ECE1778
Final Report

Onome Igharoro
Soorena Merat

April 12, 2011

A.R.I.E.S

Augmented Reality Interactive Entertainment System

Introduction

Objective
The objective of this project is to develop an application to remotely operate a paintball turret from an Android mobile phone device. The application must also display a video stream from a wireless camera mounted on the paintball turret.

Audience
The intended consumers of the application are paintball game facilities, who can to augment the turret into their existing game modes. Therefore, user-experience (UX) is a critical factor in the design of the application.

However, due to the resource, time and scope limitations of the project, it is not feasible to implement the application with a full-scale paintball turret. Therefore, the application developed in this project will be tested on a small prototype turret; capable of replicating the essential functions a full-scale model (i.e. pan/tilt motion and shooting).

Features
The final application includes the following features:

- Control the paintball turret pan, tilt and shoot functions
- Display real-time video feed from the wireless camera mounted on the turret

Overall Design

Software (Android Application)

Communications

The communications (referred to as 'Comms') module encapsulates all the functions required to interface with the Lego Mindstorms NXT (referred to simply as 'NXT') hardware, utilizing the Bluetooth API included in the Android SDK. Once initialized (with a MAC address), the Comms module creates a Bluetooth socket connection to the NXT device and opens one input and one output stream for communications.

In order to reduce communication latency with the NXT, and maintain the responsiveness of the app, the Comms module runs in the background on a separate thread from the user-interface (UI). This background thread processes and sends messages without blocking the UI thread, and monitors the input stream for responses from the NXT.

The Comms module communicates with the NXT by sending messages in the format specified by the Lego Communications Protocol (LCP) [1], through the output stream on the Bluetooth socket connection. Depending on the message sent, the Comms module receives a response from the NXT through the input stream. It then translates the response and notifies the UI thread.

User-Interface (UI)

The UI consists of the video display, several buttons and text labels, oriented in a fixed-landscape layout. The video display is the dominant component of the UI, and is stretched across the background. The buttons and text labels are overlaid on the video display.

The video display is comprised of a WebView widget from the Android SDK, which accepts the URL of the IP camera web-server. The WebView loads a page from the camera web-server that provides a live-feed from the wireless camera mounted on the paintball turret.

The layout is fixed in the landscape orientation in order to maximize the size of the video display, which has a 4:3 aspect ratio. Fixing the orientation also simplifies the application logic, since a change in orientation triggers the Android OS to destroy and restart the current Activity.

The buttons are aligned along the bottom edge of the screen to minimize obstruction of the video display. The four buttons control the following functions:

- **Shoot** – the shoot button allows the user to control the shoot function of the turret, by notifying the Comms module to send a message to the turret to activate the shooting motor for a short duration (approximately long enough to discharge two paintballs).
- **Connection** – although the application automatically connects to and disconnects from the turret on startup and closing (respectively), the Connect/Disconnect button is still provided because the connection is occasionally compromised and needs to be reset. Thus, the button allows the user to reset the connection without having to close and relaunch the application. A future improvement would be to detect when the connection with the NXT is compromised and prompt the user to reset.
- **Calibration** – the turret tilt motion is controlled by tilting the phone about the z-axis. The calibration button allows the user to set the reference baseline rotation at which the tilt is considered idle. Deviations from the baseline rotation are then measured to determine whether the turret should tilt-up or tilt-down.
- **Formatting** – the format button is required to reposition the focus of the WebView control to the video display portion of the IP camera web-server page. This is provided only as a work-around to avoid having to manually scroll the WebView widget to view the video display. In a future improved version, this can be removed by writing a custom WebViewClient, in which the formatting function can be buried in the page-loaded (onPageFinished) event listener.

The text labels (TextView widgets) provide feedback to the user for the Bluetooth connection status, and the current state of the pan and tilt motions. The feedback is provided to improve the situational awareness (SA) of the user, following the guidelines for good Human-Robot Interaction (HRI) [4]. In the latest version of the application, the feedback is provided as text, but this can be replaced with more intuitive and user-friendly widgets in future versions. For instance, replacing the text labels with a slider with heading markers (for panning), and level indicators (for tilt).

