

The background features a complex network of thin grey lines and dots, forming a web-like structure. Scattered throughout are various triangles of different sizes and orientations, some with solid grey outlines and others with dashed or dotted outlines. The overall aesthetic is technical and geometric.

Keyboard Projection And Detection

Filipe Gonçalves, 98083, 60%
Paulo Pereira, 98430, 40%



The Idea

What we wanted to accomplish

01

The Solution

What was possible to build in such a short time

02

TABLE OF CONTENTS

03

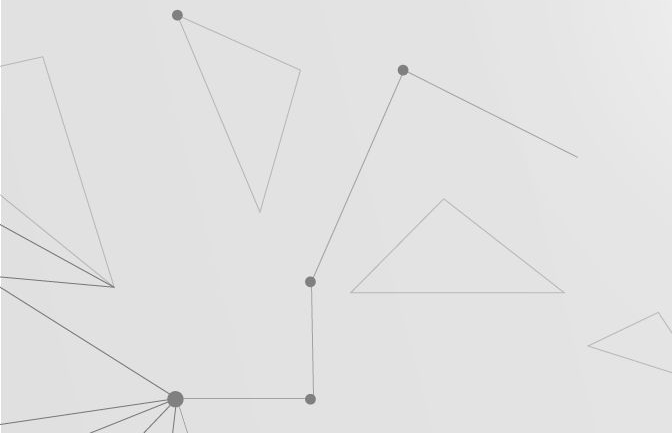
Difficulties

Obstacles we hit and workarounds

04

What's next ?

Work to be done next as the project never stops



01

The Idea

What we intended to build

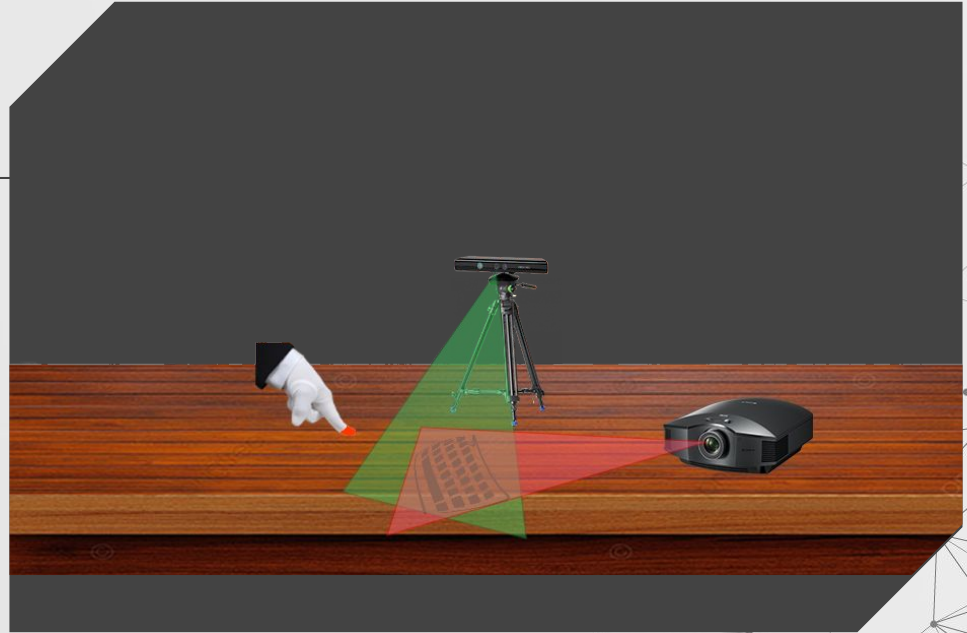


The Idea

Project the keyboard onto the table

Kinect Camera to detect depth and
record the keyboard

Glove with red point for finger
detection





02

The Solution

What was accomplished in the short time span that was given



Implementation

Keyboard
Segmenter

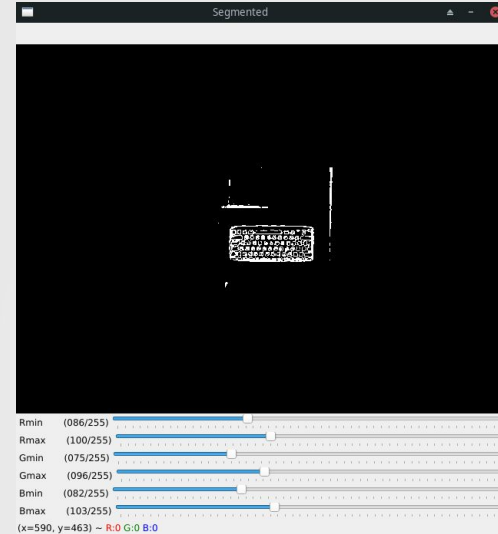
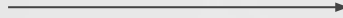
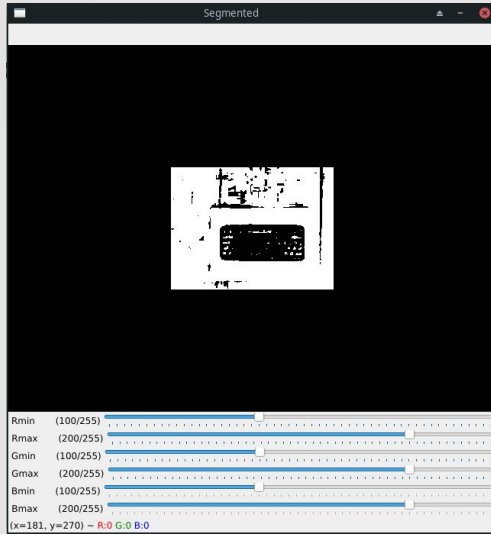
Centroid
Detection

