# POOL SHARK TECHNOLOGIES

*Simon Fraser University*
*8888 University Drive*
*Burnaby BC Canada*
*V5A 1S6          www.sfu.ca/~shark*

December 18, 2000

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6


Re: ENSC 340 Process Report for the Ace Training System

Dear Dr. Rawicz:

The attached document, Process Report for the Ace Training System (ATS), outlines the process with which we followed in designing and completing our Ensc 340 project. Our goal is to design and develop a user-friendly pool table that teaches the players how to hit the balls into the pocket.

This document details the current state of the device, our deviations from the design specification, and our future plans for the device. In addition, this document outlines some of the budgetary and time constraints we encountered and explain the inter-personal and technical experience gained from working on the project.

Pool Shark Technologies consists of five talented and ambitious 4th engineering students: Desmond Cheung, Humphrey Ng, Patrick Pun, Janice Wong, and Lawrence Wong. If you have any questions or concerns, please feel free to contact Humphrey Ng by phone at 431-6333 or by email at hngc@sfu.ca.


Sincerely,


Janice Wong
Team Leader
Pool Shark Technologies



Enclosure: ENSC 340 Process Report for the Ace Training System

# Process Report for
# Ace Training System

*Team:*  **Desmond Cheung**
**Humphrey Ng**
**Patrick Pun**
**Janice Wong**
**Lawrence Wong**

*Contact:*  **Humphrey Ng**
**hngc@sfu.ca**

*Submitted to:*  **Dr. Andrew Rawicz – ENSC 340**
**Steve Whitmore – ENSC 305**
**Jason Rothe – ENSC 340 TA**
**James Balfour – ENSC 340 TA**
**School of Engineering Science**
**Simon Fraser University**

*Issued date:*  **December 18, 2000**

*Revision:*  **1.0**

**POOL SHARK TECHNOLOGIES**
*Simon Fraser University*
*8888 University Drive*
*Burnaby BC Canada*
*V5A 1S6          www.sfu.ca/~shark*

# Table of Contents

# 1   Introduction

Pool Shark Technologies is pleased to announce the completion of our first prototype for the Ace Training System (ATS).  The ATS is a user-friendly snooker coach that teaches the players how to hit the snooker balls into the pocket.  Through the stages of analysis, design, implementation and testing, we are able to complete the ATS prototype with minimum modifications.  This document re-examines the process of the ATS development and briefly describes the experiences of each of the team member.

# 2  Current State of Design

As described in previous project documentation, the purpose of the ATS is to coach the user to become more skillful in all aspects of snooker play.  Our first product, the ATS prototype (henceforth referred to as ATS), aims to develop the foundation of the full-scale ATS.  The ATS consists of three major subsystems, the Image Processing System, the Shot Selector, and the User Interface.  Figure 1 illustrates the relationship between them.  After the user hits the shot, the Image Processing System captures a digital image of the table and reports to the Shot Selector and the User Interface the location of all the balls.  The User Interface then displays the graphical version of the ball arrangement to the user, and then accepts the user input on which ball to hit.  From this input, the Shot Selector analyzes the best possible shot to pocket this ball, and sends this information to the User Interface.  This information will then be displayed to screen, as well as to the LED display that surrounds the edges of the table.  All the user has to do at this point is hit the cue ball in the direction of the illuminated LED to pocket the ball!
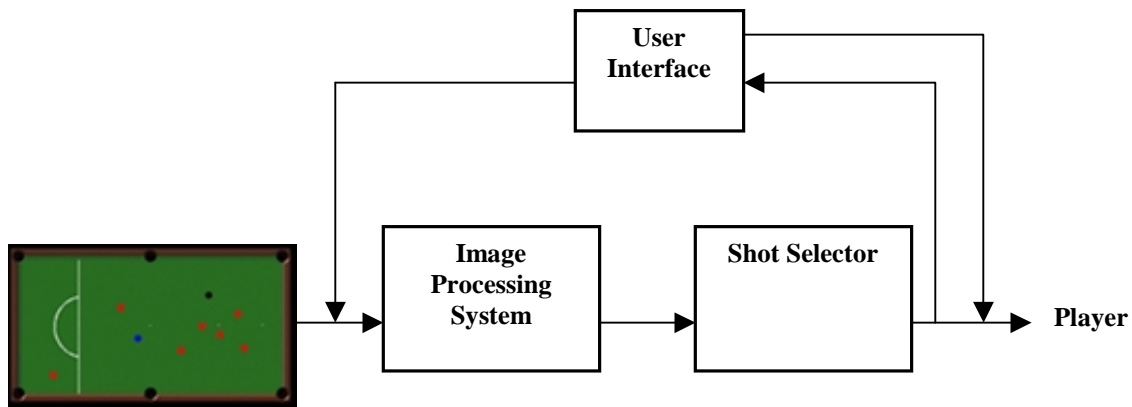