Hardware (Turret & Camera)

Turret Controller

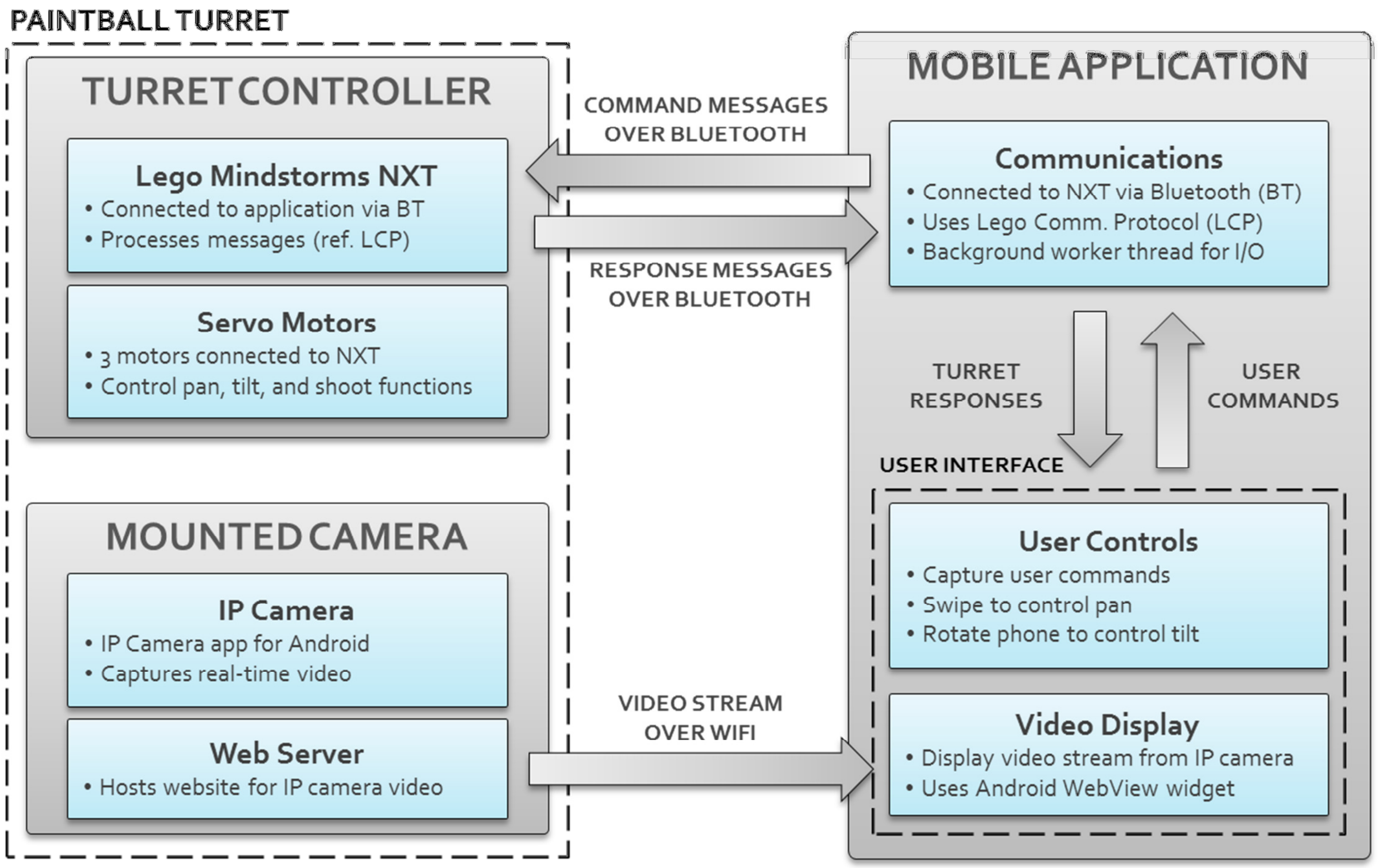
The turret controller consists of the NXT microcontroller unit and three servo motors (all connected to the NXT microcontroller). The NXT microcontroller is a battery operated unit that powers and controls the servo motors. It is paired with the phone offline for Bluetooth communication.

Wireless Camera

The wireless camera consists of a second Android phone running the IP Webcam app (downloaded from the Android Market) [2]. The camera-phone is mounted at the front of the turret, providing a view of the turret sight.

Block Diagram

Below is a block diagram outlining the software and hardware functions and interactions.



