

Bank Cam

By: Ciarán Quinn & Davy Ryan

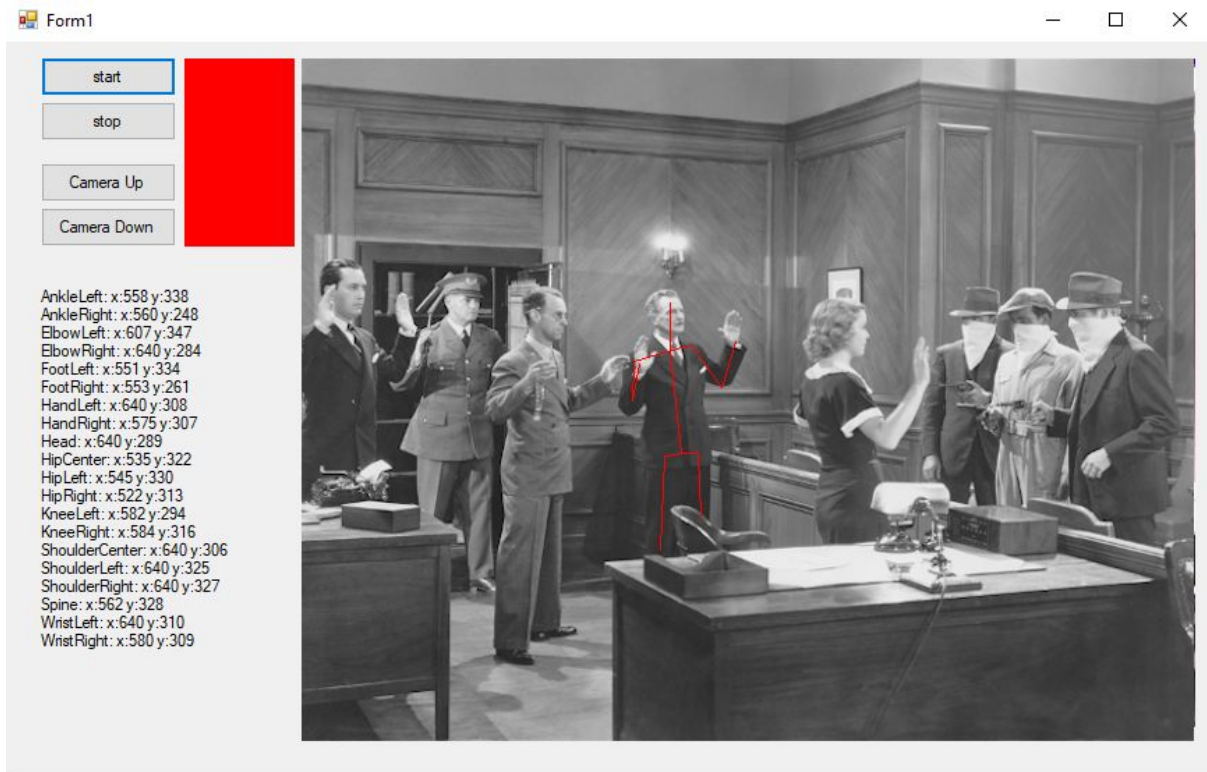


Table of Content

Introduction:	2
System Requirements For BankCam :	3
Technology Used:	3
Architecture of the solution:	4
Features of the Implementation:	6
Limitations:	7
Conclusion:	9

Introduction:

At the outset the idea for this project was to develop a robust and accessible security system that would be easy to acquire , install and maintain allowing this system to tap both commercial and civil markets.

The current build is specifically designed for use by banks , credit unions, bookies or other institutions that are liable to be assaulted or burglarized .

For this project accessibility was a key factor as we wanted to ensure that this system was open to all people be they corporations , civil , or everyman. With this remit in mind we set out to acquire hardware that we found to suit our needs and after extensive research we decided on a Kinect for Xbox360.

This was chosen for its low cost and ample online documentations and resources which in turn allowed for a smooth development cycle.

We did intensive research into the correct language to use for this undertaking as it was by far the most integral part of this project. We chose to do this project in c# because it is fast , lightweight and compatible with the hardware we were using . We also decided to use c# because a common trend in large scale security systems is robustness which is why many

institutions still use outdated software and c# integrates easily with these systems.

System Requirements For BankCam :

- Windows 7,8,10,
Windows Embedded Standard 7,
or Windows EmbeddedDOSReady 7.
- 32 bit (x86) or 64 bit (x64) processor
- Dual-core 2.66-GHz or faster processor
- Dedicated USB 2.0 bus
- 2 GB RAM
- Xbox 360 Kinect for windows

Technology Used:

- Windows forms.
- Xbox Kinect v1.7 SDK.

