



국민대학교  
전자정보통신대학  
컴퓨터공학부


# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	<i>Eye tracking Kiosk</i>
팀 명	<i>Eye can do it!</i>
문서 제목	결과보고서

Version	1.3
Date	2019-MAY-28

팀원	이용훈 (조장)
	윤진교
	한민인
	최휴영
	우리젠

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 "Eye tracking Kiosk"를 수행하는 팀 "Eye can do it"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "Eye can do it"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


<b>Filename</b>	결과보고서-15 조.doc
<b>원안작성자</b>	이용훈, 윤진교, 한민인
<b>수정작업자</b>	이용훈, 윤진교, 한민인

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-05-25	이용훈	1.0	최초 작성	최초보고서 작성
2019-05-26	윤진교	1.1	내용 수정	내용 보강 및 중간 수정
2019-05-27	한민인	1.2	내용 수정	오타 확인, 전체적인 내용 확인 및 수정
2019-05-28	이용훈	1.3	최종 수정	목차 수정 및 내용 점검

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
2	개발 내용 및 결과물	6
2.1	목표	6
2.2	연구/개발 내용 및 결과물	7
2.2.1	연구/개발 내용	7
2.2.1.1	얼굴, 눈 영역 검출에 따른 동공 영역 검출	7
2.2.1.2	검출된 영역의 좌표값을 기준으로 포인터 이동	7
2.2.1.3	포인터 출력 및 클릭 기능 구현	8
2.2.1.4	동공의 떨림 보완, 정확도 향상	8
2.2.1.5	키오스크 웹 구현	9
2.2.2	시스템 기능 요구사항	10
2.2.3	시스템 비기능(품질) 요구사항	11
2.2.4	시스템 구조 및 설계도	11
2.2.5	활용/개발된 기술	12
2.2.6	현실적 제한 요소 및 그 해결 방안	13
2.2.7	결과물 목록	14
2.3	기대효과 및 활용방안	15
2.3.1	기대효과	15
2.3.2	활용방안	15
3	자기평가	16
4	참고 문헌	17
5	부록	18
5.1	사용자 매뉴얼	18
5.2	배포 가이드	19
5.3	테스트케이스	19

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

# 1 개요

## 1.1 프로젝트 개요

현재 널리 사용되고 있는 키오스크(무인단말기)는 비장애인에게 편리하지만 손을 자유롭게 사용하지 못하는 장애인은 사용할 수 없다는 단점을 가지고 있다. 따라서 이 점을 보완하고자 시선 추적(아이트래킹) 기술을 이용하여 손을 자유롭게 사용하지 못하는 장애인을 위한 키오스크 시스템을 개발한다.

## 1.2 추진 배경 및 필요성

최저임금 상승에 따른 인건비 증가에 따라 사회 곳곳에서 효율성이 높은 키오스크를 사용하는 추세이다. 국내 프랜차이즈, 은행, 병원 등 대다수의 기관에서도 이미 키오스크를 사용중인 모습을 흔하게 볼 수 있다.



[그림 1] 국내 키오스크 시장 규모 추이 (출처, 신한금융투자)

비장애인들에게는 간편하고 빠른 주문을 도와줄 수 있는 편리한 장치이지만 손을 자유롭게 사용하지 못하는 장애인들에게 키오스크는 무용지물이다. 시각장애인들을 위한 음성인식 및 점자 시스템을 도입한 키오스크는 비교적 자주 볼 수 있지만 그에 비해 두 손을 자유롭게 사용할 수 없어 터치를 할 수 없는 척수장애인을 위한 키오스크는 거의 찾아볼 수 없다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

현재 정부에서는 지방자치단체 및 은행을 제외한 장소에 장애인용 키오스크 설치에 대한 규정을 제정하지 않아 대부분의 사업장에서는 장애인용 키오스크가 설치되어 있지 않으며, 키오스크 제작 회사에서도 장애인용 키오스크를 개발하기 위해서는 많은 비용과 시간이 들기 때문에 제작을 꺼려하는 입장이다.

