**Internet Of Things Project 1**

Group; Email;

Piotr Grzywacz S00253292@atu.ie

Alex McCullough Horan S00254980@atu.ie

Chris Lavin S00251319@atu.ie

Dara O'Rourke [S00252121@atu.ie](mailto:S00252121@atu.ie)

Github link: https://github.com/PineappleGinge/IOT

Trello link: https://trello.com/b/vCb5H5WH/iot-project

**What are we trying to solve?**

Not getting caught cheating in games.

**What does this mean.**

Most games have a traceable software that allows them to detect when a person is using some sort of hack. What we want to do I make it untraceable and that is possible with the help of an external source like an Arduino.

We made a working external cheat source that allows anyone to cheat in First Person Shooter Games. E.g. Valorant, Fortnite and Call of Duty.

**How does it work.**

It works by plugging your mouse into the Arduino and then plugging the Arduino into the laptop. After that when you load into the desired game the mouse would automatically move towards the heads of targets.

**How does the mouse move.**

This was done by uploading many pictures of the game where the head of players were visible. Though the help with coding the mouse using C-Sharp with the pictures the mouse in game moves towards

**Requirements**.

1. Has at least one external component/sensor
2. Can work on multiple games.
3. [x] number of lines of code used.

**Research:**

For our research we investigated what it means to use aimbot and what it is. We found that it moves the screen towards the player so you can shoot it without having to aim or when you fire a bullet the bullet with automatically fly towards the closet player in that facility.

We went searching for an aimbot so we could see how it works with our own eyes so we could replicate that and make it untraceable. After some time, we found a suitable aimbot to try out and collect information from it.

While also looking for an aimbot, we were also watching YouTube videos about aimbot.

Out of all the videos we watched to help understand how it works,

<https://www.youtube.com/watch?v=FM1E0ucndFw>

This video here made it possible to truly understand how it works.

**Summary of how it works:**

For 2D game it is only on one plane and therefore only uses one angle. E.g. (x,y) and the angle between them but for a 3D game is uses the planes of (x,y,z) and two angles.

What It means when using the angles is the position of the player and how they view the game in their view, higher elevation, lower elevation etc.

Using this explanation we researched we were able to make our own aimbot.

**How is it untraceable.**

By using an Arduino Leonardo, USB Host Shield, a mouse, and AI, we can modify / create genuine mouse inputs at a hardware level, with information regarding object positions on the screen provided by our AI model (Screen is simply streamed to the AI which looks like a simple screen recorder to the rest of the system). This fully eliminates the need for memory scanning which also makes our cheat unpatchable.

**Testing.**

For testing our project, we conducted tests all towards the number of pictures needed for the mouse to move towards the player.

We uploaded a set number of pictures and try it out in game to see if it works as intended.

**Test 1:**

The first test was with 100 pictures, this showed us that it did not work with only 100 pictures.

We concluded that the device couldn’t process the outline of the player through the limited number of pictures.

What we had to do was upload more pictures and make them clearer so the device can analyse them.

**Test 2:**

For test 2 we uploaded up to 500 pictures. We felt like this was a good number of pictures to add to see how it reacts.

After testing it in the game we found that the mouse moves towards the player at a good rate with no signs of it slowing down or not reacting fast enough, but it felt like it could still be improved.

**Test 3:**

This is the final test we did, and we decided to upload about 1000 pictures of the player and increase the quality of the pictures. We did this because we felt like it could improve the response time and make the mouse move towards the player with even a small portion of the body being shown.

After we tested it in the game, we found that it was more accurate than the previous tests. This means that the more pictures you add to the system the more accurate it can be and better response time.

**Security measures**

We encode the serial between our C-Sharp application and the Arduino to make it 100% untraceable, as well as giving the Arduino a genuine mouse PID so it looks exactly like a mouse to the system.

**Sketch and sample code**

A white paper with writing on it

Description automatically generated

**Code sample**

**void MouseRptParser::OnMouseMove(MOUSEINFO \*mi)**

**{**

**dx = mi->dX;**

**dy = mi->dY;**

**};**

**void MouseRptParser::OnLeftButtonUp (MOUSEINFO \*mi)**

**{**

**lmb = 0;**

**};**

**void MouseRptParser::OnLeftButtonDown (MOUSEINFO \*mi)**

**{**

**lmb = 1;**

**};**

**void MouseRptParser::OnRightButtonUp (MOUSEINFO \*mi)**

**{**

**rmb = 0;**

**};**

**void MouseRptParser::OnRightButtonDown (MOUSEINFO \*mi)**

**{**

**rmb = 1;**

**};**

**void MouseRptParser::OnMiddleButtonUp (MOUSEINFO \*mi)**

**{**

**mmb = 0;**

**};**

**void MouseRptParser::OnMiddleButtonDown (MOUSEINFO \*mi)**

**{**

**mmb = 1;**

**};**

**MouseRptParser Prs;**

This is part of the code for the project and its what allows us to map out the mouse buttons and register them within the code for it to work.

**Future Development**

For future development we would like to design a case to protect all of the components and to also make the our AI model faster for improved accuracy of our aimbot.