Architecture of the solution:

The program needs a way for the client to start and stop the camera

They need to be able to adjust the camera

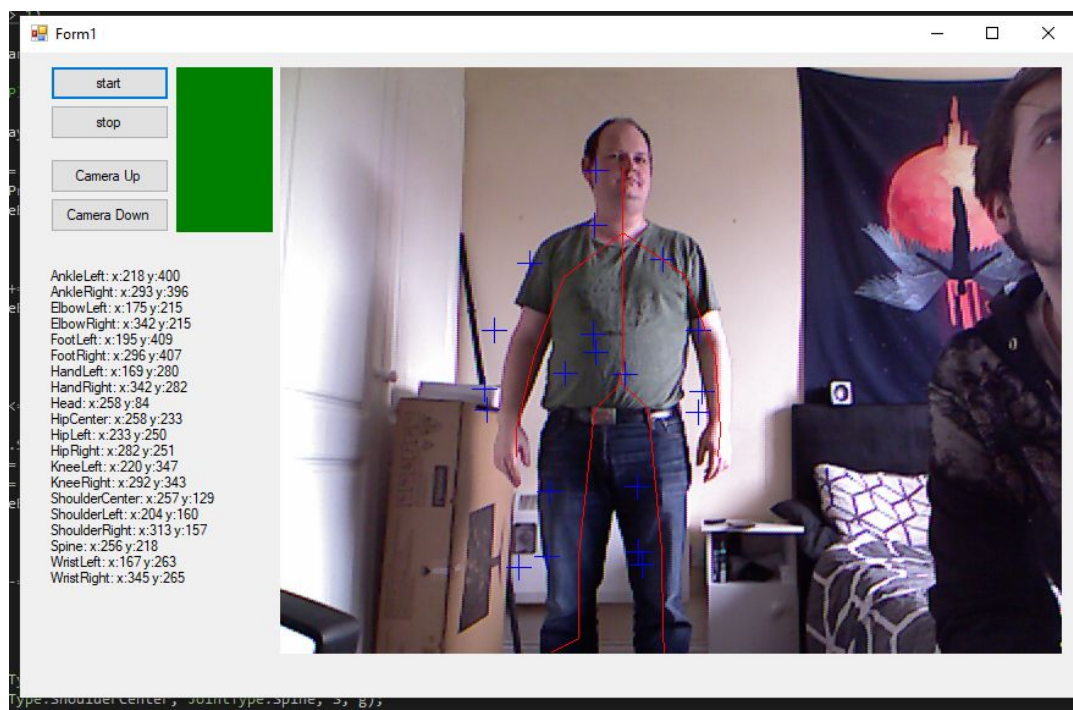
It needs to be obvious at a glance what state the alarm is in

The program needs a view from the camera so the client can set it up properly and confirm it's working

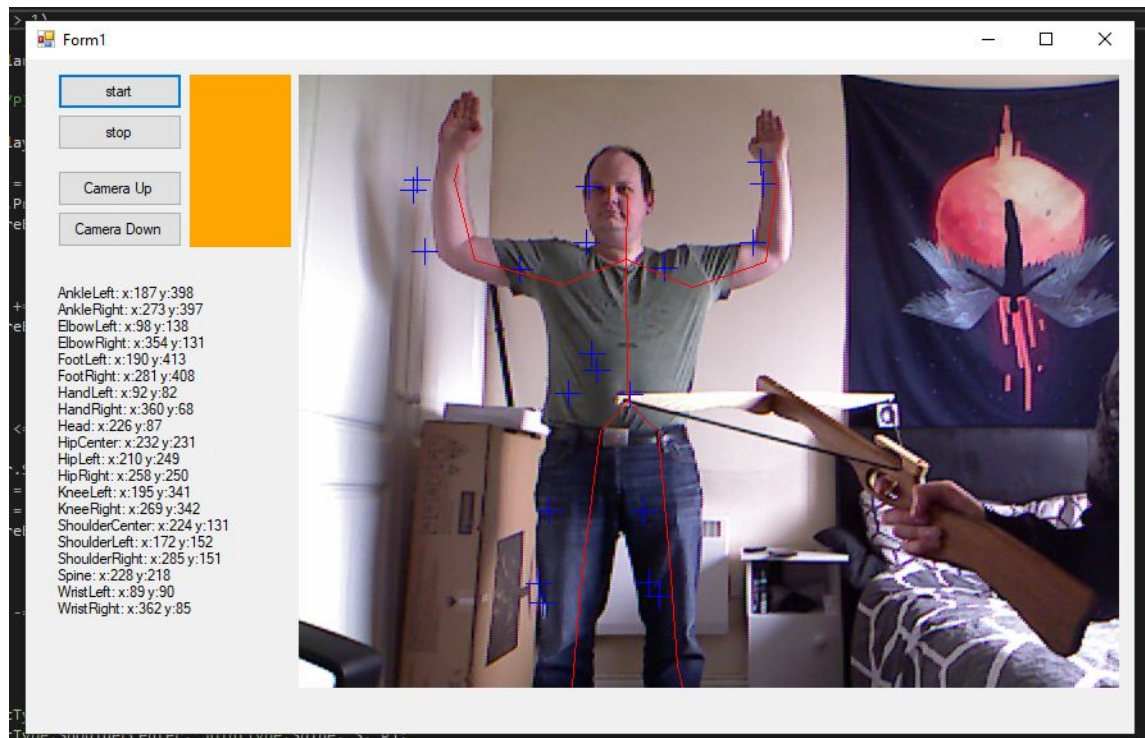
When the program is monitoring someone it shows their skeleton as red lines and the raw data for the joints as blue crosses

