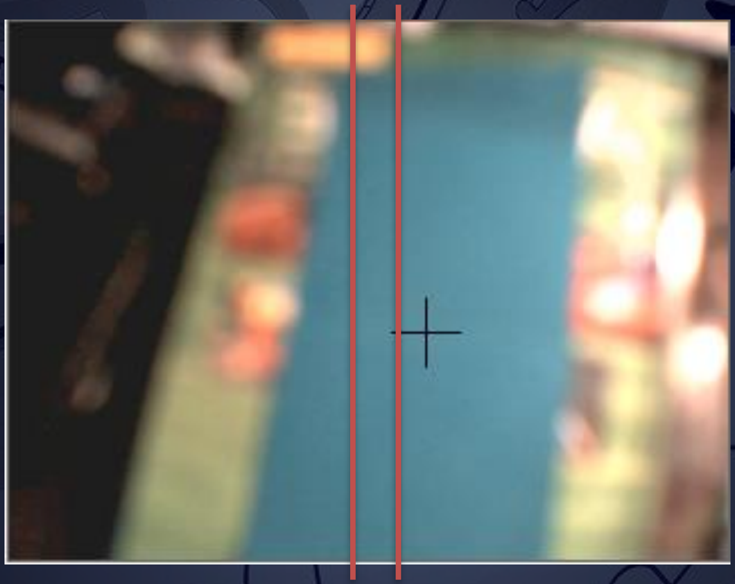
(1,1)	(2,1)	(3,1)
(1,2)	(2,2)	(3,2)
(1,3)	(2,3)	(3,3)

Count : 9
총 x좌표 값 : 18
총 y좌표 값 : 18

즉 중심점은
(18/9 , 18/9) = (2,2)

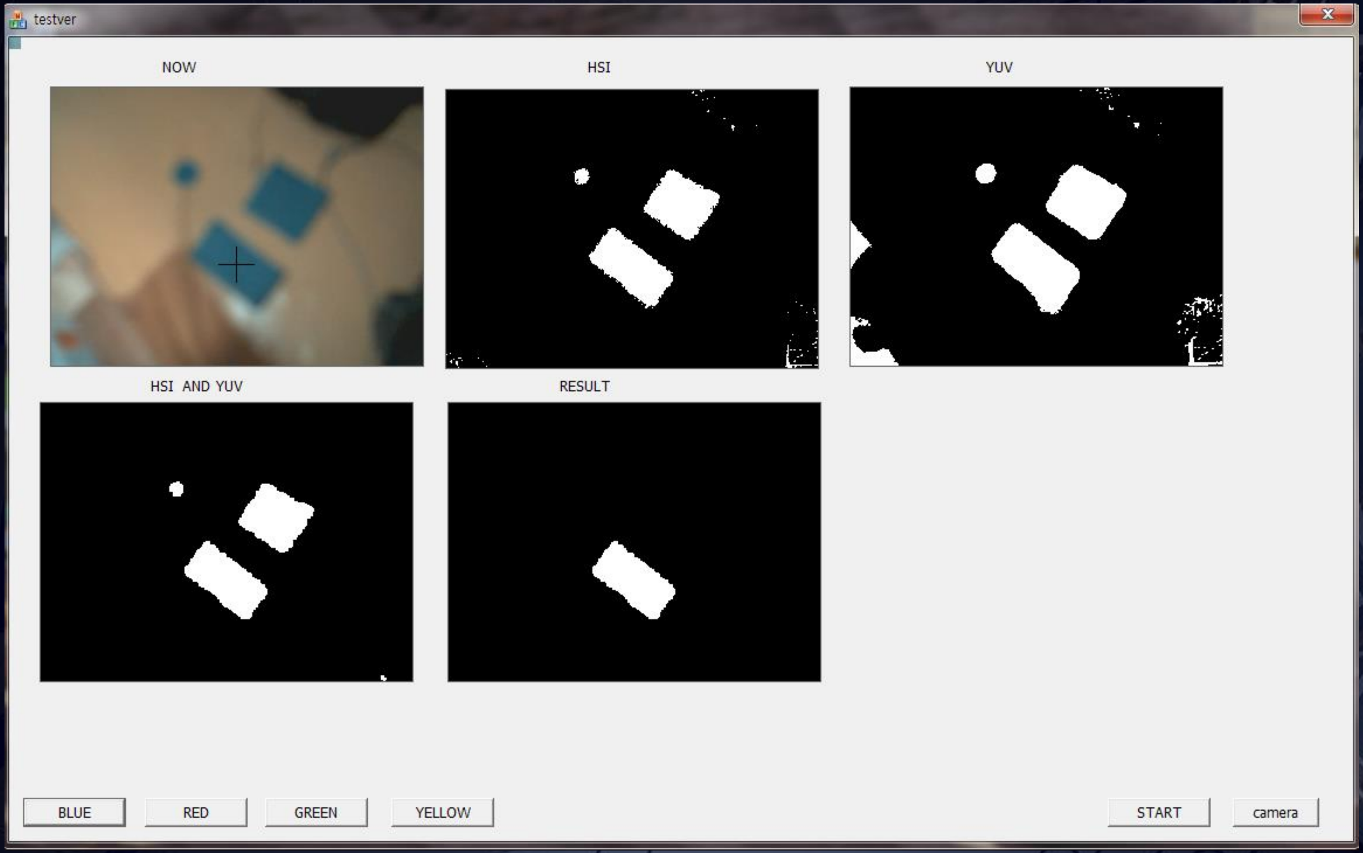
물체 인식

- 물체 위치 판단
  - 물체의 중심점의 x좌표를 이용하여 물체 위치를 판단
    - 화면의 중심의 x좌표 ± 10 (pixel)하여 물체의 중심점의 x좌표 값이 그 범위를 벗어났는지를 판단하여 물체가 왼쪽에 있는지 오른쪽에 있는지 판단한다.