따라서 손을 사용하기 어려운 장애인들에게 비교적 자유로운 부분인 눈을 이용하여 조작할 수 있는 아이트래킹을 기반으로 한 키오스크 시스템을 개발하고자 한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28


## 2 개발 내용 및 결과물

### 2.1 목표

본 프로젝트는 최근 많은 곳에서 사용되고 있는 키오스크를 사용하기 어려운, 손을 자유롭게 사용하지 못하는 장애인들을 위하여 시선추적 기술을 기반으로 한 키오스크를 제작하는 것이다. 시선추적 기술을 사용하여 장애인들은 눈으로 손의 기능을 대신할 수 있으며, 키오스크를 통하여 원하는 정보를 얻거나 상품을 주문할 수 있게 된다.

또한, 실제 키오스크 시스템처럼 Web Front-End 와 Back-End 를 구현한다. Web Front-End 에는 상품의 가격, 사진, 정보, 장바구니 등을 구현한다. Web Back-End 에는 상품이나 정보를 데이터베이스에 저장하고, 사용자가 선택한 상품을 장바구니에 담아 주고 주문한 상품을 데이터베이스에 저장한 후에 Owner 에게 전달해주어 실제 키오스크를 사용하는 것과 같이 구현할 예정이다.

시선추적 기술을 적용하여 손을 사용하지 않고도 시선의 움직임, 얼굴의 움직임을 통하여 원하는 상품정보를 얻을 수 있으며 눈을 깜빡이는 특정한 행동을 통하여 상품을 선택하고 주문할 수 있다. 기존 키오스크에 add-on 형식으로 웹캠과 해당 소프트웨어를 실행하기만 하면 사용할 수 있도록 간편한 서비스를 제공하는 것을 목표로 한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 2.2 연구/개발 내용 및 결과물


### 2.2.1 연구/개발 내용

#### 2.2.1.1 얼굴, 눈 영역 검출에 따른 동공 영역 검출

OpenCV 를 사용하여 웹캠으로 사용자의 영상을 가져오게 되며 Dlib 을 사용하여 사용자의 얼굴영역을 검출하게 된다. 얼굴영역을 검출하기 위해서는 학습된 랜드마크 모델 데이터인 'shape\_predictor\_68\_face\_landmarks.dat'을 사용한다. 이 데이터는 사람의 얼굴에 68개의 점을 찍어 랜드마크로 지정해주며 숫자에 따라 특정 얼굴 영역을 나타내게 된다. 그 중에서도 얼굴 전체 영역에서 눈 영역을 찾아 그 영역을 따로 정의해준다.

#### 2.2.1.2 검출된 영역의 좌표값을 기준으로 포인터 이동

2.1.1 에서 검출해낸 영역에서 색 반전을 통하여 눈의 검정색 동공 부분은 흰색으로, 나머지 부분은 검정색으로 변환해준다. 그 후 OpenCV 의 GaussianBlur 를 사용하여 이미지의 노이즈를 제거해주고 findContours 함수를 사용하여 동공의 영역인 흰색 부분에 외곽선을 찾아준다. 그 후 찾아낸 동공영역 중심을 기준으로 하여 Pymouse 라이브러리를 사용하여 포인터를 이동시킨다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28


### 2.2.1.3 포인터 출력 및 클릭 기능 구현

2.1.2 에서 동공을 찾아 포인터를 이동시키면 Python 라이브러리인 Pymouse 를 사용하여 마우스를 제어한다. 클릭 기능을 구현하기 위해서는 사용자의 특정한 행동을 정의해야 하는데, 단순히 눈을 깜빡일 때 클릭이 되도록 구현하게 될 경우 일반적인 눈 깜빡임에도 반응하여 무의미한 클릭이 방해요소가 될 수 있을 것이다. 따라서 눈을 길게 한 번 깜빡일 경우 클릭이 가능하도록 설정하였다. 이 때 눈의 깜빡임은 2.1.1 에서 사용한 랜드마크 모델 데이터에서 눈 윗부분의 y 축 좌표와 눈 아랫부분의 y 축 좌표를 추출하여 두 좌표 사이의 거리가 일정 수준 미만으로 작아질 경우에 눈을 깜빡였다고 인식하며, 일정 프레임 이상 두 좌표 사이의 거리가 작아지면 클릭이 될 수 있도록 설정하였다.

