Advanced Human-System Interfaces

Sixth Assignment, about hand / body gesture

Created by:

ANASS NASSIRI

Ultraleap Hand Tracking and 3D Object Rotation Visualization

Introduction

This project demonstrates how to interact with a 3D object (a cube) displayed on the screen using hand gestures captured by an Ultraleap sensor (Leap Motion). The main goal is to rotate the cube based on hand gestures and provide visual feedback with arrows indicating the direction of rotation.

Libraries and Setup

The program uses several libraries:
- **`<iostream>`**: For input and output operations.
- **`<LeapC.h>`**: Leap Motion API header for accessing Leap Motion functions and data structures.
- **`<GL/freeglut.h>`**: OpenGL Utility Toolkit for handling windowing operations and drawing.
- **`<Windows.h>`**: Windows-specific functions and macros.

Global Variables

- **`LEAP_CONNECTION connection`**: Holds the Leap Motion connection.
- **`const LEAP_TRACKING_EVENT* lastFrame = nullptr`**: Points to the last tracking frame received from the Leap Motion device.
- **`float rotationAngle = 0.0f`**: Keeps track of the cube's cumulative rotation angle.

Functions

1. **Drawing Functions**

- **`drawLine`**: Draws a line between two 3D points.
- **`drawSphere`**: Draws a sphere at a specified 3D position with a given radius.
- **`drawArrow`**: Draws an arrow at a given position with a specified direction.
- **`drawHand`**: Draws a hand using Leap Motion hand data, including the palm and fingers.
- **`drawCube`**: Draws a 3D cube with each face colored differently for visual distinction.

2. **OpenGL Display Function**

- **`display`**: Clears the screen, sets up the camera view, applies the cube's rotation, draws the cube, and displays rotation arrows. If hand tracking data is available, it draws the hands.

3. **OpenGL Idle Function**

- **`idle`**: Polls the Leap Motion device for new tracking data. If new data is received, it updates the rotation angle based on the hand's yaw (rotation around the vertical axis). It then triggers a redisplay.

4. **Initialization Functions**

- **`initLeapMotion`**: Initializes the Leap Motion connection, handling errors if the connection fails.
- **`initOpenGL`**: Sets up the OpenGL environment, initializes GLUT, creates a window, and sets up display and idle callbacks.

Main Function

```cpp
int main(int argc, char** argv) {
   initLeapMotion();
   initOpenGL(argc, argv);
   glutMainLoop();
   return 0;
}
```

- **`main`**: Initializes the Leap Motion and OpenGL, then starts the GLUT main loop to begin processing events.

How It Works

1. **Initialization**

- The Leap Motion connection is established, and OpenGL is initialized with necessary settings.

2. **Hand Tracking**

- The `idle` function continuously polls the Leap Motion device for tracking data. If a hand is detected, the yaw (rotation around the vertical axis) of the hand's palm is used to update the cube's rotation angle. This yaw value is converted from radians to degrees and accumulated.