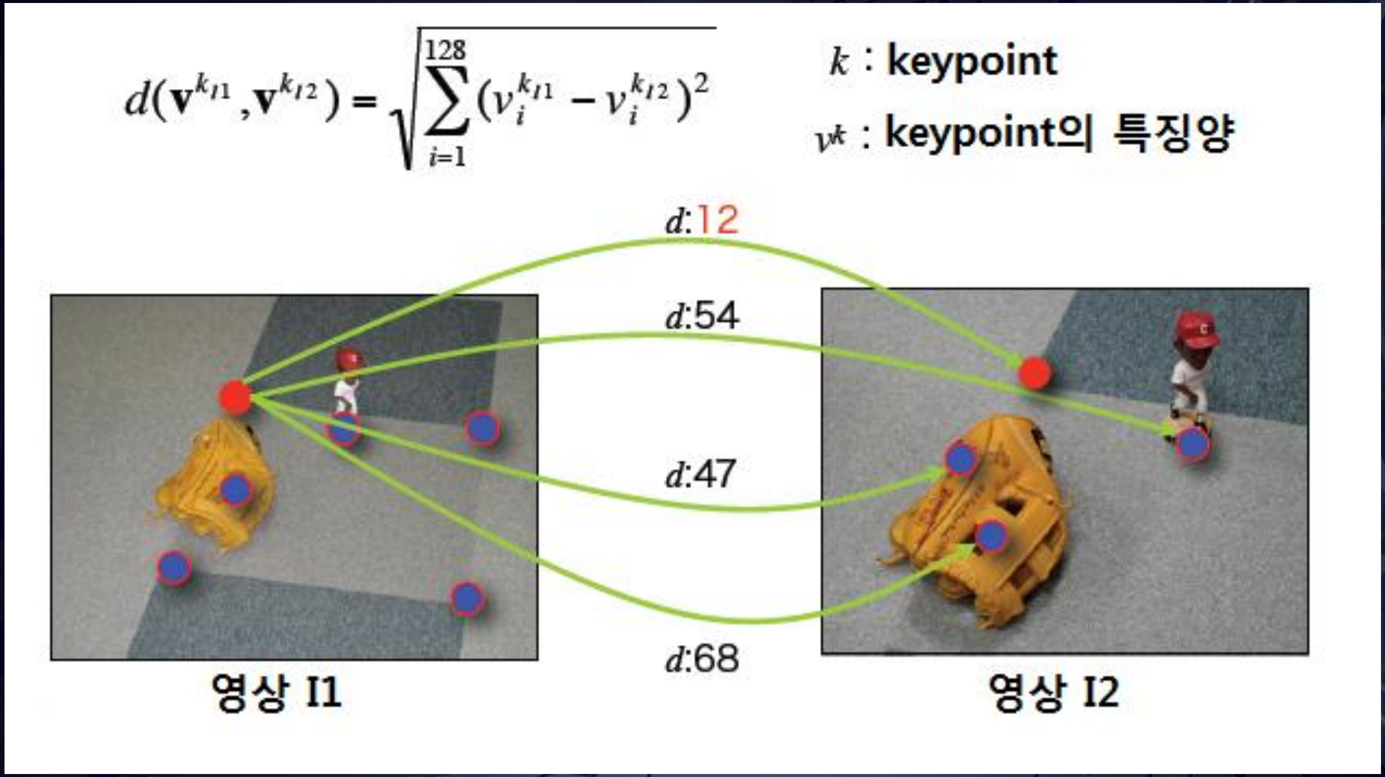


오른쪽에 있음

물체 인식


testver
NOW
HSI
YUV
HSI AND YUV
RESULT
BLUE
RED
GREEN
YELLOW
START
camera

물체 인식 - SIFT Algorithm



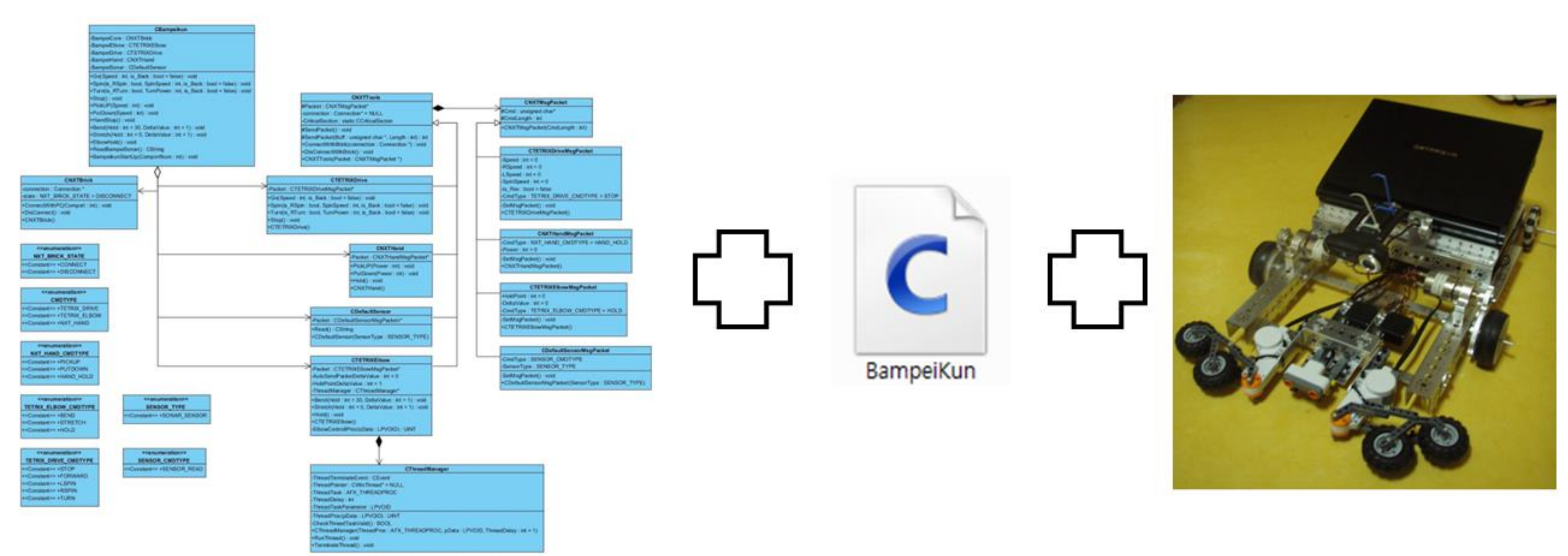
$$d(\mathbf{v}^{k_{I1}}, \mathbf{v}^{k_{I2}}) = \sqrt{\sum_{i=1}^{128} (v_i^{k_{I1}} - v_i^{k_{I2}})^2}$$

k : keypoint

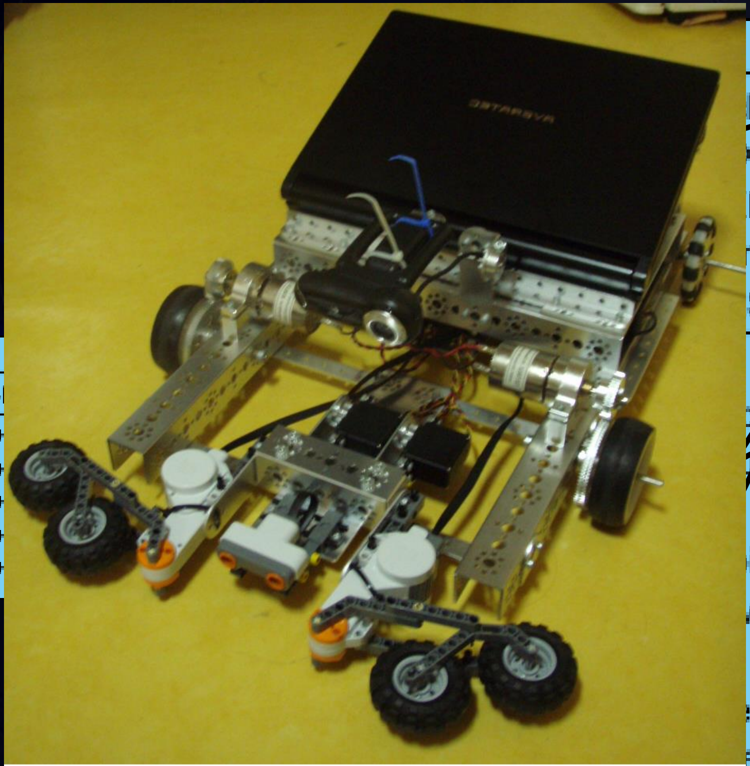
v^k : keypoint의 특징양

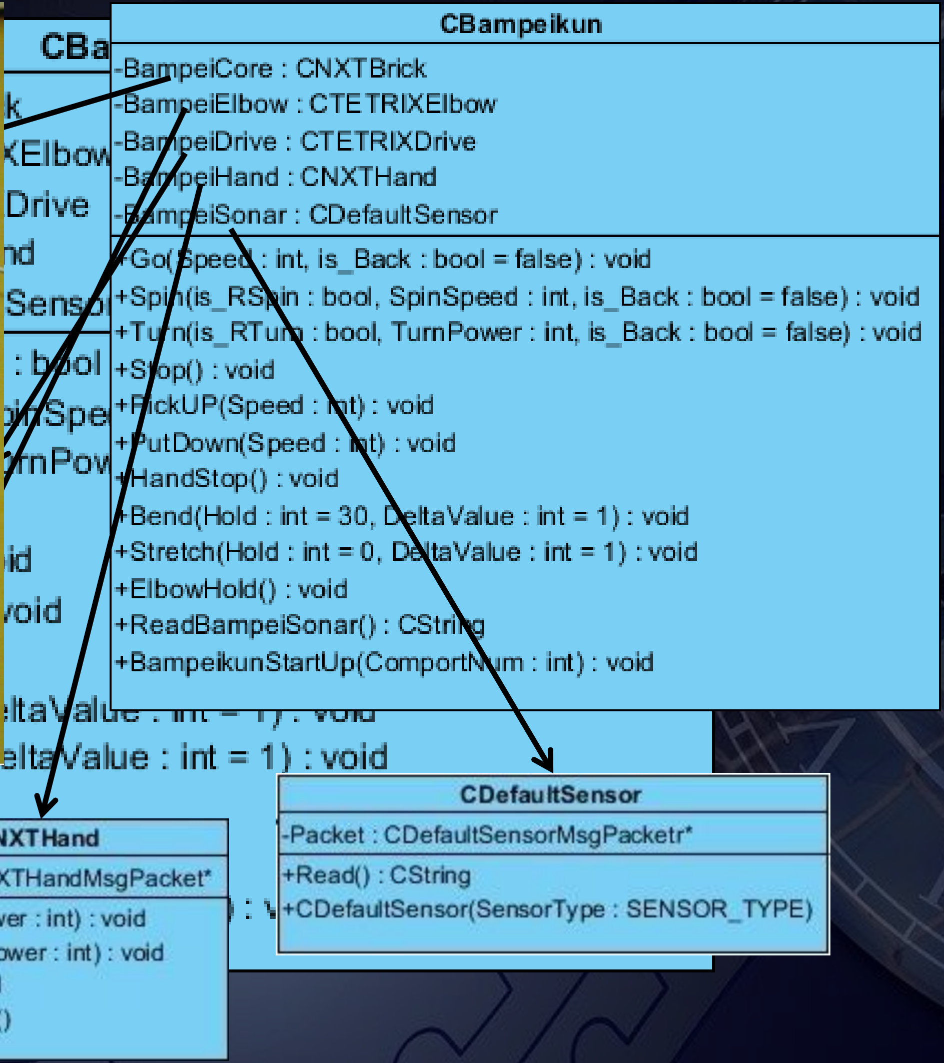
Part3 - 로봇

로봇 구성



로봇 구성





CBampeikun

-BampeiCore : CNXTBrick
-BampeiElbow : CTETRIXElbow
-BampeiDrive : CTETRIXDrive
-BampeiHand : CNXTHand
-BampeiSonar : CDefaultSensor

