HCI Proj – Controlling your computer remotely by hands

11710106 欧阳晖

11710204 苗艳超

11710511 孙龙朝

Contents Table

- Project Setting
- Function Introduction
- Demo Video
- Function Implementation
- Acknowledges

Project Setting

- CMake 3.15.3

- OpenCV 4.2.0
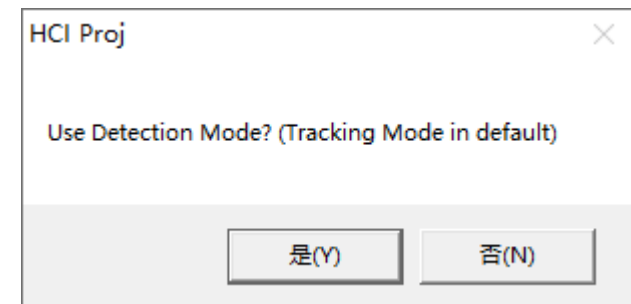- opencv_contrib 4.2.0

- MinGW-w64 x86_64 8.1.0

- CLion

Function Introduction

- Two Mode
  - Tracking Mode
  - Detection Mode

- Users can choose the mode freely
  - The default mode is Tracking mode
  - Choose the Detection mode when the user check the message box

- Press the ESC exit the program
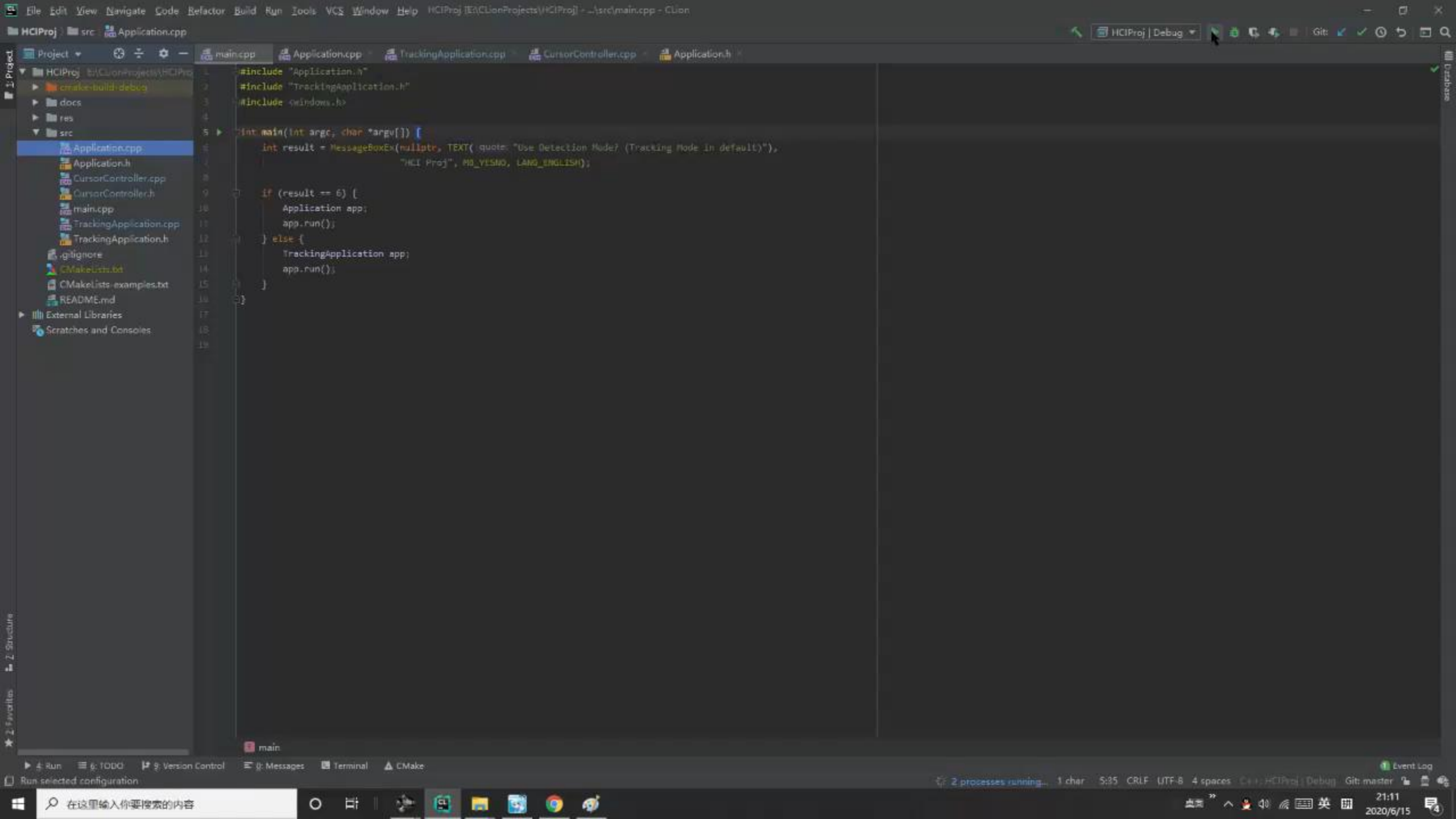
Function Introduction