### 2.2.1.4 동공의 떨림 보완, 정확도 향상

위의 2.1.1~3 을 수행하면서 동시에 진행한 문제이다. 한 곳을 정확하게 응시하고 있더라도 동공의 중심은 미세하게 떨리고 있기 때문에 검출되어 표시한 점도 지속적으로 흔들리게 되어 불안정한 모습을 보여준다. 하지만 키오스크의 특성상 상품의 정보를 얻고 선택을 하는 것을 위주 작동하는 기계이기 때문에 시선의 모든 위치를 추적할 필요가 없다고 판단하였다. 따라서 화면을 분할하여 시선이 기준에서 일정 범위 이상 벗어날 경우 특정 위치에 값을 고정시킬 수 있도록 하였다. 그렇게 할 경우 포인터의 이동을 기다릴 필요 없이 즉각적으로 버튼 위에 포인터가 위치하기 때문에 더욱 효율적인 사용이 가능할 것이다.



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

### 2.2.1.5 키오스크 웹 구현

아이트래킹 키오스크가 실생활에서도 충분히 문제없이 사용될 수 있음을 보여주기 위하여 데모환경을 최대한 실제상황과 유사하게 구현해야 한다.


현재 사회 곳곳에서 사용되는 대부분의 키오스크들은 우리가 익숙한 윈도우 환경에서 운영되고 있으며, 이러한 실제 키오스크와 최대한 유사하게 구현하기 위하여 실시간으로 고객들과 주문정보를 빠르게 주고 받을 수 있는 웹서버를 사용한다.

키오스크 데모는 일상에서 흔히 볼 수 있는 카페의 커피주문 키오스크를 모델로 했으며, 실제 키오스크처럼 뜨거운 음료, 차가운 음료, 기타 음료 등을 고를 수 있으며 수량도 조정할 수 있게 구현하였다. 또한 물품을 선택하고 장바구니에 담았다가 일괄 주문하는 식의 방법을 선택하였다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 2.2.2 시스템 기능 요구사항

대분류	소분류	기능	완료여부
Eye tracking	눈 영역 검출	Dlib 을 사용하여 동공의 시선을 추적할 수 있도록 먼저 눈 영역을 검출한다	완료
	동공 검출	검출한 눈영역에서 눈의 검은자 부분을 색반전을 통하여 외곽선을 추출하고, 그 영역 안에 중심 부분인 동공을 찾는다.	완료
	포인터 이동	동공을 검출한 후에 그 좌표를 기준으로 움직임을 인식하여 포인터를 이동시킨다.	완료
	포인터 클릭	눈을 일정 프레임 이상 깜빡이는 행동을 통하여 현재 포인터의 위치를 클릭한다.	완료
Kiosk Web	키오스크 UI/UX	현재 상용화 되어있는 키오스크의 UI/UX 를 최대한 유사하게 구현하였다	
	팝업창	선택한 음료의 사이즈, 샷 추가, 수량 선택할 수 있는 팝업창 구현	
	알림 prompt 창	최종결정을 확인하기 위한 알림창 구현	


 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

### 2.2.3 시스템 비기능(품질) 요구사항

분류	요구사항	완료여부
신뢰성	사용자의 시선을 추적할 때 80% 이상의 정확성을 가져야한다.	완료
	데이터 초기화는 100% 정확성을 가져야 한다.	완료
효율성	빠른 시간 내에 사용자의 주문데이터가 매장 측으로 전달되어야 한다.	완료
	현재 창에서 원하는 정보 창으로 이동하는 시간은 3 초 이내이어야 한다.	완료

### 2.2.4 시스템 구조 및 설계도



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 2.2.5 활용/개발된 기술

### 1. Eye tracking

Dlib 라이브러리를 사용하여 학습된 랜드마크 모델 데이터를 사용하였고, 그 데이터를 이용하여 얼굴 영역을 검출하였다. 또한 실시간 영상 입력 및 노이즈 제거, 흑백 처리 등 영상 처리를 하기 위하여 OpenCV 를 사용하였다. 포인터의 이동 및 클릭을 위해서 Python 라이브러리인 Pymouse 를 사용하였다.

### 2. Kiosk Web

현재 사회 곳곳에서 사용되는 대부분의 키오스크들은 우리가 익숙한 윈도우 환경에서 운영되고 있으며, 이러한 실제 키오스크와 최대한 유사하게 구현하기 위하여 실시간으로 고객들과 주문정보를 빠르게 주고 받을 수 있는 웹서버를 사용했다.