+Go(Speed : int, is_Back : bool = false) : void
+Spin(is_RSpin : bool, SpinSpeed : int, is_Back : bool = false) : void
+Turn(is_RTurn : bool, TurnPower : int, is_Back : bool = false) : void
+Stop() : void
+PickUP(Speed : int) : void
+PutDown(Speed : int) : void
+HandStop() : void
+Bend(Hold : int = 30, DeltaValue : int = 1) : void
+Stretch(Hold : int = 0, DeltaValue : int = 1) : void
+ElbowHold() : void
+ReadBampeiSonar() : CString
+BampeikunStartUp(ComportNum : int) : void

-HoldPointDeltaValue : int = 1
-ThreadManager : CThreadManager*

+Bend(Hold : int = 30, DeltaValue : int = 1) : void
+Stretch(Hold : int = 0, DeltaValue : int = 1) : void
+Hold() : void
+CTETRIXElbow()
-ElbowControllProc(pData : LPVOID) : UINT

CNXTHand

-Packet : CNXTHandMsgPacket*

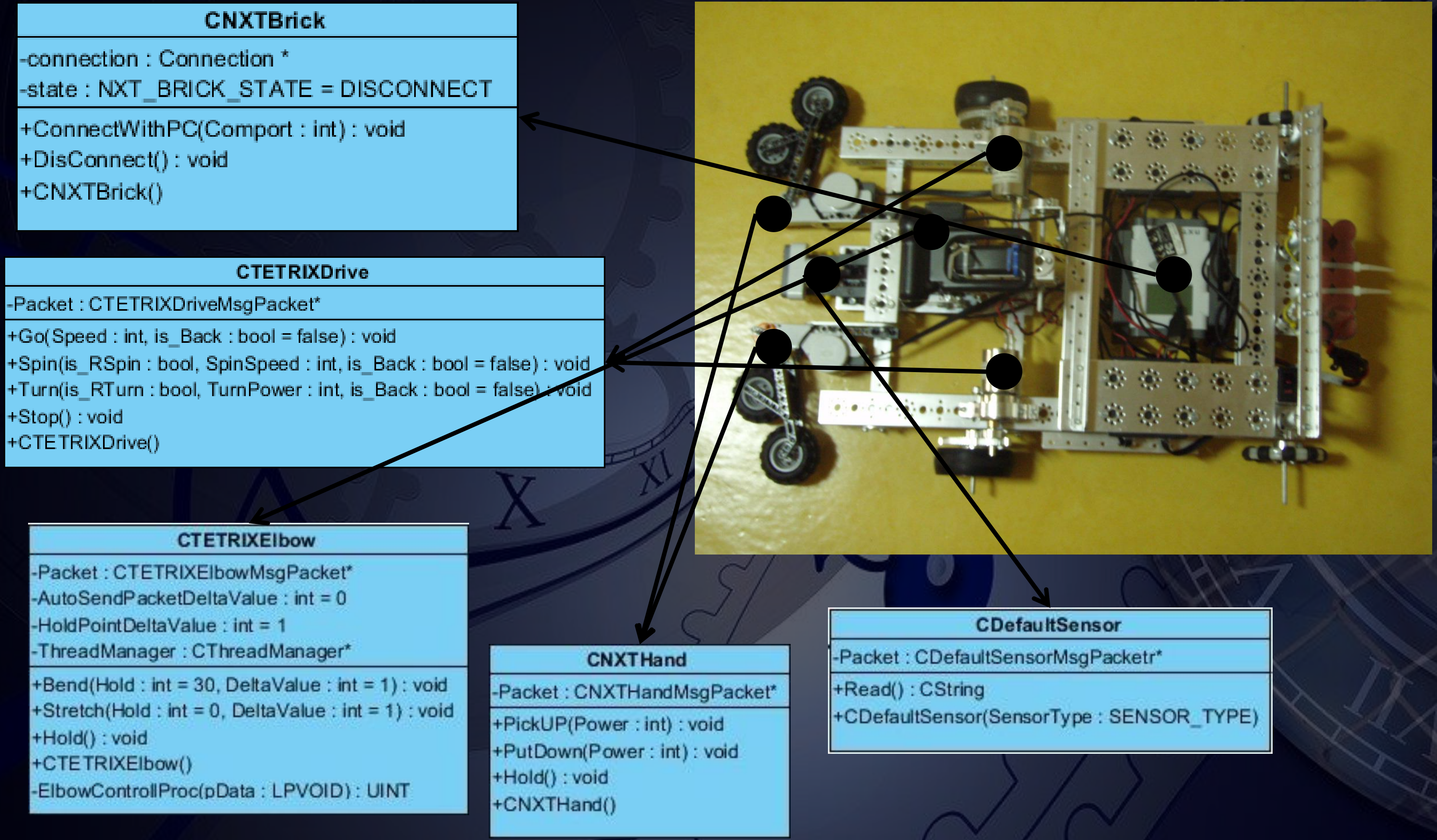
+PickUP(Power : int) : void
+PutDown(Power : int) : void
+Hold() : void
+CNXTHand()

CDefaultSensor

-Packet : CDefaultSensorMsgPacketr*

+Read() : CString
+CDefaultSensor(SensorType : SENSOR_TYPE)

로봇 구성



CNXTBrick

-connection : Connection *
-state : NXT_BRICK_STATE = DISCONNECT

+ConnectWithPC(Comport : int) : void
+DisConnect() : void
+CNXTBrick()

CTETRIXDrive

-Packet : CTETRIXDriveMsgPacket*

+Go(Speed : int, is_Back : bool = false) : void
+Spin(is_RSpin : bool, SpinSpeed : int, is_Back : bool = false) : void
+Turn(is_RTurn : bool, TurnPower : int, is_Back : bool = false) : void
+Stop() : void
+CTETRIXDrive()

CTETRIXElbow

-Packet : CTETRIXElbowMsgPacket*
-AutoSendPacketDeltaValue : int = 0
-HoldPointDeltaValue : int = 1
-ThreadManager : CThreadManager*

+Bend(Hold : int = 30, DeltaValue : int = 1) : void
+Stretch(Hold : int = 0, DeltaValue : int = 1) : void
+Hold() : void
+CTETRIXElbow()
-ElbowControllProc(pData : LPVOID) : UINT