PAINTBALL TURRET
TURRET CONTROLLER
Lego Mindstorms NXT
• Connected to application via BT
• Processes messages (ref. LCP)
Servo Motors
• 3 motors connected to NXT
• Control pan, tilt, and shoot functions
MOUNTED CAMERA
IP Camera
• IP Camera app for Android
• Captures real-time video
Web Server
• Hosts website for IP camera video
COMMAND MESSAGES OVER BLUETOOTH
RESPONSE MESSAGES OVER BLUETOOTH
VIDEO STREAM OVER WIFI
MOBILE APPLICATION
Communications
• Connected to NXT via Bluetooth (BT)
• Uses Lego Comm. Protocol (LCP)
• Background worker thread for I/O
TURRET RESPONSES
USER COMMANDS
USER INTERFACE
User Controls
• Capture user commands
• Swipe to control pan
• Rotate phone to control tilt
Video Display
• Display video stream from IP camera
• Uses Android WebView widget

Figure 1 - Application Block Diagram

Successes, Failures and Improvements

Overall, the final version of the application successfully achieved the project objective; it enabled remote control of the paintball turret, and displayed video images from the turret-mounted camera. The specific successes, failures and areas for improvements are discussed in the following sections.

Successes

Remote Control

Earlier versions of the application (as demonstrated in Spiral 2 and Spiral 4 presentations) suffered from severe communications latency, which resulted in poor control of turret. The communication latency was significantly reduced in the final version of the application, by:

- **Multithreading** – in earlier versions, the Comms module was executed on the UI thread, which caused the UI to block with every communication (i.e. connecting, panning, tilting, etc).
- **Reusing I/O streams** – in earlier versions, the output stream was recreated and closed after every command request. Coupled with the fact that the Comms module was executed on the UI thread, this contributed greatly to the latency issues. In the latest version, the input and output stream are created and opened when the Comms module connects to the NXT, and closed when the Comms module disconnects.

Remote Video Display

Using a WebView widget and webcam viewer provided by the web-server created by the IP Webcam app, the remote video display feature was successfully implemented. Unfortunately, since the webcam viewer is web-based (with frames updated by JavaScript), the frame rate of the video stream was relatively poor.

Turret Functionality

The prototype paintball turret successfully emulated the functionality of a full-scale version (see Figure 2), i.e. pan/tilt motion and shooting function.

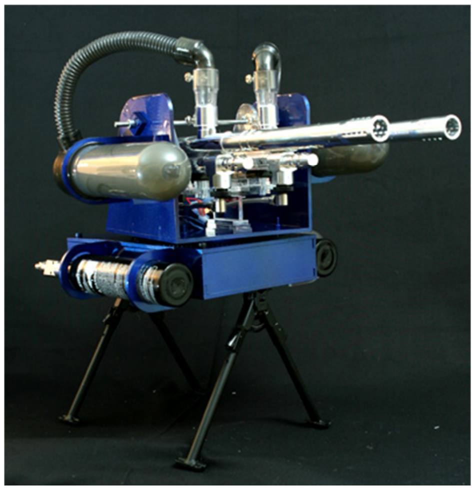


Figure 2 - An Example of a Full-scale Paintball Turret [3]

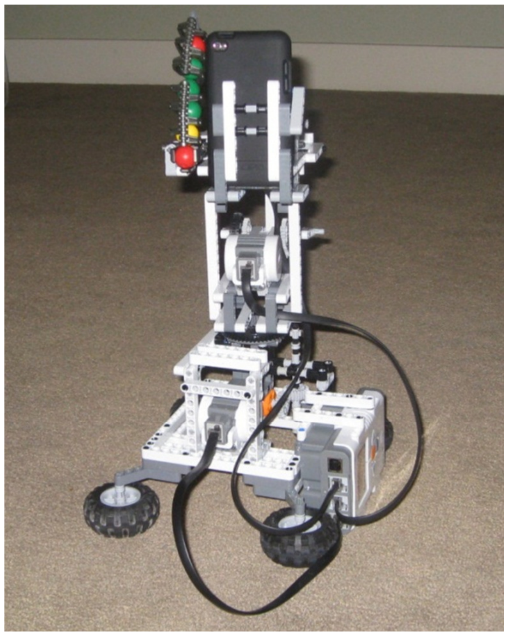


Figure 3 - Prototype Paintball Turret (Final Revision)

Failures

Remote Video Streaming (Direct)

Initially, the video display feature involved utilizing the MediaPlayer (from the Android SDK) class to connect to the IP camera web-server to obtain video data. This approach was motivated by the Android API documentation for the MediaPlayer class functionality. Per the API, the MediaPlayer class is able to playback video streamed from a network source via HTTP or RTSP. This was confirmed by successful implementations of streaming videos (3GP encoded) from a video upload website in early versions of the application.