웹의 전체적인 틀은 시간단축을 위하여 웹 프레임워크인 bootstrap 을 사용하였다.

키오스크 데모는 일상에서 흔히 볼 수 있는 카페의 커피주문 키오스크를 모델로 했으며, 실제 키오스크처럼 뜨거운 음료, 차가운 음료, 기타 음료 등을 고를 수 있으며 수량도 조정할 수 있게 구현하였다. 또한 원하는 물품을 선택하고, 음료의 수량, 사이즈, 샷 추가를 조정한 뒤에 장바구니에 담았다가 일괄 주문하는 식의 방법을 선택하였다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 2.2.6 현실적 제한 요소 및 그 해결 방안

### 1. 모니터, 웹캠과 사용자 간의 거리

모니터와 사용자가 멀리 떨어져 있을 경우, 상품을 선택할 때 사용자의 시야에 모니터가 한 눈에 들어와 동공의 움직임이 거의 없어진다. 따라서 동공의 움직임이 생기기 위해서는 일정거리 안에 사용자가 위치해 있어야 한다. 또한 웹캠과의 거리도 마찬가지로 동공의 움직임이 생기게 하기 위하여 모니터와 비슷한 거리를 유지해야 한다. 또한 웹캠과의 거리가 멀어지면 얼굴에 들어오는 빛의 양이 달라져 Threshold 값을 변경시켜줘야 하기 때문에 일정 거리에 사용자를 고정시켜야 값을 변경시키지 않고도 프로그램이 원활하게 실행될 수 있다.


### 2. 동공의 상하 움직임

현재 눈의 색반전을 통하여 동공을 찾고 움직임을 인식하여 포인터를 움직이는 방법을 사용하고 있다. 눈을 살펴보면 흰자는 검은자의 좌우를 감싸고 있기 때문에 동공의 좌우 움직임은 쉽게 구현할 수 있지만, 검은자의 위아래에는 흰자가 없기 때문에 눈의 상하 움직임을 감지하기 어렵다. 따라서 자연스러운 움직임을 구현하기 위하여 포인터의 상하 이동은 동공이 아닌 눈 영역의 중심을 찾아 사용자의 눈 위치를 고정된 값으로 설정시킨 후에 그 중심이 설정된 범위 이상으로 벗어나면 상하 움직임을 할 수 있도록 구현했다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 2.2.7 결과물 목록

대분류	소분류	기능	기술문서 유/무
시선 추적 기술	얼굴 및 눈 영역 검출	Dlib 과 OpenCV 를 사용하여 얼굴 및 눈 영역을 검출 및 표시한다.	
	동공 영역 검출 및 표시	아이트래킹을 위한 동공의 중심을 찾고 그 부분을 점으로 표시한다.	
	동공 영역 좌표값	동공의 중심을 좌표로 나타내 준다.	
	포인터 이동 및 클릭	계산된 이동거리의 값을 바탕으로 포인터 이동 및 클릭기능 구현	
	포인터 특정 좌표 이동	포인터를 특정한 좌표에 이동시킨다.	
Front-end	키오스크 웹페이지	HTML, CSS, JS 를 사용하여 키오스크 구현	
	구입하고 싶은 음료의 종류 선택	실제 카페의 키오스크처럼 Hot, Iced, Beverage 로 페이지 구분	
	장바구니 구현	이용자가 원하는 제품을 장바구니에 담아 수령조정 및 최종 확인 후 일괄결제	

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28


## 2.3 기대효과 및 활용방안

### 2.3.1 기대효과

- 실생활에 퍼져 있는 기존의 키오스크를 장애인들이 이용할 때에는 주변에 비장애인들의 도움이 필요했다. 아이트래킹 기술을 이용해 장애인들을 위한 키오스크 시스템을 개발하여 장애인들이 스스로 사용할 수 있게 하고, 사용할 때 불편함과 소외감을 없애 준다.
- 또한 점주 입장에서도 기존에 설치되어 있는 키오스크에 웹캠만 add-on 하면 되기 때문에 많은 비용을 들이지 않고 장애인을 위한 시스템을 도입할 수 있어 높은 보급률을 기대할 수 있을 것이다.