CNXTHand

-Packet : CNXTHandMsgPacket*

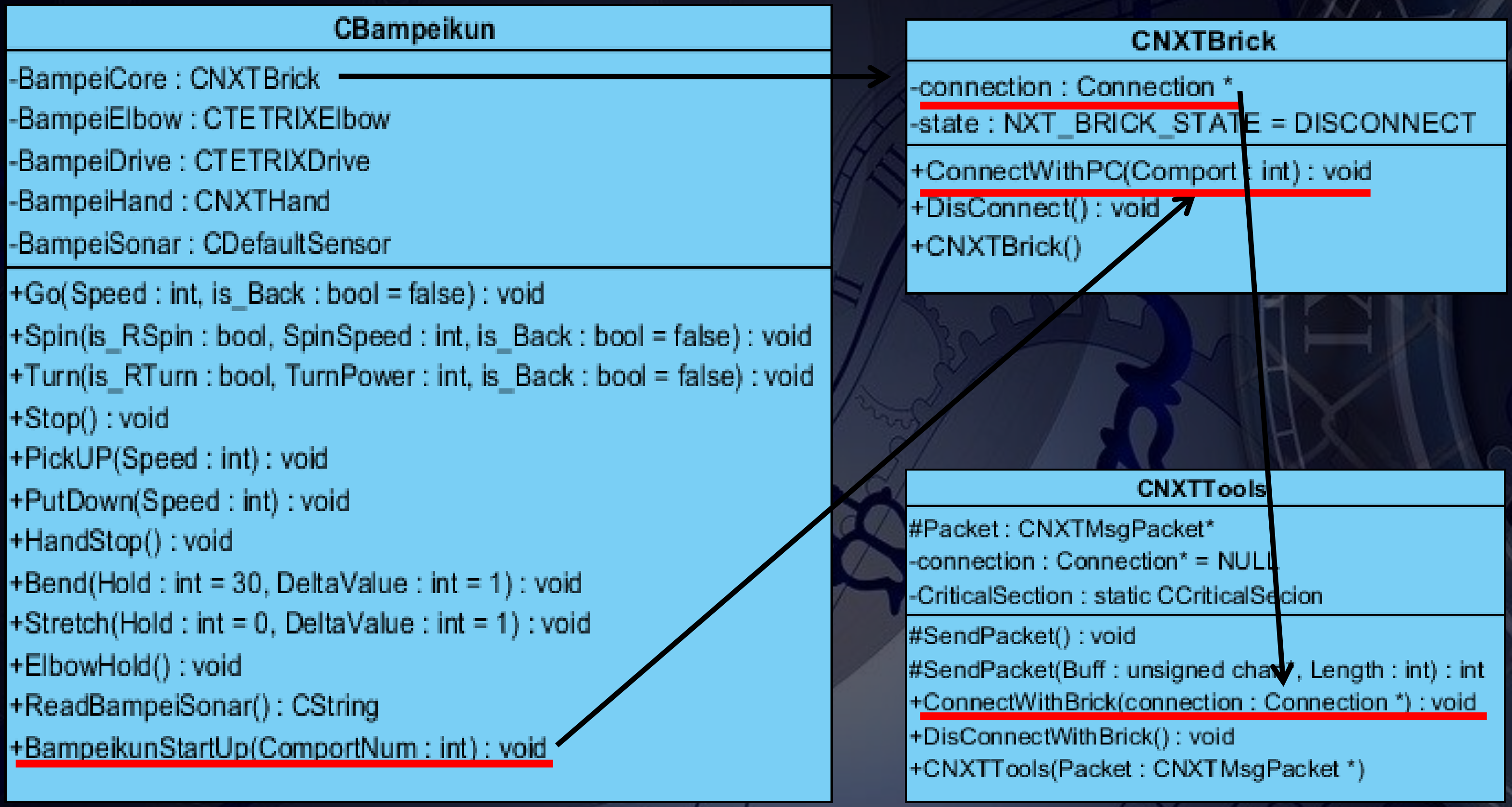
+PickUP(Power : int) : void
+PutDown(Power : int) : void
+Hold() : void
+CNXTHand()

CDefaultSensor

-Packet : CDefaultSensorMsgPacketr*

+Read() : CString
+CDefaultSensor(SensorType : SENSOR_TYPE)

로봇제어



CBampeikun

-BampeiCore : CNXTBrick
-BampeiElbow : CTETRIXElbow
-BampeiDrive : CTETRIXDrive
-BampeiHand : CNXTHand
-BampeiSonar : CDefaultSensor

+Go(Speed : int, is_Back : bool = false) : void
+Spin(is_RSpin : bool, SpinSpeed : int, is_Back : bool = false) : void
+Turn(is_RTurn : bool, TurnPower : int, is_Back : bool = false) : void
+Stop() : void
+PickUP(Speed : int) : void
+PutDown(Speed : int) : void
+HandStop() : void
+Bend(Hold : int = 30, DeltaValue : int = 1) : void
+Stretch(Hold : int = 0, DeltaValue : int = 1) : void
+ElbowHold() : void
+ReadBampeiSonar() : CString
+BampeikunStartUp(ComportNum : int) : void

CNXTBrick

-connection : Connection *
-state : NXT_BRICK_STATE = DISCONNECT

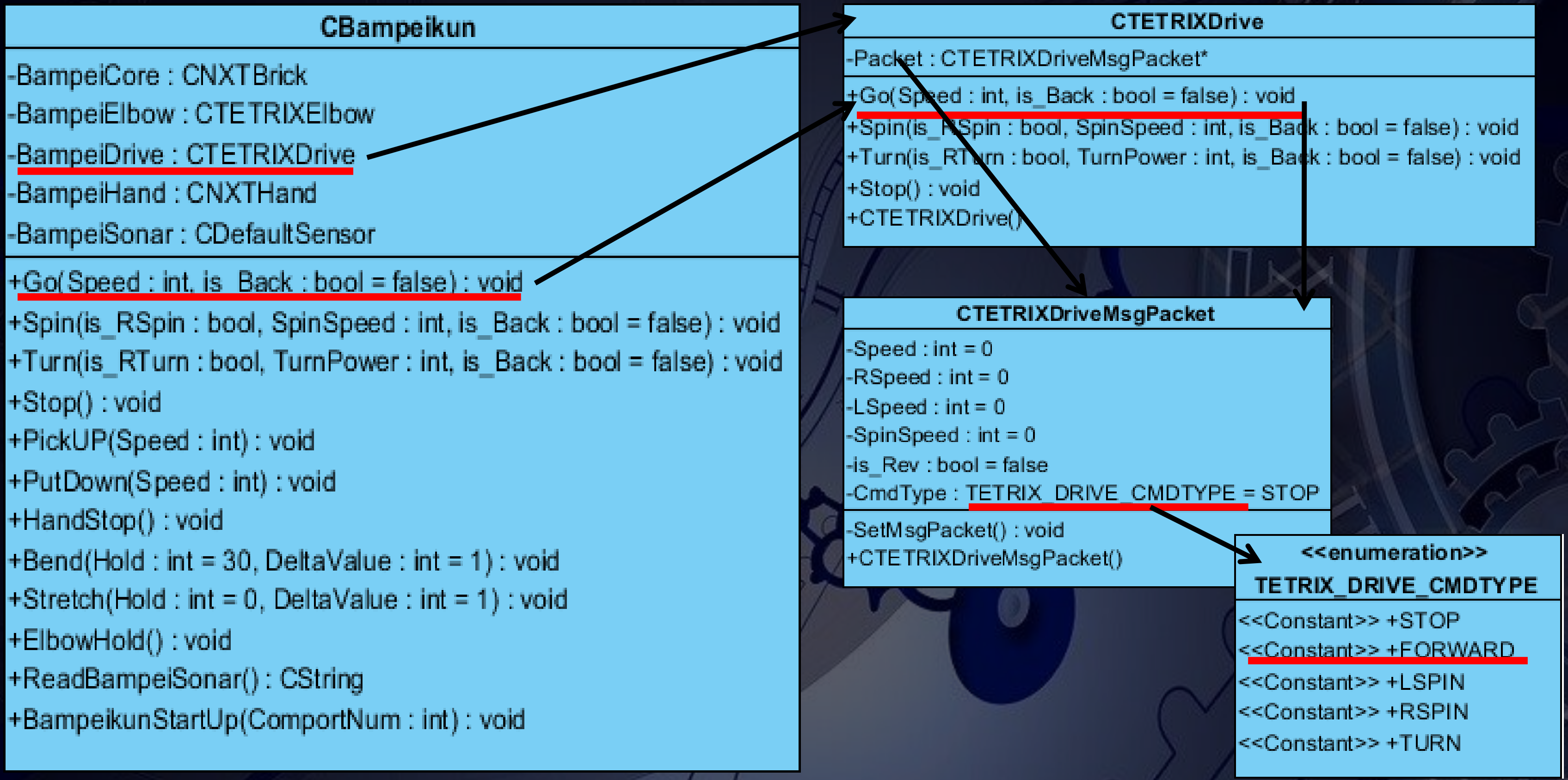
+ConnectWithPC(Comport : int) : void
+DisConnect() : void
+CNXTBrick()

CNXTTools

#Packet : CNXTMsgPacket*
-connection : Connection* = NULL
-CriticalSection : static CCriticalSection

#SendPacket() : void
#SendPacket(Buff : unsigned char*, Length : int) : int
+ConnectWithBrick(connection : Connection *) : void
+DisConnectWithBrick() : void
+CNXTTools(Packet : CNXTMsgPacket *)

로봇제어



CBampeikun

-BampeiCore : CNXTBrick
-BampeiElbow : CTETRIXElbow
-BampeiDrive : CTETRIXDrive
-BampeiHand : CNXTHand
-BampeiSonar : CDefaultSensor

+Go(Speed : int, is_Back : bool = false) : void
+Spin(is_RSpin : bool, SpinSpeed : int, is_Back : bool = false) : void
+Turn(is_RTurn : bool, TurnPower : int, is_Back : bool = false) : void
+Stop() : void
+PickUP(Speed : int) : void
+PutDown(Speed : int) : void
+HandStop() : void
+Bend(Hold : int = 30, DeltaValue : int = 1) : void
+Stretch(Hold : int = 0, DeltaValue : int = 1) : void
+ElbowHold() : void
+ReadBampeiSonar() : CString
+BampeikunStartUp(ComportNum : int) : void