Depth Detection

Finger Position

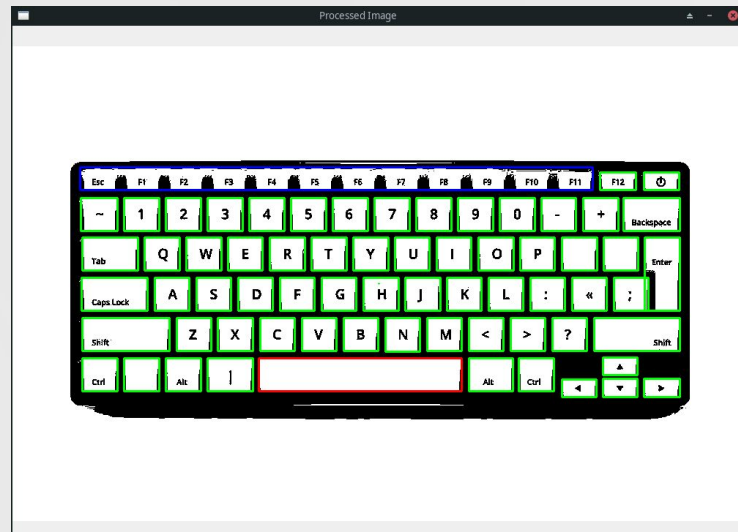
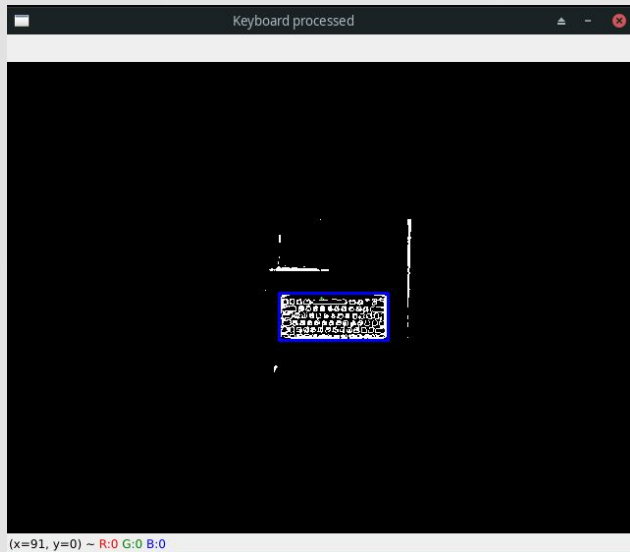
**Detect touch
on keyboard**

Keyboard Segmenter



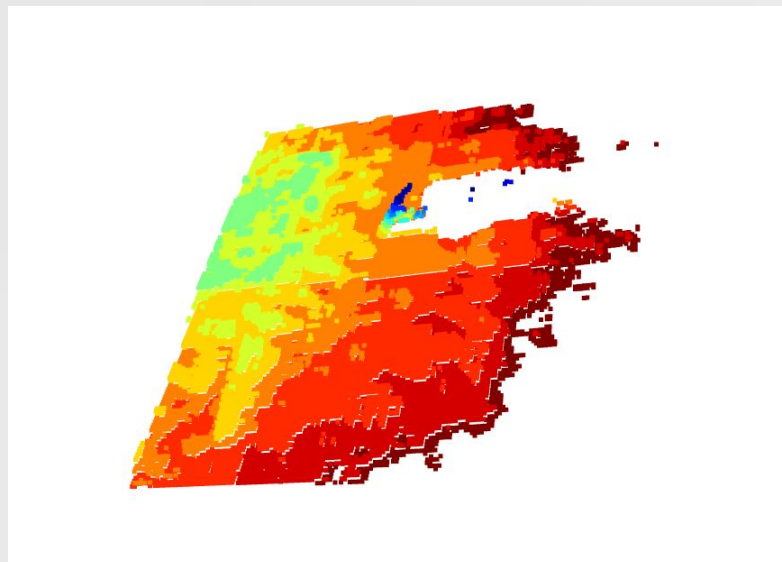
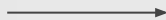
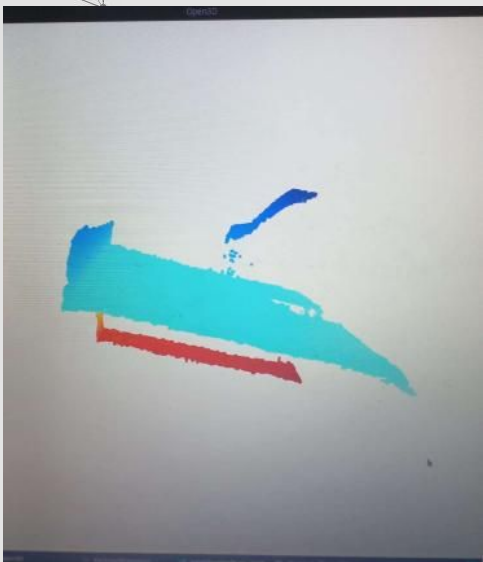
- Create a mask with the range values selected
- Logical_not the mask in the image
- Threshold all values different than 0 to be 1
- Save the range to use in the main file