### 2.3.2 활용방안

- 시선 추적 기술, 키오스크 웹 화면 및 데이터베이스, 서버를 구현하는 전체적인 키오스크 시스템을 제작하여 키오스크가 도입되어 있지 않은 곳에서 시선추적 기술 기반 키오스크를 도입할 수 있으며, 이미 키오스크가 도입되어 있는 곳에서도 add-on 형식으로 웹캠과 소프트웨어를 설치하면 기존 키오스크에서 시선추적이 가능한 키오스크로 활용할 수 있다.
- 매장 뿐만 아니라 공항, 관광지 등 정보안내를 위한 곳에서도 장애인을 위한 시스템을 활용할 수 있을 것이다.
- 시선 추적 기술을 이용하여 키오스크가 아닌 다른 전자기기 혹은 프로그램에도 접목시켜 다양한 분야에서 장애인들의 접근성을 높일 수 있다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

### 3 자기평가

처음 프로젝트 주제를 선정할 때, 아이디어는 불편으로부터 나오는 경우가 많기 때문에 개인 혹은 주변에서 불편함을 겪고 있는 사례를 많이 찾아보았다. 그 중에서도 특히 장애인들은 비장애인들에 비해 많은 불편함을 겪고 있는 경우가 많다. 여러 기사를 찾아보다가 휠체어나 손을 자유롭게 사용하지 못하는 장애인들이 키오스크를 이용하는데 불편함을 많이 겪는다는 사실을 알게 되었다. 현재 많은 곳에서 인력을 대체하기 위하여 키오스크를 도입함에 따라 이러한 불편함을 겪는 장애인들 또한 증가하는 문제점이 생겼다. 이를 해결하기 위하여 손을 대신할 수 있는 방법을 찾아보았고, 시선추적 기술을 사용하여 눈의 움직임으로 손의 사용을 대체할 수 있을 것이라고 판단하였다.

시선추적 기술을 사용할 때 Pygaze 라는 오픈소스를 발견하여 활용하려고 했으나, 설치방법이 까다롭고 사용방법이 제대로 설명되어 있지 않아 비교적 사용법이 쉽고 영상처리에 다방면으로 사용되는 OpenCV 를 사용하기로 했다

팀원들 모두 Github 를 개인 과제에만 사용한 경험이 있어, 처음 팀 프로젝트를 시작할 때 사용법을 숙지하는데 오랜 시간이 걸렸다. 개인 branch 를 생성하여 issue 에 따라 관리하였지만 중간에 혼동이 생겨 충돌이 발생하기도 하였고, 그것을 복구하는데 꽤나 많은 시간이 소요되기도 하였다. 하지만 그런 시행착오를 겪고 난 후에 충돌에 대한 조심성을 가지게 되었고 각자의 branch 를 잘 관리하게 되어 수월하게 협업을 진행할 수 있었다.



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	서적	가상현실과 증강현실 기반 주요 게임 활용사례 및 연구개발 동향과 참여업체 현황	좋은정보사	2016.11.09	좋은정보사	
2	기사	"보편화된 키오스크 시대에 소외된 장애인"	에이블뉴스	2019.01.04	김경식	
3	온라인 백과사전	아이트래킹	위키피디아			
4	사이트	아이트래킹 제품 정보	Tobii Technology			
5	기사	"엘리비전, 사회약자·장애인 전용 무인결제 키오스크 개발...12 월 런칭"	파이낸셜뉴스	2018.11.22	김경아	
6	보고서	Global Eye Tracking Devices Market 2017-2021	technavio	2017.07		
7	서적	러닝! OpenCV 3	위키북스	2018.04.19	아드리안 캘러 / 개리 브래드스키	
8	서적	파이썬 코딩 도장	길벗	2018.11.30	남재윤	
9	서적	Do it! 자바스크립트 + 제이쿼리 입문	이지스퍼블리 싱	2018.04.06	정인용	