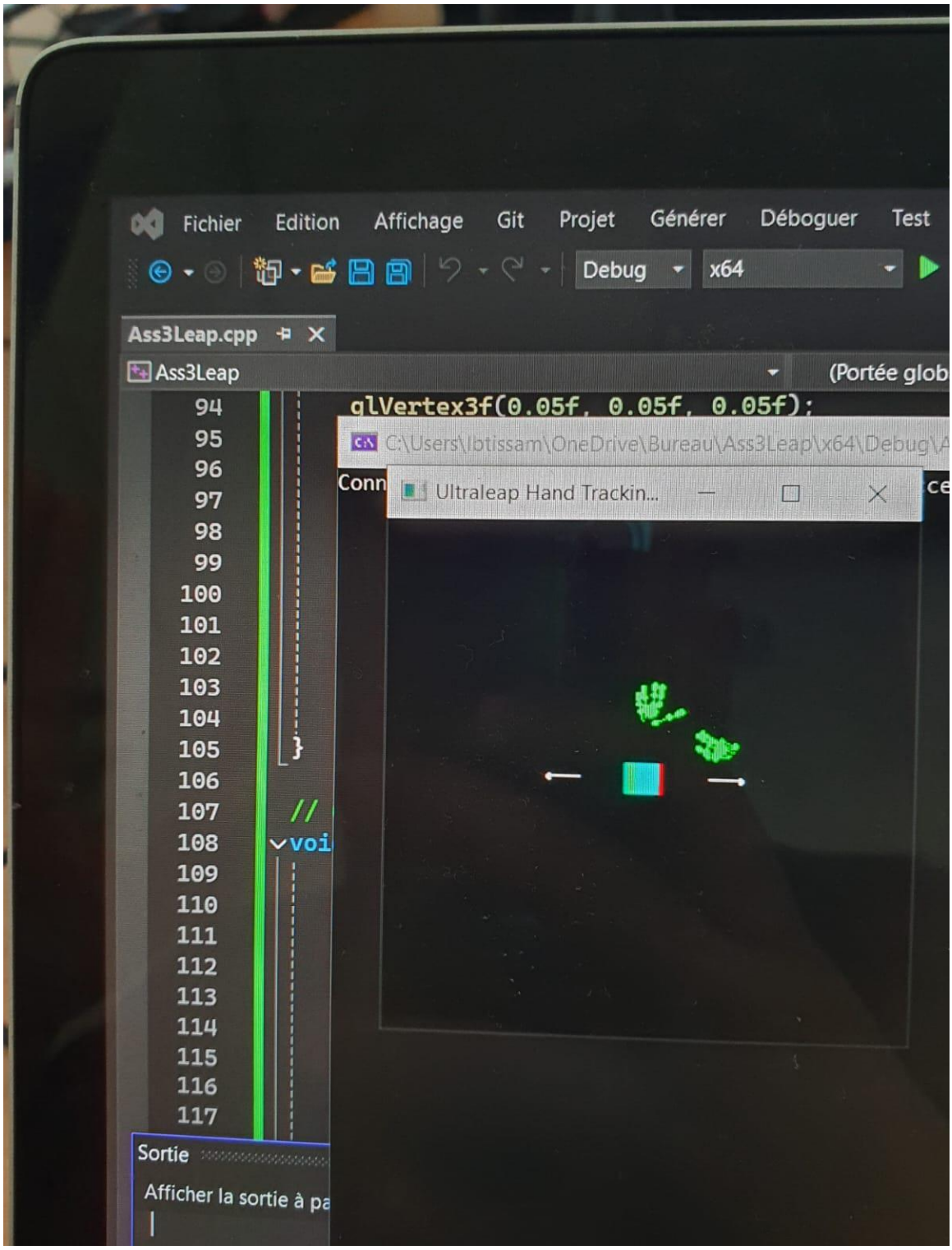
3. **Rendering**

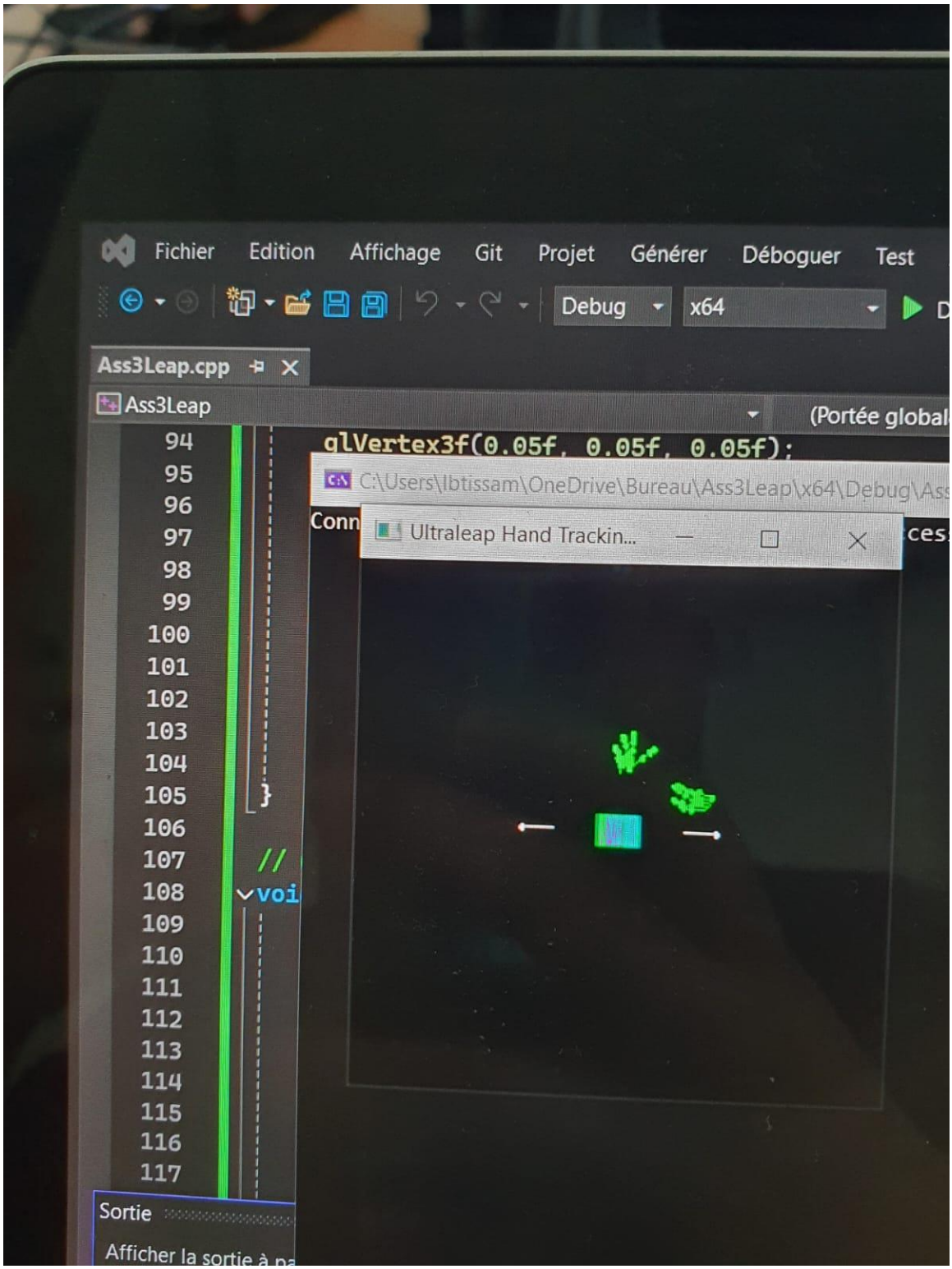
- The `display` function handles the rendering. It sets up the camera, applies the accumulated rotation to the cube, and draws it. Arrows indicating left and right rotation are drawn to provide visual feedback. If hand tracking data is available, it also draws the detected hands.