**Figure 1: System Block Diagram**

The physical arrangement for the ATS prototype consists of a single corner pocket and three balls in play- the white, black, and red balls.  Other modifications to the system will be discussed in subsequent sections.

## 2.1  Image Processing System

The Image Processing System consists of the Image Acquisition Unit and the Image Decoder Unit.

### 2.1.1  Image Acquisition Unit

The Image Acquisition Unit is responsible for controlling the actions the Kodak DC290 Digital Camera.  The Software Development Kit from Kodak was used to implement this unit.  Simple functions were written to provide an interface for the Image Decoder Unit.  These functions include:

- SetupCamera - opens the camera and setups all the camera parameters
- TakePicture - takes a picture of the table
- RetrievePicture - retrieves picture taken from camera

- CloseCamera - closes camera

The software in the Image Acquisition Unit configures the camera to store the image file in JPEG format, which is quite precise in its representation of the real image, and is also relatively small in size compared to TIFF and GIF files. This file is then streamed to the computer and saved in the hard drive, awaiting analysis by the Image Decoder Unit.

### 2.1.2   Image Decoder Unit

The Image Decoder Unit is composed of image processing functions provided in Victor Image Processing Library[1]. The library contains several fundamental functions that are useful for simplifying our program in processing the image of the table. They are used to retrieve basic image information and make modifications and manipulations to the image properties to provide the best image before detecting the ball locations. The specific functions that we use include:

- allocimage - allocate space for an image in global memory
- freeimage - free an allocated image
- loadtif - load TIFF file
- convertrgbtopal - RGB image to 8-bit palette color image (Color reduction)
- threshold - set minimum brightness level
- getpixelcolor - read pixel value at buffer (x,y)

The first three functions help loading the image into the memory. The process of distinguishing the balls from the table requires color reduction before applying a threshold level to the image based on the ball colors and brightness. Our functions, *decode_loc()* and *decode_col()*, will scan the image horizontally and output the ball locations.

## 2.2  Shot Selector

The Shot Selector function is responsible for surveying the situation on the pool table and deciding the direction to hit the appropriate target ball. The Shot Selector function can be divided into three sub-functions: Shot History, Shots Calculation, and Best Shot Evaluation functions. The input of the function is the User entry and the output is the value of LED that the user should aim at.

### 2.2.1   Shot History

The Shot History sub-function determines the colour of the target ball depending on the prior shot. When the previous target ball is red, then the next target ball that the program considers will be a non-red ball (i.e. black). In our prototype there are only 2 possible targets, black and red, and therefore the algorithm is simplified. The Shot History function will be bypassed if it detects the UI had already specified a particular ball for the next target.

### 2.2.2   Shots Calculation

The Shots Calculation function determines all the possible vectors to shoot the target ball, whose colour is determined by Shot History, into the pocket. This function uses the following algorithm that has been modified from our design spec:

---

[1] Victor Image Processing Library web site: http://www.webcom.com/victor/download/dwnload.html

1. Determine the closest pocket to hit into by calculating the minimum distance between the target ball and the six pockets;
2. Generate a vector between the cue ball and each LED location;
3. Test if the vector will hit the target ball and if the final vector, the vector after collision, will result into the target pocket;
4. Ensure the test vector and the final vector are not blocked by any other non-target balls.