Ultimately, it was discovered that the IP webcam video used MJPEG encoding, which is not supported by the MediaPlayer class, and the decision to switch to the WebView control was made.

Lego NXT Firmware Replacement

The first version of the application featured the use of the leJOS SDK for communication with the NXT, which required a custom firmware replacement for the NXT microcontroller. Whilst uploading the firmware to the NXT, there was an error that broke the NXT software, resulting in a clicking sound and failure to boot. Unfortunately, there is no known solution for this problem – only a few accounts of miraculous recoveries while connected to numerous PCs. Fortunately, however, the NXT reset while connected to a PC installing the LabView driver.

Improvements

UI Redesign and Polish

As discussed in the User-Interface section, there are several Button and TextView widgets in the UI of the application. The buttons are overlaid on the video display and obstruct the view of the field. Guidelines for good HRI recommend minimizing the obstruction of the video feedback [4]. Therefore, to improve the UI in future versions, the number of buttons can be reduced. Furthermore (as discussed earlier), the text-based feedback can be converted to more user-friendly graphical widgets.

Robust Prototype

The turret prototype turret successfully fulfilled the requirements for testing the application. However, it has several limitations and areas for improvements:

- **Improve tilt mechanism** – the crank mechanism that controls the tilt motion folds if the servo motor that controls it is rotated too far. This often locks the mechanism, as the motor is unable to drive the mechanism to unfold.
- **Improve rigidity** – the modular nature of Lego results in relatively loose connections between pieces which, coupled with the backlash from the gears, leads to occasional loss of control due to gears failing to mesh. This can be mitigated by designing the assembly to minimize loads on the gear-trains and reinforcing areas with high loading.
- **Resolve motion restrictions** – the wires connecting the servo motors to the microcontroller currently restrict the pan/tilt motion of the turret. This can be improved by using longer cables; although that would not solve the problem, it would at least extend the range of motion. A better solution would be to route the cables through the rotation axes.

Lessons Learned

User-Experience

As discussed in the Introduction, the user-experience is a critical factor in the design of this system. Due to the difficulties encountered developing the core functionalities of the application, the UX was not considered until a much later stage than we initially intended. We eventually discovered that the original proposed input method that featured crosshairs to be dragged by the user to pan/tilt the turret (as described in the Project Design Plan) resulted in poor UX. This was largely due to the fact that a significant portion of the video display was obstructed by user's finger as they dragged the crosshairs. Furthermore, the high frequency of dropped commands inherent in the NXT microcontroller made the turret feel unresponsive.

Ultimately, the input method was re-designed to implement a 'swipe-and-tilt' control scheme, where the tilt of the phone about the z-axis controlled the turret tilt motion, and the user swiped across the screen to control the pan motion. The new control scheme greatly improved the UX.

The lesson learned from this was that such a critical design factor should not have been left so late. Had we addressed UX earlier, we may have discovered the limitations of the platform and modified the design accordingly.

Software-Hardware Interface

Interfacing with hardware components from software is always a challenging endeavor. In this project, we intended to utilize the leJOS SDK because of the extensive documentation provided, and available community support.

Unfortunately, since we were unable to proceed with leJOS (following the near destruction of the NXT microcontroller); we defaulted to the Bluetooth Development Kit provided by Lego and, somewhat naively, began coding the communication based on the Lego Communications Protocol (LCP). However, our limited experience with both Android and Lego Bluetooth platforms made it difficult to make progress on the application, as it proved difficult to debug (often the error was simply a lack of response from the turret).

Fortunately, after discovering several leJOS-specific samples online and subsequently reviewing the leJOS source-code, we made some breakthroughs with our implementation and successfully completed the application.