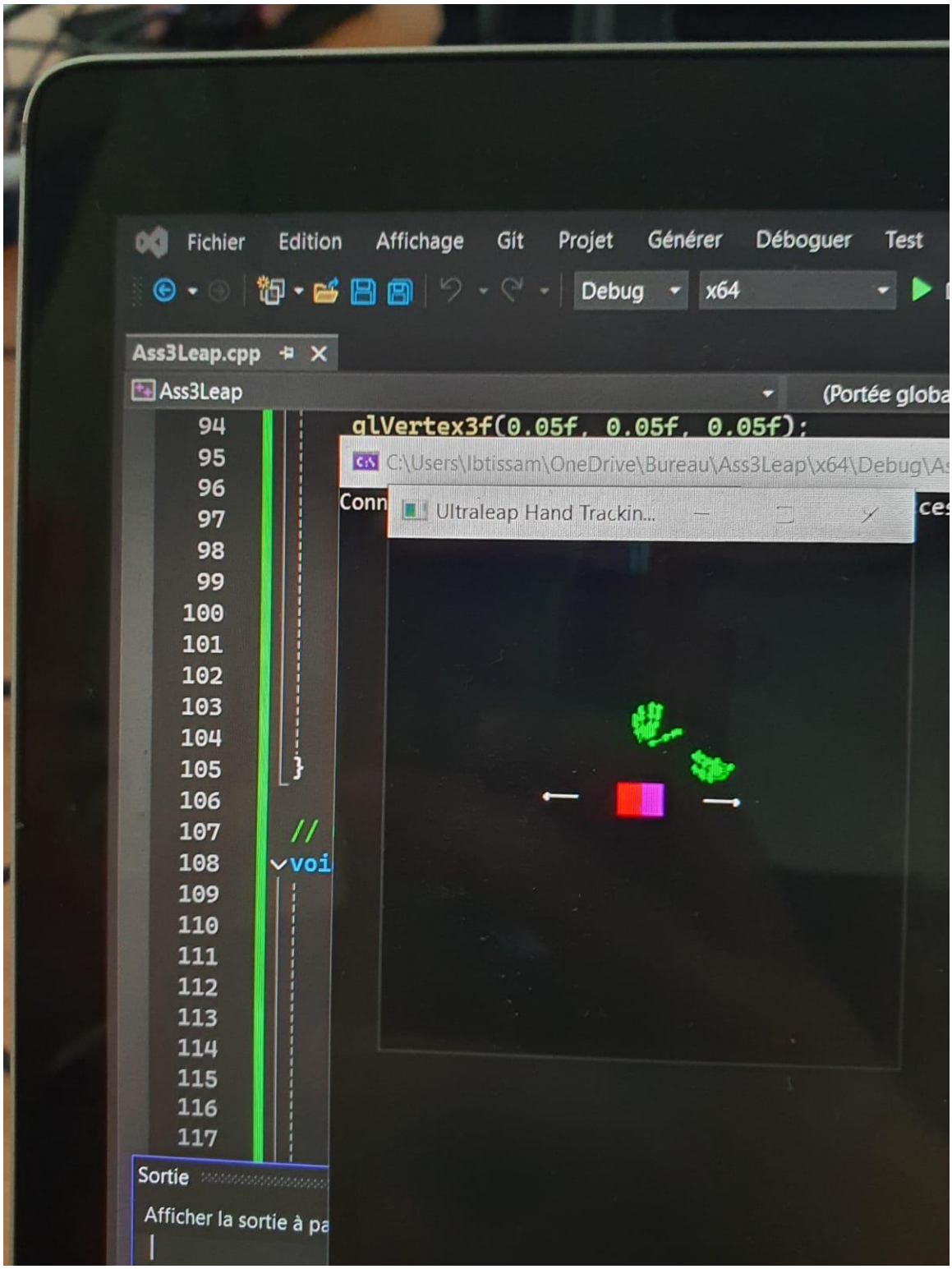
4. **Interaction**

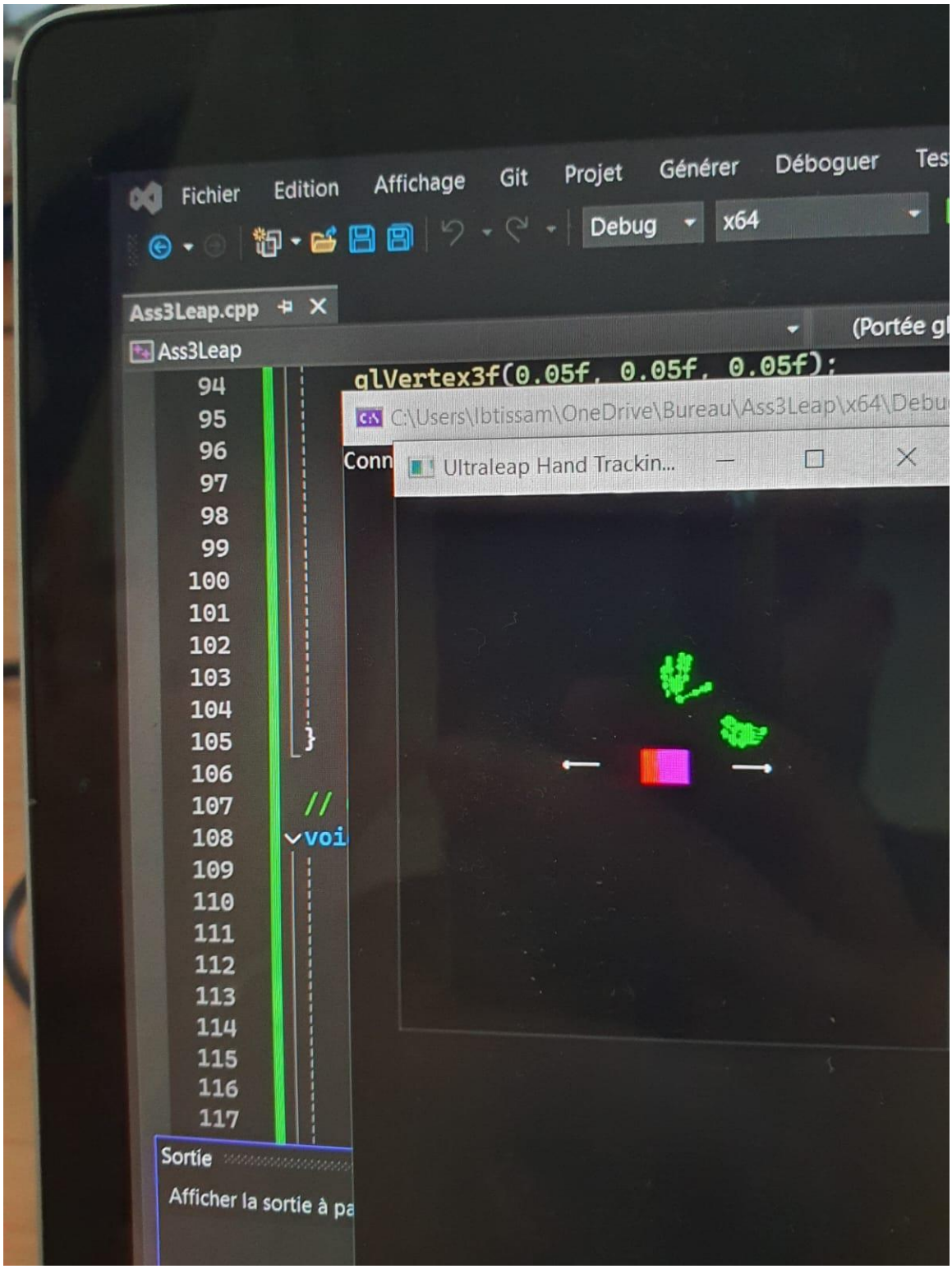
- As the user rotates their