### 2.2.3  Best Shot Evaluator

The Best Shot Evaluator determines the most appropriate vector to hit the target.  If the vector from Shots Calculation function can hit the target into the pocket without blockage of non-target balls, then this vector is automatically the best shot and will be outputted to User Interface.  In cases when the vector can hit the target but cannot be hit into the pocket, then the best shot will be a safety shot, which touches the target ball without scratching.  When there are no vector that can touch the target ball, the function will output "Out of Range".

## 2.3  User Interface

The User Interface is a combination of two modules.  The Monitor Display Unit doubles as the main program, calling the other subsystems at the appropriate time and managing user input and output.  The LED Display is the second part of the User Interface.

### 2.3.1  Monitor Display Unit

The monitor display unit is a graphical user interface that allows users to see the results of the Image Decoder Unit and Shot Selector.  After the image processing program has detected the ball colors and locations, the user interface program displays balls on the snooker table along with the best shot indicator (LED) around the table.  The LED number is also displayed beside the table to indicates exactly which LED should light up.  This process is repeated when the user hits the "Update" button.  Also, the user can choose a specific ball to hit.  This parameter is then passed to the shot selector and the new best shot indicator will be displayed.

Another important responsibility of the Monitor Display Unit is that it acts as the main program and coordinates other subsystems in the ATS.  This is done by calling executables of the other subsystems at the appropriate times.  Necessary parameters are also passed when the calling the executables.

### 2.3.2  LED Display

The LED Display is implemented with Lumex 67-1045ND rectangular LEDs.  This array of LEDs is controlled by an array of Texas Instrument SN74154N 4-to-16 line decoders.  Figure 2 illustrates how this array was implemented.
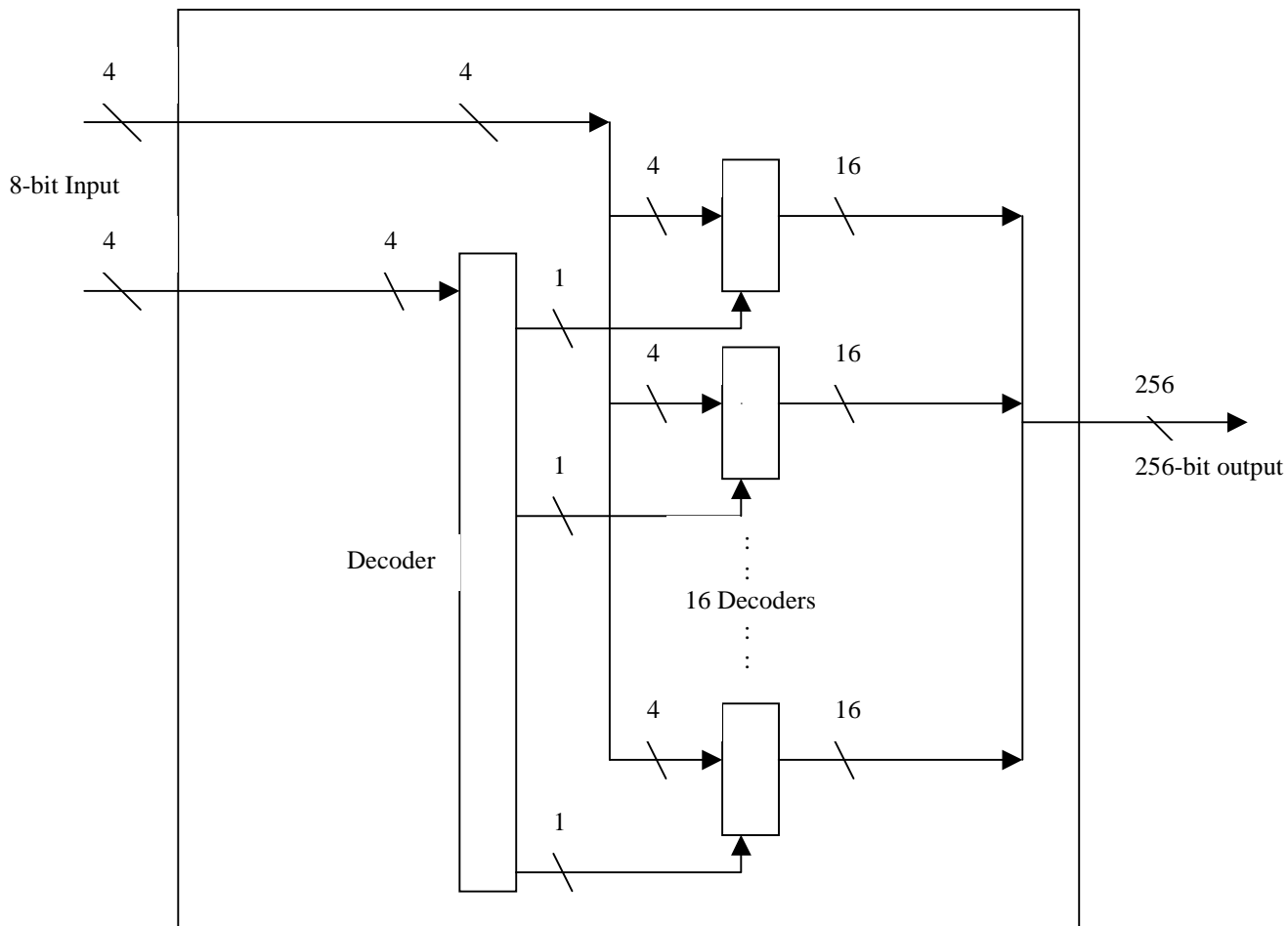
**Figure 2: 8-to-256 Decoder**