Centroid Detection



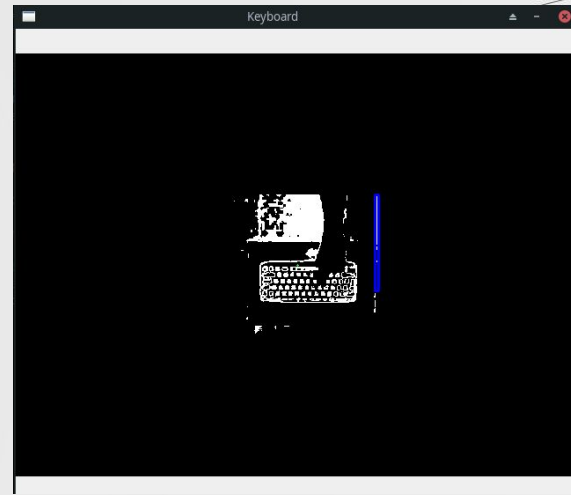
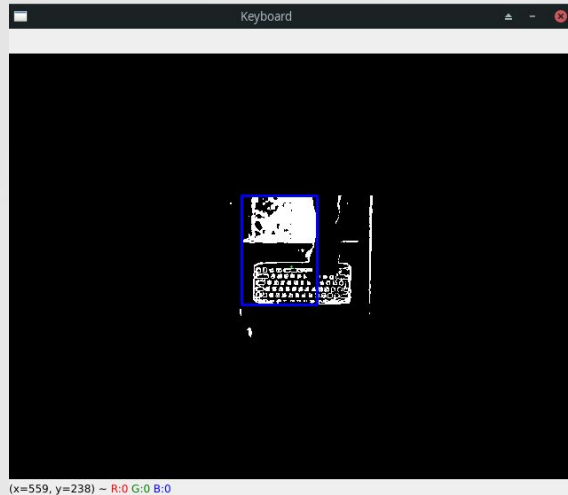
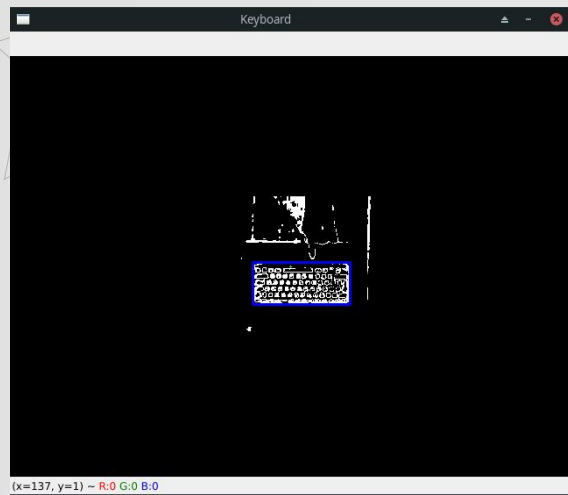
- Mask the image to only look into a small rectangle in the middle of the image
- Process the image with the range values segmented before
- `cv2.connectedComponentsWithStats` to get the centroids
- Biggest centroid after the whole image is the keyboard

Depth Detection



- Filtering point clouds to get the base of the board
- Detect when there is a big "shadow", as the hand is starting to move to the keyboard
- Detect when the hand is close enough to the board to deduce touch

Finger Position



- Process the frame only if there was touching deduced
- Identify the biggest centroid
- If the keyboard calculated is different from the calibrated (points are different, as area is always different because of the segmenter), there was touch!