hand left or right, the yaw is detected, and the cube rotates accordingly. The arrows on the screen indicate the direction of rotation, providing an intuitive interaction experience.

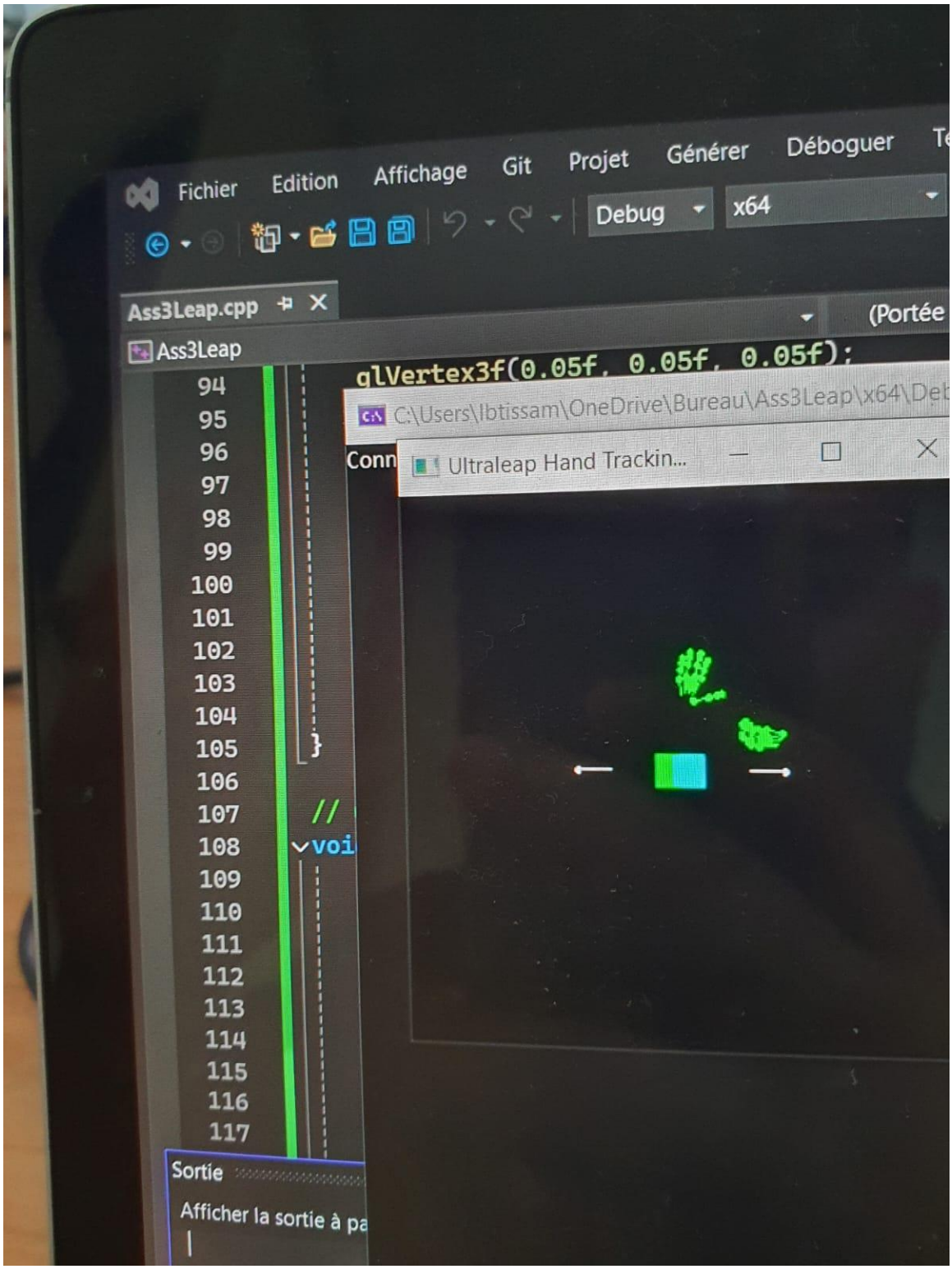
Some images of the application:

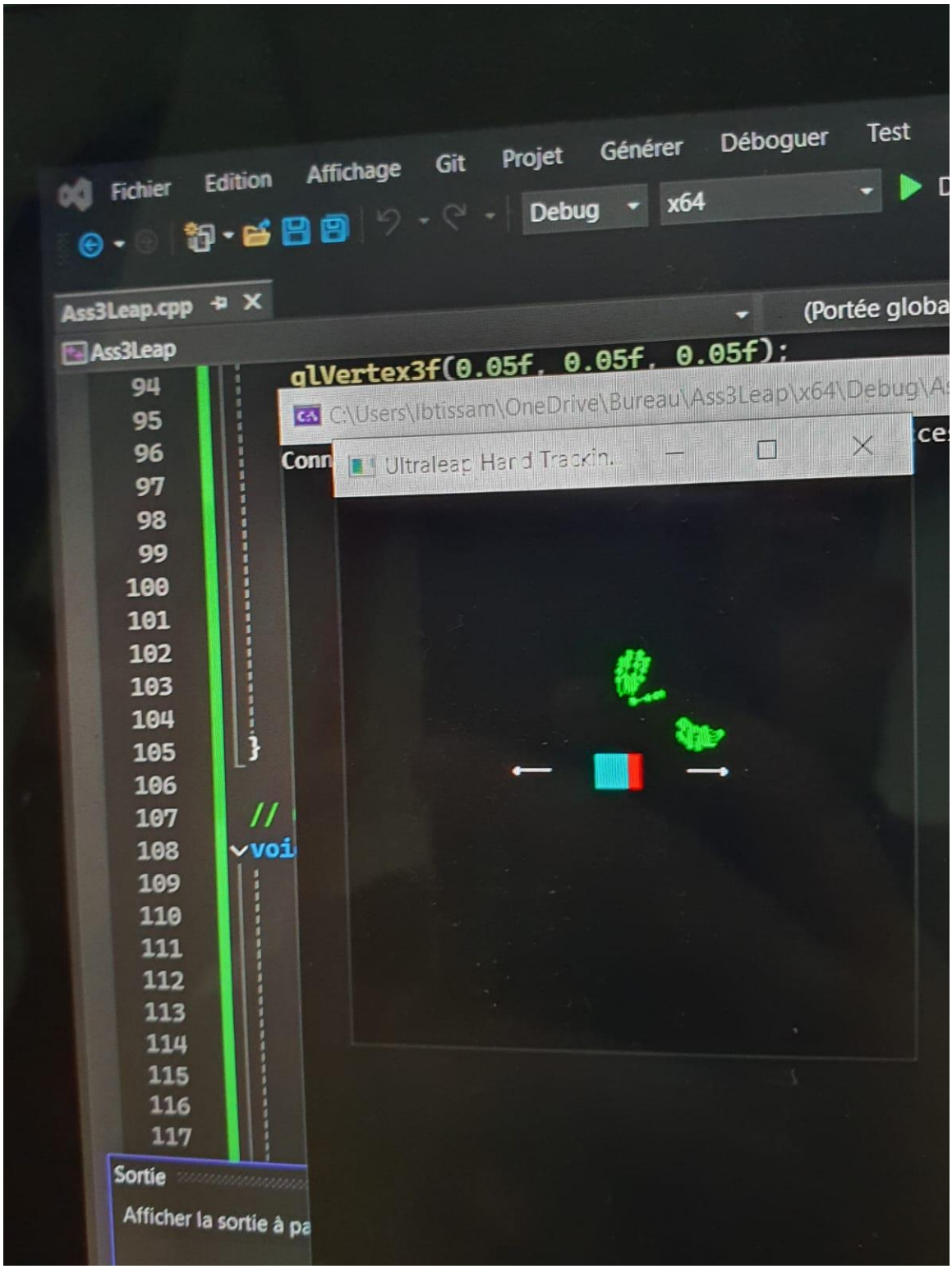

Fichier
Edition
Affichage
Git
Projet
Générer
Déboguer
Test
Debug
x64
Ass3Leap.cpp
Ass3Leap
(Portée glob
glVertex3f(0.05f, 0.05f, 0.05f);
C:\Users\Ibtissam\OneDrive\Bureau\Ass3Leap\x64\Debug\A
Ultraleap Hand Trackin...
Sortie
Afficher la sortie à pa


Fichier
Edition
Affichage
Git
Projet
Générer
Déboguer
Test
Debug
x64
Ass3Leap.cpp
Ass3Leap
(Portée global
glVertex3f(0.05f, 0.05f, 0.05f);
C:\Users\Ibtissam\OneDrive\Bureau\Ass3Leap\x64\Debug\Ass
Ultraleap Hand Trackin...
Sortie
Afficher la sortie à pa


Fichier
Edition
Affichage
Git
Projet
Générer
Déboguer
Test
Debug
x64
Ass3Leap.cpp
Ass3Leap
(Portée globa
glVertex3f(0.05f, 0.05f, 0.05f);
C:\Users\Ibtissam\OneDrive\Bureau\Ass3Leap\x64\Debug\As
Ultraleap Hand Trackin...
Sortie
Afficher la sortie à pa


Fichier
Edition
Affichage
Git
Projet
Générer
Déboguer
Debug
x64
Ass3Leap.cpp
Ass3Leap
glVertex3f(0.05f, 0.05f, 0.05f);
C:\Users\Ibtissam\OneDrive\Bureau\Ass3Leap\x64\Debu
Ultraleap Hand Trackin...
Sortie
Afficher la sortie à pa


Fichier
Edition
Affichage
Git
Projet
Générer
Déboguer
Debug
x64
Ass3Leap.cpp
Ass3Leap
(Portée
glVertex3f(0.05f, 0.05f, 0.05f);
C:\Users\Ibtissam\OneDrive\Bureau\Ass3Leap\x64\Deb
Conn
Ultraleap Hand Trackin...
Sortie
Afficher la sortie à pa


Fichier
Edition
Affichage
Git
Projet
Générer
Déboguer
Test
Debug
x64
Ass3Leap.cpp
Ass3Leap
(Portée globa
glVertex3f(0.05f, 0.05f, 0.05f);
C:\Users\Ibtissam\OneDrive\Bureau\Ass3Leap\x64\Debug\A
Ultraleap Hand Tracking
Sortie
Afficher la sortie à pa

Conclusion

This project successfully demonstrates the integration of Leap Motion hand tracking with OpenGL for real-time interaction with a 3D object. The cube rotates based on the user's hand gestures, and visual indicators (arrows) show the direction of rotation. This setup can be extended for more complex interactions and visualizations using Leap Motion and OpenGL.