Power to the decoder array is supplied by the computer's parallel port. A driver was written for the parallel port that controls specific output bytes at the port address of 378 and 37A. In total we used 11 of the parallel port's 25 pins to control the decoder array. When activated, the pins of the parallel port supply a voltage of +5V.

Since only one output is activated at a time, the current sinking into the decoder is small. We further reduced it by connecting a 300Ω resistor in series with the LED.

# 3   System Modifications

## 3.1   Overall System

The most significant change from the original proposal is that from our prototype we only incorporate a section of the table, and only two balls (red and black).  This simplifies the circuitry for implementing the LED indicator displays, while enables us to focus on the implementation of the important functions.

We have also changed our product to a software package instead of a "system in a box".  This decision is made due to both market need as well as our technical deficiency in choosing the best hardware to use.  Originally we were to implement the system using a specialized microprocessor, however because of our relatively large image processing and shot selection algorithm size, we realized that utilizing an existing computer system will be the platform which yields the most profit and can attract the largest market.

The following details the deviation to the original proposal for the subsystems.

## 3.2   Image Processing Unit

### 3.2.1   Image Acquisition Unit

The Image Acquisition Unit was implemented as proposed.

### 3.2.2   Image Decoder Unit

The deviation of the Image Decoder Unit is that only three target balls need to be detected, which very much improves the system response without changing the algorithm of the original functions.  The improved response is from the reduced time spent on detecting the ball center locations.  Once the function detects the ball surface, it will jump into a loop for calculating the location of the ball's center.  Thus, the algorithm does not require any modifications when full set of ball is used.

The Image Decoder Unit contains the following functions that use the helper functions in Victor Library aforementioned:

- decode_loc – output the locations of the balls on the table.
- decode_col – output the ball colors based on the locations retrieved from *decode_loc*.

## 3.3   Shot Selector

The main deviation of the Shot Selector is that it implements only two target balls.  Since the change did not alter the algorithm of the function, we are able to focus on one ball and decide the most appropriate shot to make.

We also will need an algorithm to determine the choice of the ball to hit according to the points that each ball scores.  Thus black is the top priority, and pink is the second priority, etc.

Bank shot was not implemented because at the time of programming, there are more urgent needs for the other areas of the project.  Implementing the bank shots will be a logic that detects when the ball hit the cushion of the table, the resultant angle will be symmetrical to the normal of the contact point.

Our first intention for this Shot Selector function was to import a developed pool program which can determine the best vector to hit more accurately. However, it was very difficult to look through other developer's source code and determining which parts of the code are useable. Due to time constraint, we finally decide to implement a simpler, although sometimes buggy, code.

## 3.4  User Interface

### 3.4.1  Monitor Display Unit

As described in the project documentation, the user has the capability to handle the full set of balls. However, for our prototype, we have slightly modified and commented out part of our program so that it takes care of just three balls. For handling all the subsystems in the ATS, we have once considered using dynamic linked libraries to call other subprograms. However, considering both complexity and speed, we have decided that calling executables is the most appropriate solution for our purposes. Other parts of the program are indicated exactly as what was stated in the design specifications.