03

Difficulties

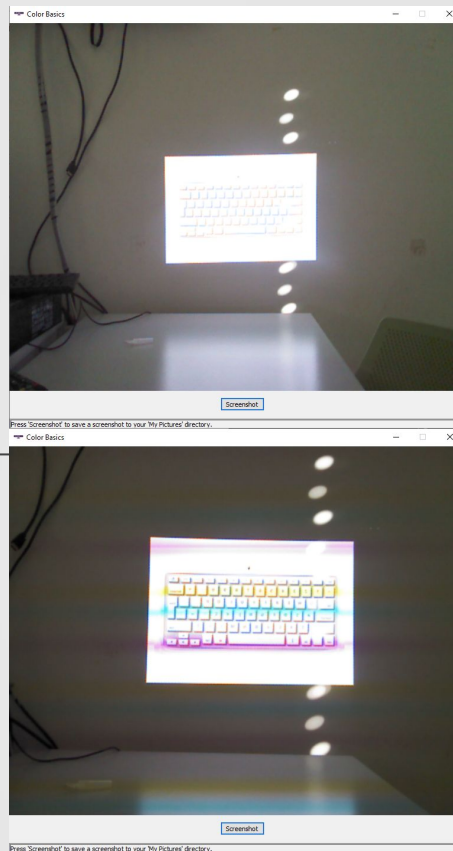
Obstacles we came across and how we worked around them



Difficulties

Refresh Rates

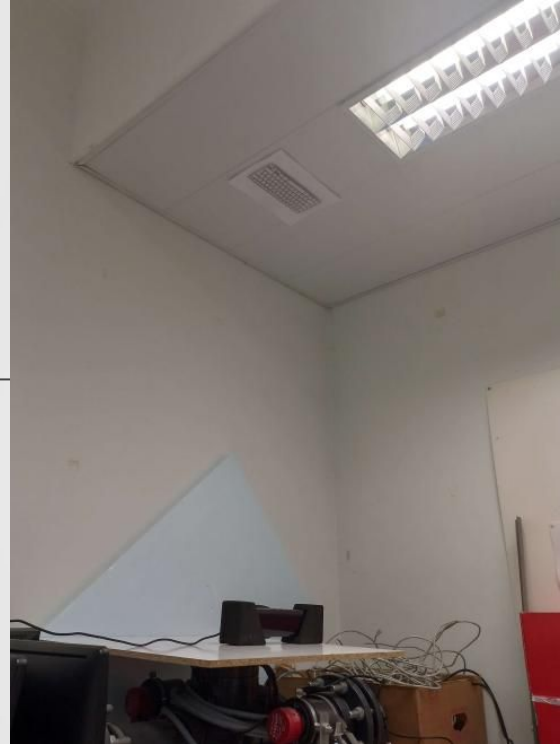
The projector and the Kinect camera both have different refresh rates and the video outputting from the camera had some color problems



Difficulties

Camera Setup

For a static image we needed to have the kinect fixed in a point, and the easiest mode was to use the safety blocks of the kinect as tripe and record a keyboard in the ceiling





Difficulties

Contact detection

The point clouds needed to be trimmed and segmented in order to find the correct baseline for the keyboard plane



Difficulties

Kinect Connection

The kinect camera only works in windows, and connect it to python is a difficult job, so we decided to record video and depth point clouds using Matlab and utilize them in the python code created prior

```
clear

diskLogger = VideoWriter("VideoColor.avi");
diskLogger.FrameRate = 30; % Sets the framerate of the recorded and saved avi video file

% Create the VIDEOINPUT objects for the two streams
colorVid = imaq.VideoDevice('kinect',1);
depthVid = imaq.VideoDevice('kinect',2);

open(diskLogger);

step(colorVid);
step(depthVid);

color = uint8(zeros(480,640,3,100));
depth = uint16(zeros(480,640,100));

a = 1

for i=1:100
    color(:,:,i) = step(colorVid);
    depth(:,:,i) = step(depthVid);
end

b = 186129476796318273916

% Stop the devices
release(colorVid);
release(depthVid);

for i=1:100
    ptCloud = pcfromkinect(depthVid,depth(:,:,i), color(:,:,i));
    pcwrite(ptCloud, sprintf('pointclouds/object3d%d.pcd',i),'Encoding','ascii');
end

writeVideo(diskLogger,color);
% imshow(colorFrameData);

close(diskLogger);
```



04

What's Next ?

Next steps to build the idea

What's Next



Camera

Better Kinect setup and video streaming directly to the code

Use the Projector to project the keyboard

Projector

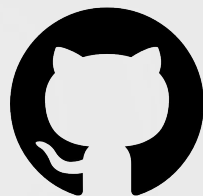


Calibration

Improve the settings and detection ranges for the hand touching the board as well as the hand above the keyboard



THANKS



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.