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 5 부록

### 5.1 사용자 매뉴얼

1. 이 프로그램을 사용하기 위해서는 OpenCV 3.4, Dlib, Pymouse 의 설치가 필요하다. 또한 Pymouse 를 사용하기 위해서 PyUserInput 이 필요하기 때문에 함께 설치를 해주도록 한다.  
<https://opencv.org/opencv-3-4/> - OpenCV 3.4  
<https://pypi.org/project/dlib/> - Dlib  
<https://pypi.org/project/PyMouse/> - Pymouse  
<https://pypi.org/project/PyUserInput/0.1/> - PyUserInput
2. 처음 사용자를 인식하게 되면 잠깐의 timeout 이후에 인식된 위치를 기준으로 인식하게 된다. 기준점이 잘못 설정될 경우 얼굴이 인식되지 않게 카메라의 범위 밖으로 갔다 다시 카메라에 얼굴을 인식시켜주면 된다.
3. 키오스크는 웹으로 구현하였기 때문에 F11 을 눌러 전체화면으로 실행한다(Firefox 기준)
4. 원하는 메뉴로 시선을 이동시킨 후, 눈을 길게 한 번 깜빡이는 행동으로 원하는 메뉴를 선택하면, 선택한 메뉴에 대한 pop-up 창이 뜬다
5. pop-up 창에서 선택 메뉴에 대한 사이즈, 샷 추가, 수량에 대한 추가 정보를 선택하면 원래 창의 장바구니로 선택한 메뉴에 대한 정보가 쌓인다
6. 장바구니를 최종확인 후 주문하기 버튼을 누른다.


 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

## 5.2 배포 가이드

[https://github.com/kookmin-sw/2019-cap1-2019\\_15](https://github.com/kookmin-sw/2019-cap1-2019_15) 에서 다운 받을 수 있다.

## 5.3 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
Eye tracking	얼굴 및 눈 영역 검출	Dlib 과 OpenCV 를 사용하여 얼굴 및 눈 영역을 검출 및 표시한다.	프로그램을 실행하여 창을 하나 띄운 후, 선 혹은 점으로 표시하여 눈 영역에 표시되는지 확인한다.	눈에 가로, 세로 선이 하나씩 그어진다..	성공
	동공 영역 검출 및 표시	시선추적을 위한 동공의 중심을 찾고 그 부분을 점으로 표시한다.	흑백화 된 이미지를 띄워주는 창을 별개로 설정하고, 해당 영역에 있는 눈의 검은자 부분이 흰색으로 출력되는지 확인한다.	동공영역이 검정 바탕에 흰색으로 표시된다.	성공
	동공 영역 좌표값	동공의 중심을 좌표로 나타내 준다.	동공 영역을 검출 한 후에 그 중심의 값을 print 하여 좌표가 출력되는지 확인한다.	일정한 좌표값이 출력된다.	성공
	포인터 이동 및 클릭	계산된 이동거리의 값을 바탕으로 포인터 이동 및 클릭기능 구현	화면 가운데를 기준으로 설정한 후에 상, 하, 좌, 우 움직임이 잘 이루어지는지 확인한다.	상, 하, 좌, 우로 움직여 원하는 곳에 포인터가 위치한다.	성공

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	Eye tracking Kiosk	
	<b>팀 명</b>	Eye can do it!	
	Confidential Restricted	Version 1.3	2019-MAY-28

Front-end	키오스크 웹페이지	HTML, CSS, JS 를 사용하여 키오스크 구현	index.html 을 Firefox 에서 열어본다	해당 키오스크의 메인 페이지가 열린다.	성공
	구입하고 싶은 음료의 종류 선택	실제 카페의 키오스크처럼 Hot, Iced, Beverage 로 페이지 구분	메인창에서 자신이 원하는 음료의 종류를 클릭한다	해당 음료의 종류들이 나열된다.	성공
	원하는 음료 선택	원하는 음료의 옵션을 선택한다	사용자가 원하는 제품을 클릭하면 새로운 popup 창이 열리고, 그 안에서 그 제품에 대한 추가 정보를 선택한다 ( 사이즈, 샷 추가, 수량)	선택정보가 메인화면의 장바구니에 추가된다.	성공
	장바구니 구현	이용자가 원하는 제품을 장바구니에 담아 수량조정 및 최종 확인 후 일괄 결제	원하는 음료들이 장바구니에 스택으로 쌓이게 된다. 사용자는 장바구니의 내용을 최종확인 후에 주문하기를 클릭한다	주문완료 창으로 이동하게 된다.	성공