CTETRIXDrive

-Packet : CTETRIXDriveMsgPacket*

+Go(Speed : int, is_Back : bool = false) : void
+Spin(is_RSpin : bool, SpinSpeed : int, is_Back : bool = false) : void
+Turn(is_RTurn : bool, TurnPower : int, is_Back : bool = false) : void
+Stop() : void
+CTETRIXDrive()

CTETRIXDriveMsgPacket

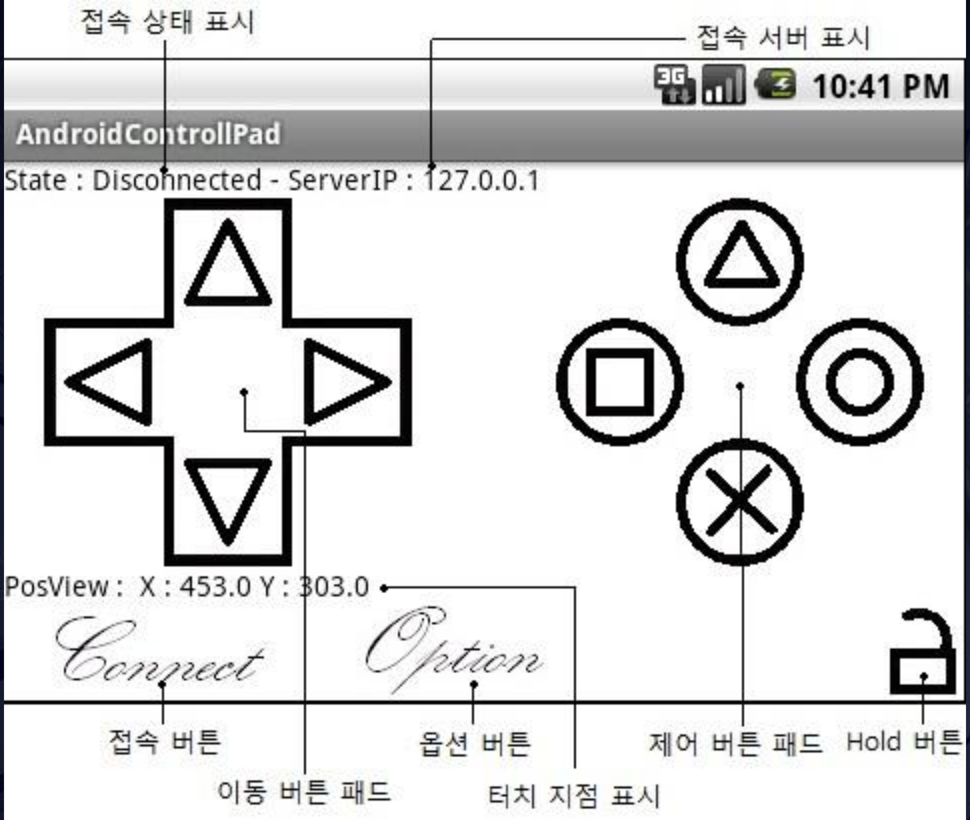
-Speed : int = 0
-RSpeed : int = 0
-LSpeed : int = 0
-SpinSpeed : int = 0
-is_Rev : bool = false
-CmdType : TETRIX_DRIVE_CMDTYPE = STOP

-SetMsgPacket() : void
+CTETRIXDriveMsgPacket()

<<enumeration>>
TETRIX_DRIVE_CMDTYPE

<<Constant>> +STOP
<<Constant>> +FORWARD
<<Constant>> +LSPIN
<<Constant>> +RSPIN
<<Constant>> +TURN