There were two lessons learned from this:

1. Perform more thorough research on SDKs prior to trying them out (particularly ones that have potential to cause significant damage, e.g. leJOS firmware replacement).
2. Perform more thorough research on existing solutions and the approaches taken by existing solutions before designing and working to implement a specific solution.

Spiral Method

Utilizing the spiral method was a highly effective strategy. It allowed us to identify and tackle the major challenges and issues in a relatively organized and manageable fashion. For instance, dividing the implementation of the remote video display into segments (local streaming, streaming from the internet, and attempt at streaming from a device) allowed us sufficient time to familiarize ourselves and build confidence working with the MediaPlayer class in early implementations.

Contributions

Onome Igharoro

- Designed and built all revisions of the application software
- Troubleshooted issues with the prototype paintball turret and suggested revisions
- Co-authored final presentation and report

Soorena Merat

- Designed and built all revisions of the prototype paintball turret
- Designed the application user-interface layout (based on HRI guidelines)

- Co-authored final presentation and report

Apper Context

To recap the Project Design Plan, the Apper's (Soorena Merat) background is in Mechatronics and Communications, and he is currently pursuing an M.Eng degree in Mechatronics. His research/project interest is in mobile robotic design, and teleoperation of robotic systems.

In this project, the Apper designed and built the prototype paintball turret that was controlled by the application. The Apper also researched and employed HRI guidelines to design the user-interface to maximize the user-friendliness of the application.

The application developed from this project provides a valuable tool for prototyping and developing commercial implementations (e.g. commercially available teleoperated turrets), for the following reasons:

- **Modular** – the communications and UI layers are decoupled, so the UI can be easily redesigned to test alternative concepts.
- **Flexible** – although the application is currently written to function specifically with the Lego Mindstorms NXT, it can be easily adapted to communicate with other Bluetooth-enabled devices.

The prototype paintball turret constructed provided an effective tool to test and evaluate the UX and user-friendliness of the application. Earlier revisions (before Spiral-4) were largely unfriendly, as the lag in communication and high frequency of missed messages made the turret highly unresponsive and difficult to control.

Improvements made on the application software (Spiral-4) and hardware (increasing the gear-ratio to slow down the tilt) improved the UX by decreasing the lag in communication, and effects of lag on the turret motion. However, the greatest improvement to UX was achieved by the modifications made in the final application. The further reduction in lag and switch to the swipe-and-tilt control greatly improved the user-friendliness of the application.

In conclusion, the project provided a practical implementation of good HRI practices in design. The application developed can be the basis for further work examining the effectiveness of various control schemes for teleoperation of turret-mounted robotic systems.

Future Work

As discussed in the Introduction, the target audience for this application would be paintball game facilities. Given the size of the paintball industry (considering the relatively large number of peripheral hardware suppliers), a system such as A.R.I.E.S has potential for commercial success. However, the current application requires significant improvements to the UI and performance, as well as testing with full-scale devices before it can be commercialized.

With more time and resources, there are several areas the application can benefit from enhancement for commercialization:

- **Full-Scale Hardware** – test the application on a full-scale paintball turret (see Figure 2). This would eliminate some of the limitations brought on by the NXT hardware.
- **QR-Codes** – implement the QR codes as IDs to allow players to 'capture' (i.e. take control of) a given turret by snapping the QR code mounted on the turret. This would add the last crucial element required for use of the platform in competitive gameplay.
- **Expand Hardware Scope** – increase the scope of compatible hardware devices to enhance gameplay potential. For instance allow players to take control of trap doors, spy cameras, traps, etc.

References

1. Lego Mindstorms NXT – Bluetooth Developer Kit:
    a. Appendix 1: LEGO MINDSTORMS NXT Communication protocol
    b. Appendix 2: LEGO MINDSTORMS NXT Direct commands
2. IP Webcam App – Android Market: http://www.appbrain.com/app/ip-webcam/com.pas.webcam
3. Image from InventGeek.com – http://www.inventgeek.com/Projects/paintball-turret/OverView.aspx
4. Yanco, H.A., Drury J.L., 2007, "Rescuing Interfaces: A Multi-Year Study of Human-Robot Interaction at the AAAI Robot Rescue Competition", Special Issue on the Science Behind Embodied AI: The Robots of the AAAI Competition and Exhibition, p333-352.

Word Count: 2,481