- Tracking Mode
- More stable, but the user need to specify the object (hand)

  - specify the object (hand) first
  - press the **enter** or **space** in your keyboard

  - the tracker will track the object (with green rectangle)
  - the movement of the object will move the cursor

  - if you change your hand to fist
  - the cursor will double click (with red rectangle)

Function Introduction

- Detection Mode
- Not very stable, but the user need not to specify the object (hand)

    - the program will auto detect your hand
    - In palm, pink circle
    - In fist, blue circle

    - when your hand in palm, the cursor will keep click
    - when your hand in fist, the cursor will release click
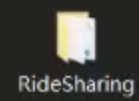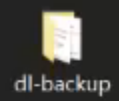   - when your hand move, the cursor also move

Demo Video

- Tracking mode (playing the games from 4399.com)

```cpp
#include "Application.h"
#include "TrackingApplication.h"
#include <windows.h>

int main(int argc, char *argv[]) {
    int result = MessageBoxEx(nullptr, TEXT( quote: "Use Detection Mode? (Tracking Mode in default)"),
                            "HCI Proj", MB_YESNO, LANG_ENGLISH);

    if (result == 6) {
        Application app;
        app.run();
    } else {
        TrackingApplication app;
        app.run();
    }
}
```
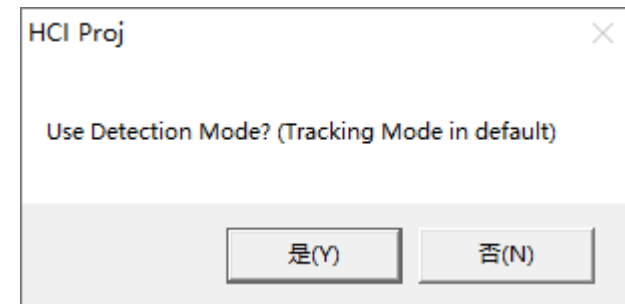
Demo Video

- Detection mode (opening the files)

Function Implementation

- Windows API
  - "windows.h"

- Mode Selection
  - A windows messages box from Windows API
  - Return 6 if choose YES
  - Detection mode selection

  - Return 7 if choose NO
  - Tracking mode in default

Function Implementation

- Cursor Controller
  - Further encapsulated class written by ourselves
  - Use the function which can control the cursor from Windows API
  - Set Cursor Position, Get Cursor Position, Move Cursor Position, Left Click, Right Click, Double Click and so on

- Camera Stream
  - Load each frame by VideoCapture from OpenCV library

- Exit Program
  - waitKey() function from OpenCV
  - Return 27 if the user press ESC button

Function Implementation

- Tracking Mode
  - Use CSRT Tracker & Cascade Classifier from OpenCV library

  - Select ROI and initialize the Tracker in the first frame
  - Update the Tracker for each frame, move the cursor by Cursor Controller (green rectangle)
  - Use the fist Classifier to detect the fist
  - If detect the fist, Double Click by Cursor Controller (red rectangle in next 3 frame)

  - Furthermore, use palm Classifier to avoid wrong fist detection (only double click when fist detect and no palm detect)

Function Implementation

- Detection Mode
  - Only use Cascade Classifier from OpenCV library

  - Use palm Classifier to detect palm (pink circle)
  - Use the fist Classifier to detect the fist (blue circle)

  - Set the relative Position of cursor by the detection of palm
  - Keep the left click on when detect fist
  - Release the left click when no detect fist

Function Implementation

- More Details
  - Reduce Jitter
    - Set a threshold
    - Just Move when the movement big enough
    - The effect is average, but better than the original

  - Solved wrong detection
    - The model detect the palm and fist is not very well
    - Use two Classifier in tracking mode reduce wrong click
    - Multiple flag values and logic in detection mode (still debugging)

  - Different scalar between the screen and camera
    - The model detect the palm and fist is not very well
    - Use two Classifier in tracking mode reduce wrong click
    - Multiple flag values and logic in detection mode (still debugging)

Acknowledges

- Open Source trained model to detection palm and fist
  - https://github.com/OAID/AndroidCVDemo/edit/master/app/src/main/cpp/handdetect/fist.xml
  - https://github.com/OAID/AndroidCVDemo/edit/master/app/src/main/cpp/handdetect/rpalm.xml

- Mike Chester Wang
  - Experience this app in the demo videos
  - Provide some useful help

Thanks for Your Listening

11710106　欧阳晖

11710204　苗艳超

11710511　孙龙朝

Q&A

11710106 欧阳晖

11710204 苗艳超

11710511 孙龙朝