로봇제어

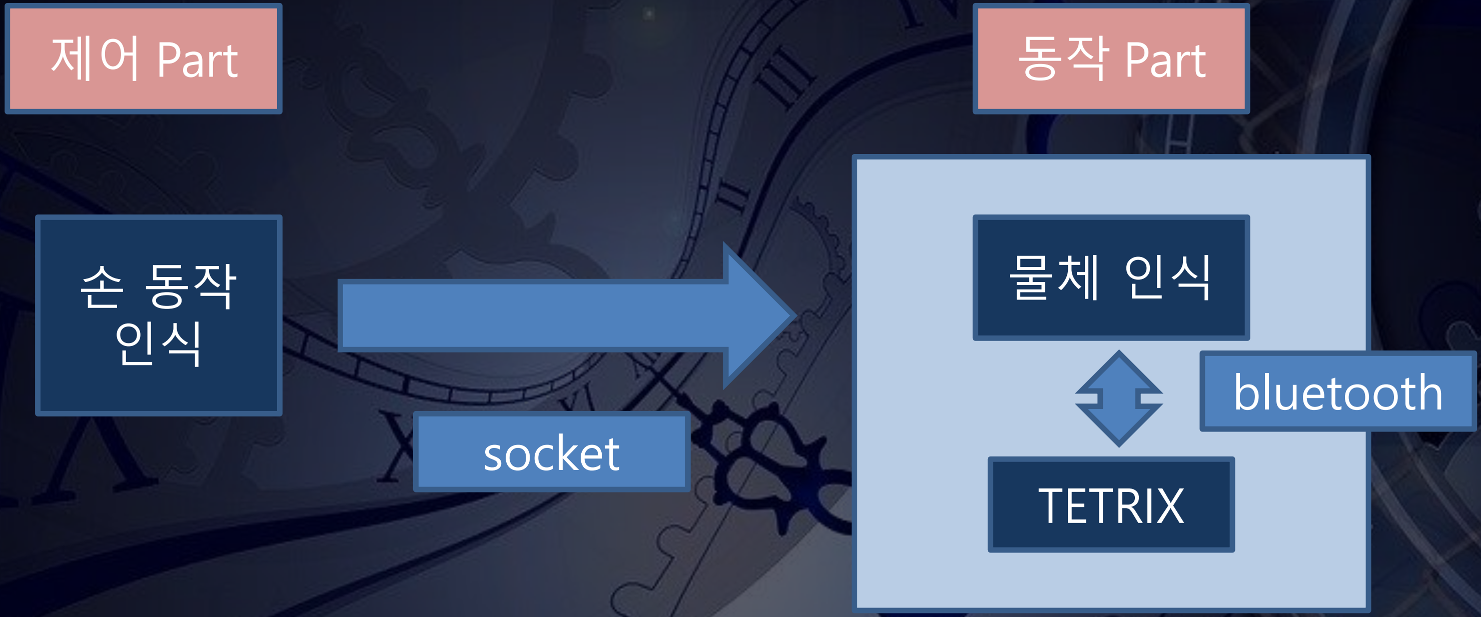


State : Disconnected - ServerIP : 127.0.0.1

PosView : X : 453.0 Y : 303.0

전체 시스템

환자보조로봇



제어 Part

손 동작 인식

socket

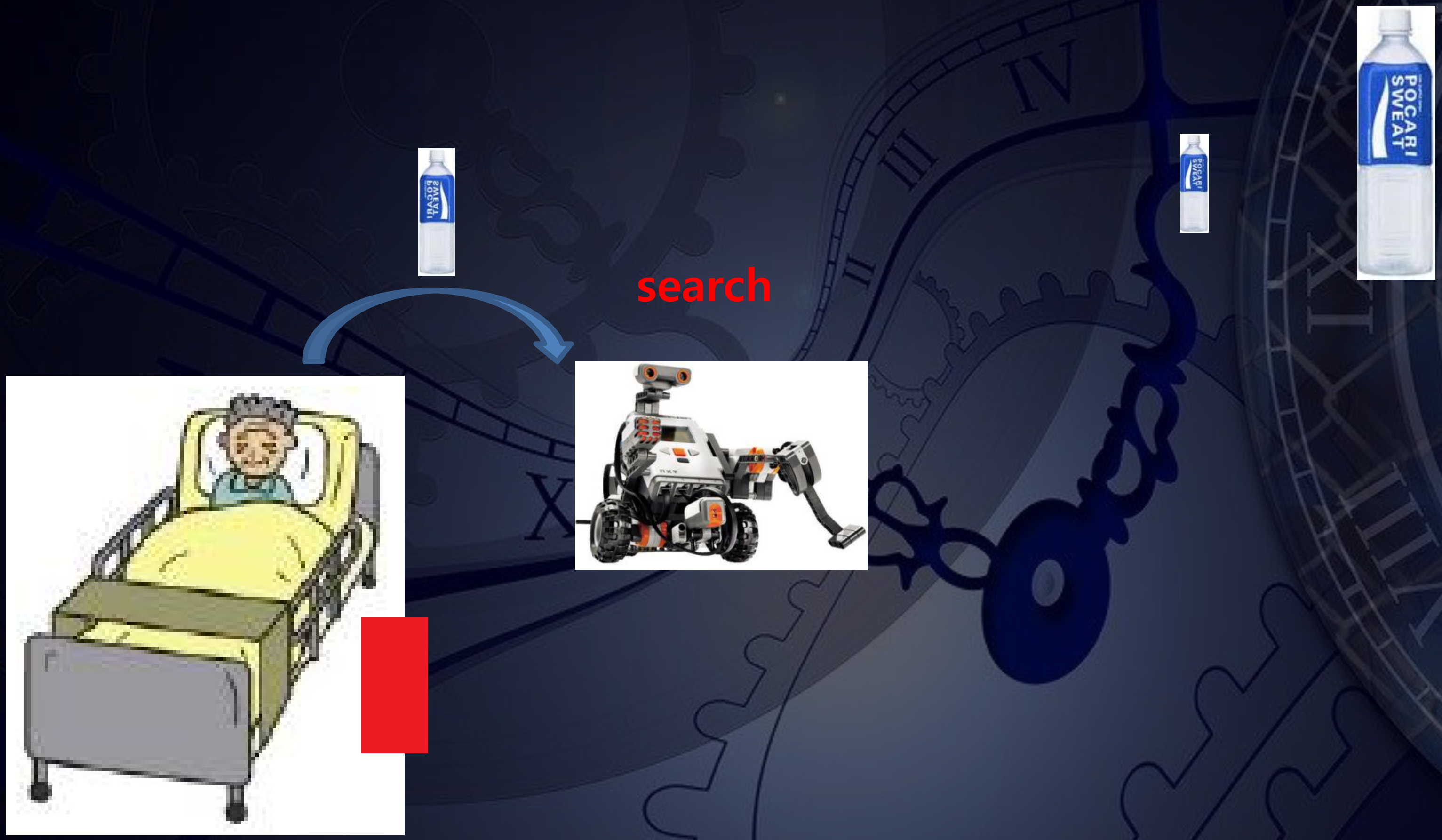
동작 Part

물체 인식

bluetooth

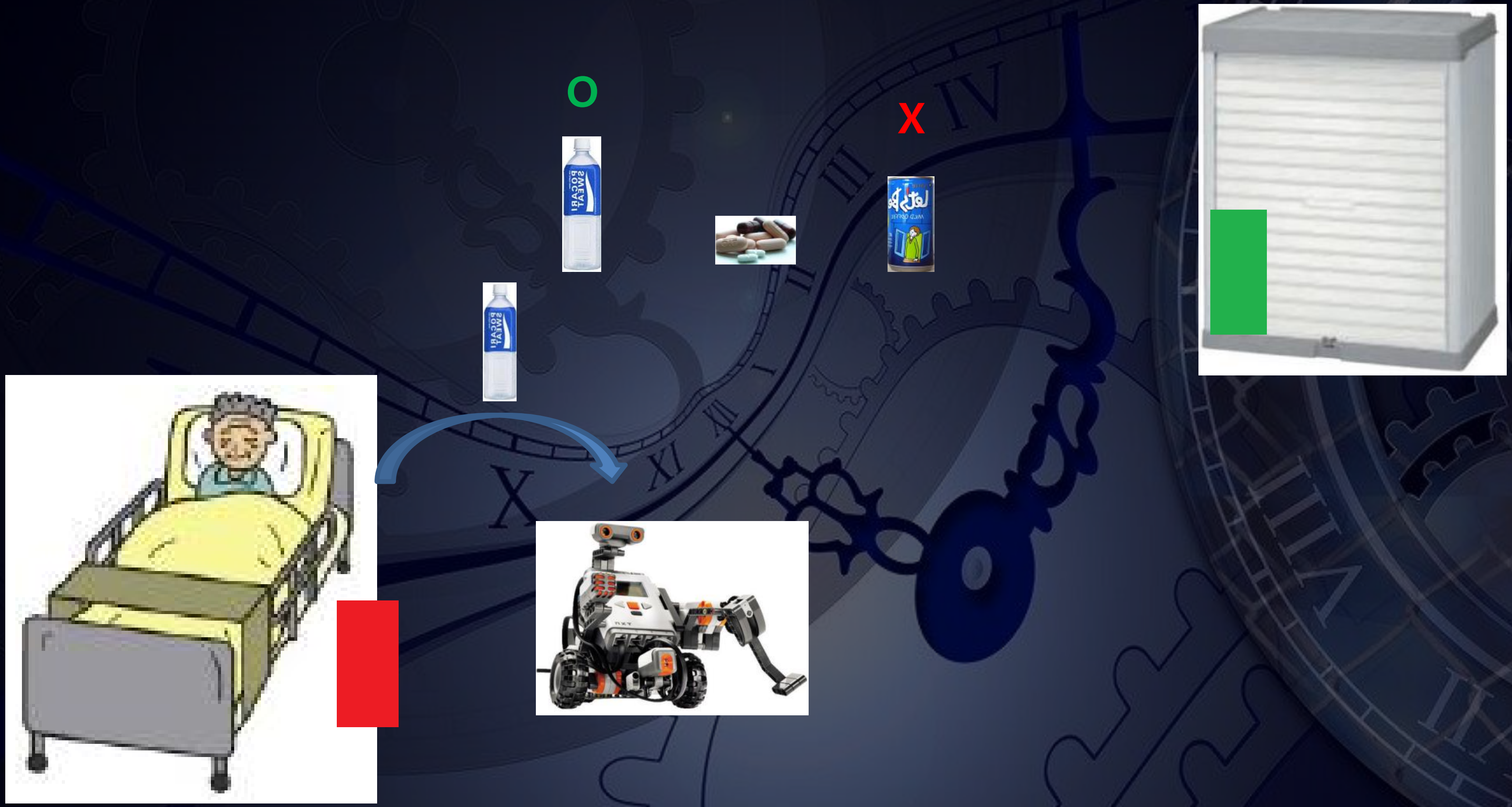
TETRIX

기본 시나리오



search

시나리오

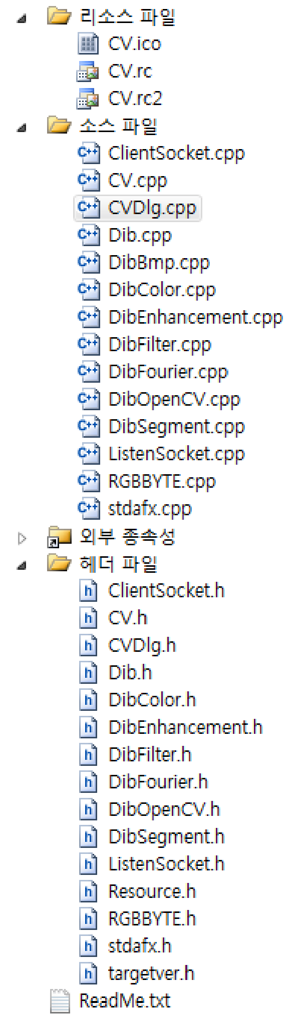


시연 (동영상)