The state of the alarm is shown as a coloured box next to the camera

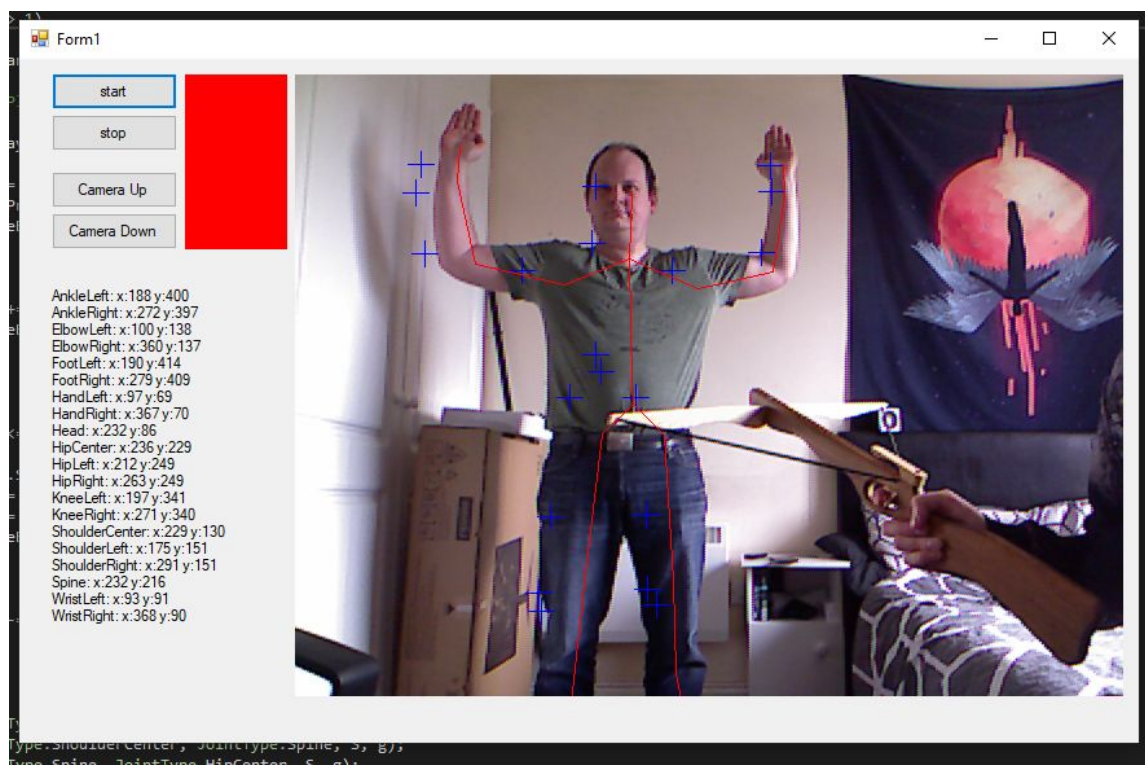
Green for safe



Yellow when the system detects someone raising their hands



Red when someone has their hands up for more than three seconds



Features of the Implementation:

- Backwards compatibility with a majority of widespread systems.
- Low initial , running, and maintenance cost.
- Intuitive interface and Enhanced user-Friendliness.
- The use of c# means this software can be run by even the most rudimentary system.
- Set-up-and-forget. It will constantly run in the background and should only be noticable when triggered.
- The system can be adapted to work with most cameras that boast the same features as the xbox360 Kinect.

Limitations:

- Currently has a limited library of recognised suspicious actions and body positions.
- Requires a specialized camera with IR , Motion Tracking, and Depth sensors built-in although it is relatively simple to acquire a camera with these capabilities.
- Can have issues when used with large crowds or in foggy/misty conditions where conditions are detrimental to vision or may interrupt the camera.
- Select software is proprietary to the Microsoft Corporation although these are free to use regardless.

Recommendations for Future Development:

- Integration with Azure Kinect for when it's released.
Azure kinect recognises fine movements such as fingers and facial recognition, this could be used to alert if someone is planning a robbery(scoping out the bank).
- Machine learning can be used to detect the reactions to criminal activity meaning it can be used to detect crimes contextually through tracking of civilians or other innocent presences in proximity .
- Recognition of weapons , both fire-arms and edged/pointed/weighted, different body-positions(laying on the ground, hostage situation, etc.)
- Voice control and commands by integrating VoiceAttack or other similar software.

Conclusion:

We set out to create a security system that utilizes skeletal tracking in order to detect suspicious behaviour in banks and institutions that handle cash. We did substantial amounts of research into the use of tracking and detection in modern security systems and we have improved majorly in the use of C#.

We feel this project was a success and hope to continue working on it in the future . The use of this technology in security systems is growing and we have found that this project has equipped us to work with this technology in the future.

This program has an incredible amount of potential, such as prison surveillance and event security. it can be modified to fill a wide variety of roles by using external libraries and extensions.