### 3.4.2  LED Display

By deciding to limit the area of coverage by the ATS to the corner pocket, the number of LEDs needed drastically reduced. In truth, if we followed our ambition to cover the entire table, the notion of using LEDs is not very realistic. By covering one pocket, we scaled the project down so that it is still feasible to control and implement the LED array.

# 4   Budget and Time Analysis

## 4.1  Budget

Upon our decision to make the project as software package, our cost of the materials drastically decreases. We have also found that one of our group members, Desmond, has a digital camera that can be used for this project.  Therefore we were able to cut down the cost significantly.

**Table 1: Budget Comparison**

| Equipment Name | Proposed Cost | Actual Cost |
|---|---|---|
| Processor | $30 | N/A |
| Evaluation board | $300 | N/A |
| RAM | $30 | N/A |
| Camera | $400 | Own |
| Hardware (Decoders, resistors, LEDs) | N/A | $110 |
| Pool table (2$^{nd}$ hand) | $200 | $50 |
| Balls and cues | $50 | Free with table |
| Miscellaneous (AC adapter, switch cable, wires and other) | $150 | $70 |
| **Total** | **$1160** | **$230** |

## 4.2  Time

We were able to meet the schedule that has been set for the documents.  However, our internal schedule was very hard to keep up because each of us has lots of laboratory work and assignments for other courses. There was not enough pressure to finish off our project ahead of schedule that maybe over-confidently set during our proposal.  We have had some unexpected technical difficulties, which also delay our progress.

# 5  Future Plans

### 5.1.1  Overall System

For the next phase of the ATS, we need to incorporate the full set of balls.  This is not difficult to do because when we were implementing the system, we structured the separates functions such that a full set of balls can be used.

We can also implement the ATS as a "system in a box".  We realized because the buyers for the ATS are mainly home users and pool hall/pub owners, thus although implementing a PC software package for the home users is a good idea (due to the highly availability of PCs in homes), it may not be as easy to sell the same system to pool halls and pubs.  We need to implement the system using a specialized microprocessor, enough disk space and other system requirements, so that it can be used in pool halls and pubs.

Because in North America Pool is much more popular than snooker, therefore we should modify our system to Pool.  This can be done by modifying the Shot Selector function to pool, and by modifying some image processing algorithm to take care of the stripe and solid balls.

Although the speed and accuracy for our current system satisfied what we designed for, we should improve on these parameters in the future.

The following are the suggested improvements for different components:

### 5.1.2  Camera

Although the unit was successfully implemented, there were few other problems that we have encountered. The quality of the image is very important to image processing.  Therefore, we had to try different configurations on the camera, and different light settings to produce the best image easiest for the Image Decoder Unit to recognize the balls on the table.

### 5.1.3  Image Processing

In our Image Acquisition Unit, we realized the constraint that image has to be taken, transferred, and processed in every play.  This slows down the whole system and users have to wait before proceeding to the next play.  In order to improve this drawback, we could use a digital video camera instead of a digital camera.  This provides a real-time processing capability and everything can be processed faster.  In addition, the system can read every single play in motion.   More features can then be implemented to the Ace Training System such as correcting players aiming skills.  However, this increases the complexity of the system in terms of both hardware and software.  Other than a digital video camera, a real time processing graphics board, more sophisticated read time software are required.

### 5.1.4  Shot Selector

The shot selector we have implemented in our system is very primitive because we did not have enough time to research on the physics of a real pool table.  To improve on it, we can implement this section using existing software, such as porting the Virtual Pool game developed by Electronic Arts, and incorporate the software with our system.  The more realistic shot selector results in a much more accurate shots!

### 5.1.5   Monitor Display Unit

Currently our monitor display unit is implemented on a computer.  It would be more convenient to have the graphical user interface implemented on a wireless device so that the user can choose ball to hit and do all the screen updates right at his/her spot.  However, this is much more complicated and requires a lot more time and research.