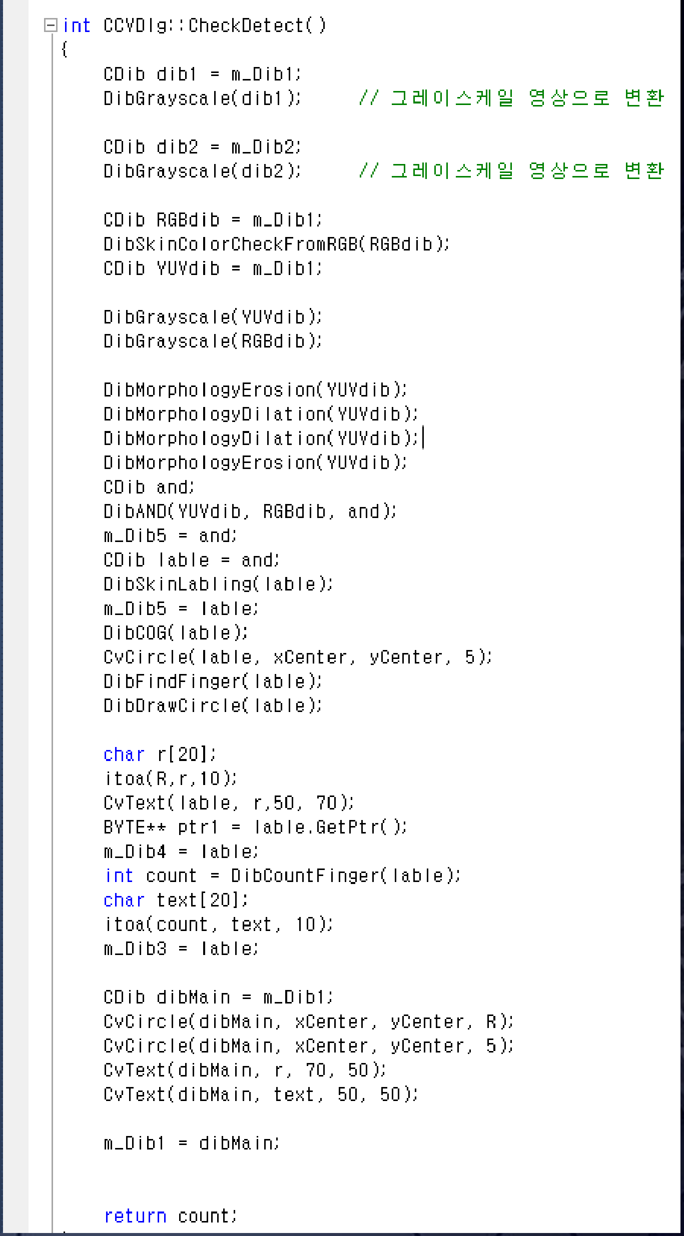
영상처리를 이용한 환자 보조 로봇

ISIS조

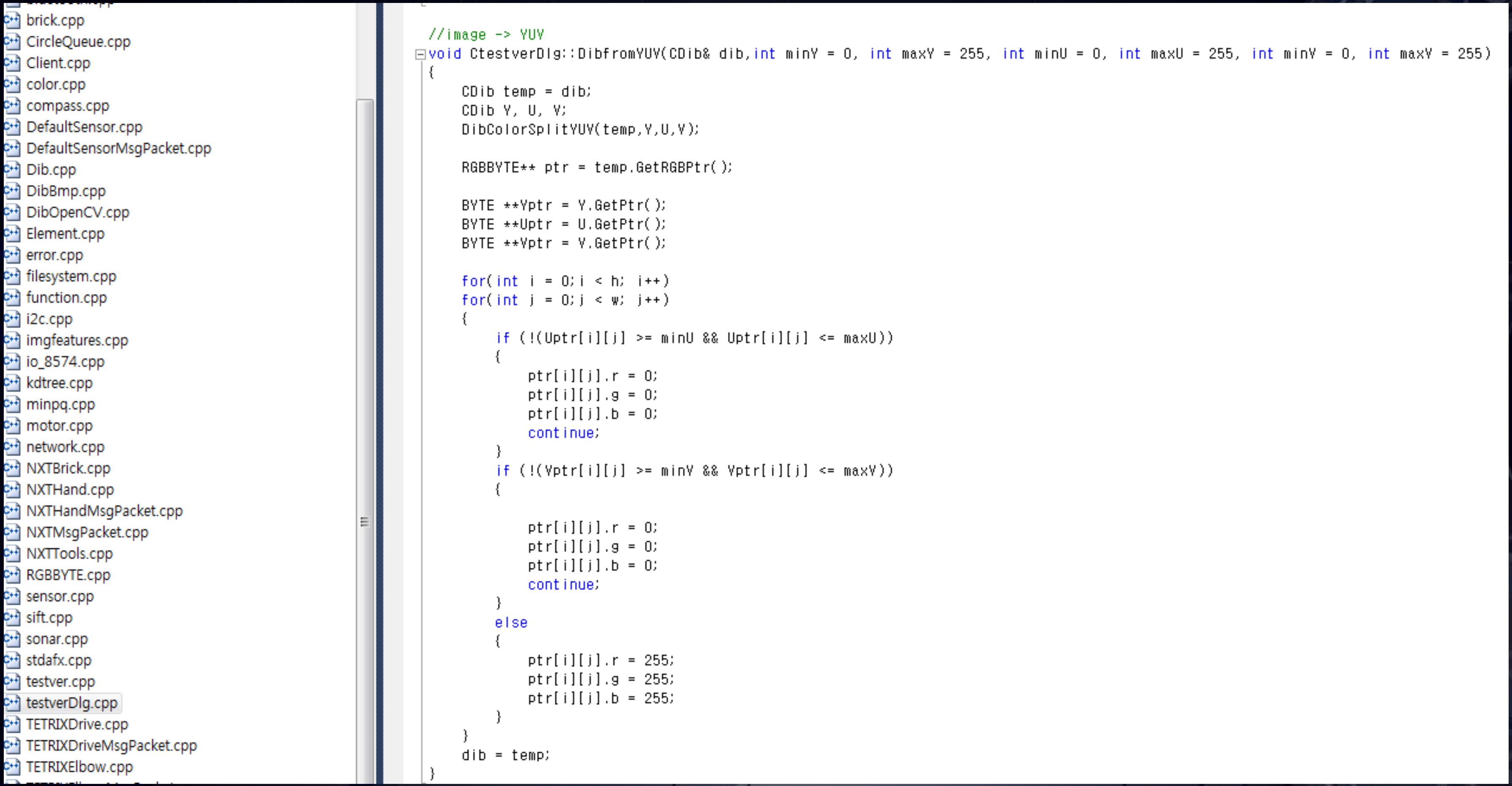
코드



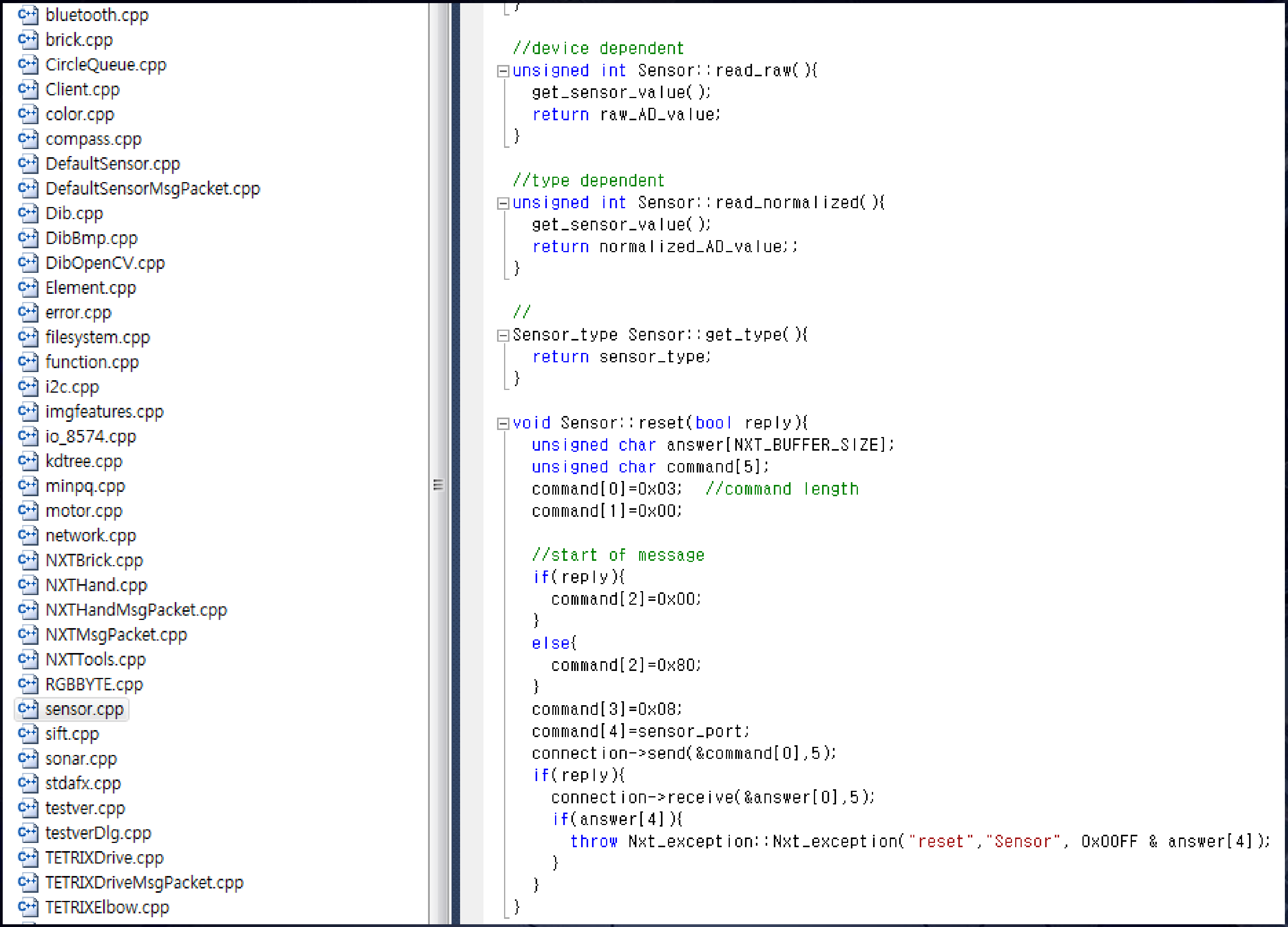
- 리소스 파일
  - CV.ico
  - CV.rc
  - CV.rc2
- 소스 파일
  - ClientSocket.cpp
  - CV.cpp
  - CVDlg.cpp
  - Dib.cpp
  - DibBmp.cpp
  - DibColor.cpp
  - DibEnhancement.cpp
  - DibFilter.cpp
  - DibFourier.cpp
  - DibOpenCV.cpp
  - DibSegment.cpp
  - ListenSocket.cpp
  - RGBBYTE.cpp
  - stdafx.cpp
- 외부 종속성
- 헤더 파일
  - ClientSocket.h
  - CV.h
  - CVDlg.h
  - Dib.h
  - DibColor.h
  - DibEnhancement.h
  - DibFilter.h
  - DibFourier.h
  - DibOpenCV.h
  - DibSegment.h
  - ListenSocket.h
  - Resource.h
  - RGBBYTE.h
  - stdafx.h
  - targetver.h
- ReadMe.txt



```
int CCVDlg::CheckDetect()
{
    CDib dib1 = m_Dib1;
    DibGrayscale(dib1);     // 그레이스케일 영상으로 변환

    CDib dib2 = m_Dib2;
    DibGrayscale(dib2);     // 그레이스케일 영상으로 변환

    CDib RGBdib = m_Dib1;
    DibSkinColorCheckFromRGB(RGBdib);
    CDib YUVdib = m_Dib1;

    DibGrayscale(YUVdib);
    DibGrayscale(RGBdib);

    DibMorphologyErosion(YUVdib);
    DibMorphologyDilation(YUVdib);
    DibMorphologyDilation(YUVdib);
    DibMorphologyErosion(YUVdib);
    CDib and;
    DibAND(YUVdib, RGBdib, and);
    m_Dib5 = and;
    CDib lable = and;
    DibSkinLabling(lable);
    m_Dib5 = lable;
    DibCOG(lable);
    CvCircle(lable, xCenter, yCenter, 5);
    DibFindFinger(lable);
    DibDrawCircle(lable);

    char r[20];
    itoa(R,r,10);
    CvText(lable, r,50, 70);
    BYTE** ptr1 = lable.GetPtr();
    m_Dib4 = lable;
    int count = DibCountFinger(lable);
    char text[20];
    itoa(count, text, 10);
    m_Dib3 = lable;

    CDib dibMain = m_Dib1;
    CvCircle(dibMain, xCenter, yCenter, R);
    CvCircle(dibMain, xCenter, yCenter, 5);
    CvText(dibMain, r, 70, 50);
    CvText(dibMain, text, 50, 50);

    m_Dib1 = dibMain;


    return count;
```

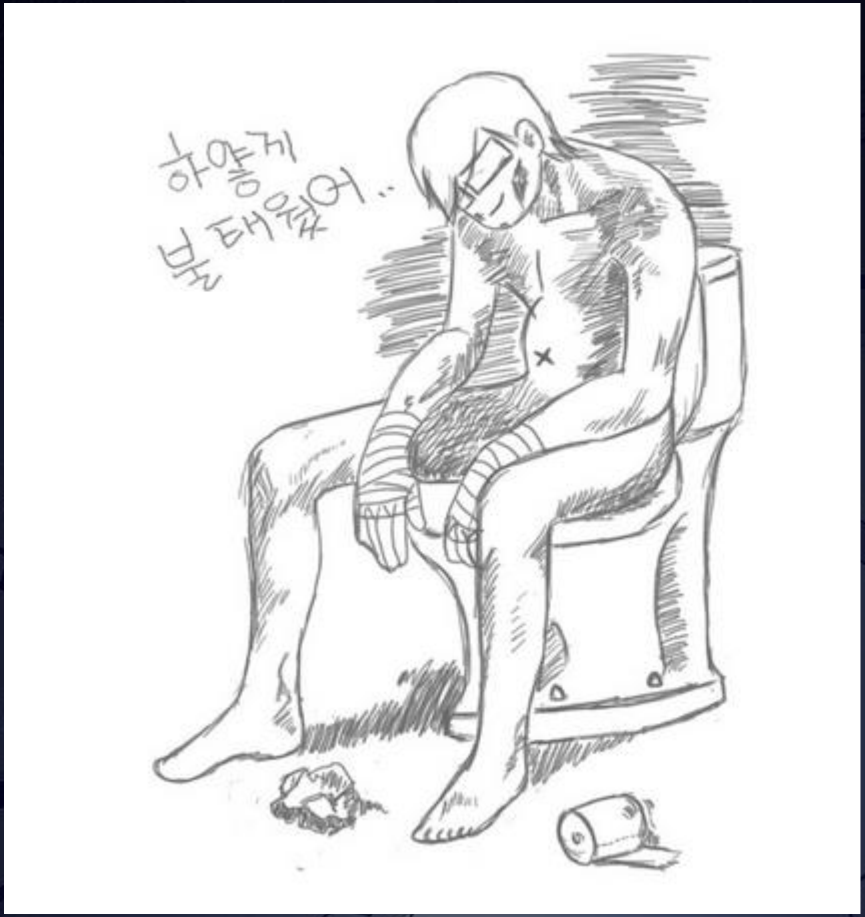


```
//image -> YUV
void CtestverDlg::DibfromYUV(CDib& dib,int minY = 0, int maxY = 255, int minU = 0, int maxU = 255, int minV = 0, int maxV = 255)
{
    CDib temp = dib;
    CDib Y, U, V;
    DibColorSplitYUV(temp,Y,U,V);

    RGBBYTE** ptr = temp.GetRGBPtr();

    BYTE **Yptr = Y.GetPtr();
    BYTE **Uptr = U.GetPtr();
    BYTE **Vptr = V.GetPtr();

    for(int i = 0;i < h; i++)
    for(int j = 0;j < w; j++)
    {
        if (!(Uptr[i][j] >= minU && Uptr[i][j] <= maxU))
        {
            ptr[i][j].r = 0;
            ptr[i][j].g = 0;
            ptr[i][j].b = 0;
            continue;
        }
        if (!(Vptr[i][j] >= minV && Vptr[i][j] <= maxV))
        {

            ptr[i][j].r = 0;
            ptr[i][j].g = 0;
            ptr[i][j].b = 0;
            continue;
        }
        else
        {
            ptr[i][j].r = 255;
            ptr[i][j].g = 255;
            ptr[i][j].b = 255;
        }
    }
    dib = temp;
}
```



bluetooth.cpp
brick.cpp
CircleQueue.cpp
Client.cpp
color.cpp
compass.cpp
DefaultSensor.cpp
DefaultSensorMsgPacket.cpp
Dib.cpp
DibBmp.cpp
DibOpenCV.cpp
Element.cpp
error.cpp
filesystem.cpp
function.cpp
i2c.cpp
imgfeatures.cpp
io_8574.cpp
kdtree.cpp
minpq.cpp
motor.cpp
network.cpp
NXTBrick.cpp
NXTHand.cpp
NXTHandMsgPacket.cpp
NXTMsgPacket.cpp
NXTTools.cpp
RGBBYTE.cpp
sensor.cpp
sift.cpp
sonar.cpp
stdafx.cpp
testver.cpp
testverDlg.cpp
TETRIXDrive.cpp
TETRIXDriveMsgPacket.cpp
TETRIXElbow.cpp

```
}

//device dependent
unsigned int Sensor::read_raw( ){
  get_sensor_value( );
  return raw_AD_value;
}

//type dependent
unsigned int Sensor::read_normalized( ){
  get_sensor_value( );
  return normalized_AD_value;;
}

//
Sensor_type Sensor::get_type( ){
  return sensor_type;
}

void Sensor::reset(bool reply){
  unsigned char answer[NXT_BUFFER_SIZE];
  unsigned char command[5];
  command[0]=0x03;  //command length
  command[1]=0x00;

  //start of message
  if(reply){
    command[2]=0x00;
  }
  else{
    command[2]=0x80;
  }
  command[3]=0x08;
  command[4]=sensor_port;
  connection->send(&command[0],5);
  if(reply){
    connection->receive(&answer[0],5);
    if(answer[4]){
      throw Nxt_exception::Nxt_exception("reset","Sensor", 0x00FF & answer[4]);
    }
  }
}
```


하얗게 불태웠어..

다음에는

Part 2. 윈도우 프로그래밍 변천사
(WinAPI32 - MFC - WPF)

Part 3. 윈도우 기반 방화벽 만들기

Part 4. U-health system

Part 5. 스마트TV 해킹하기

Part 6. Live Migration for VM

Thank you