### 5.1.6   LED Display

This is the most obvious modification that we need to make.  While the current scheme of LED arrays works reasonably well under the conditions of the ATS prototype, it definitely is not viable with the full-scale ATS.  The number of LEDs will just be too many, making it impossible to control and too complicated.  A likely version of the display would be implemented with a laser pointer that is mounted on a rectangular frame that is positioned high above the table.  The laser pointer will then be moved across the frame using tiny motors and sensors, and points to the appropriate position on the edge of the table, just like the LED does.  Actually, this was our first idea for the display, but we felt that it was more important to build a strong foundation for the ATS's different components.

# 6  Personal Experience

## 6.1  Desmond Cheung

My main role was implementing the Image Acquisition Unit and interfacing different modules together.   In the technical aspect, I was able to gain more experience in Windows programming.  Most of the lower division computer courses teach programming in terms of algorithms, terminology, or skills.  There was not much Windows programming, which is very important when you are out in the industry.  Moreover, I realized the importance of good programming style and habits.  As there were few modules to be put together, good coding style and comments were crucial when everything was linked together.

From this project course, I was also able to experience the difficulties of team dynamics.  Although I have worked in teams as a team member during my coop terms, everything was managed and organized by the team leader.  By only giving suggestions or taking commands from your boss, there were few opportunities that you can see the problems in terms of team dynamics.  On the other hand, everyone has his/her own schedule, taking different courses, and different learning curves.  It is always difficult to accommodate everyone's wishes.  It is important that we all realize these difficulties and try to understand each other in order to obtain a good team chemistry.

## 6.2  Humphrey Ng

One of the most important things I have learned throughout Ensc 340 is time management.  Because all of us have many other courses that demand lots of our time during the semester, it is very important to set a weekly goal for our project.  This way we can avoid the mad rush we have at the end of the semester.

Technically, I have gained a great deal of confidence in programming C++.  Although I have programmed before in my courses and my work terms, I have mostly been modifying the existing files given by the instructor or supervisor.  This is truly the first time which I have implemented a large algorithm from scratch single handedly.

Besides programming, I was also involved in designing the system and finding the parts suitable for the design, particularly choosing the microprocessor and evaluation board for which our project was intended to be on.  After some investigation, we have decided that from both market and technical point of view, it makes much more sense to package our project as software rather than using expensive microprocessors.  This gives us more flexibility in programming and shifts our focus to implement a working prototype.  Even though the research was not useful in our final decision, I had learned many considerations in choosing an appropriate platform for our project.

Finally from the project I have learned the importance of making trade offs.  When there are many tasks to do, I have learned to prioritize the important ones.  These are experiences which I will make use of in my future career.

## 6.3  Patrick Pun

In my opinion, ENSC 340 has provided me with the opportunity to work more than eight hours after school each day and show up in SFU seven days a week.  It is really something more than I ever have in my entire life, but the experience that I have gained from this course is truly valuable.

From a technical aspect, the coding experience that I had was with image processing. I expended my knowledge of image manipulations and C++ programming. The core difficulty with respect to locating the balls was the quality and brightness of the image. The fact that balls and table have similar colors and brightness caused problems with distinguishing the balls and the table. These problems lead to the necessity of careful scan of the image and complex algorithm, which slow down the response of the system.

Personally, I learned that time management is the most important. We had numerous deadlines broken due to over optimistic time estimations on completing subcomponents because of some unexpected difficulties. Even though we seemed to identify all the problems and risks that we might have in our project earlier in both functional and design specifications, we came into contact with several unexpected difficulties in the actual developments and implementations. Fortunately, it was good that we solved them and found out proper alternatives immediately after group discussions. There were no major interpersonal problems within the group, just high tension around deadlines and after late nights. I am satisfied with what we have accomplished this semester.

## 6.4  Lawrence Wong

In the beginning of the semester it was suggested that, and I'm paraphrasing, we enter 340 as an engineering student and exit 340 as an engineer. I would tend to agree. In past courses, the workload would be just as formidable and perhaps even more difficult. But there were always deadlines to guide your time management and TAs to assist on the difficulties encountered. In our development of the ATS, although the work was not as complicated as certain "shaft" engineering courses, there certainly was plenty of it and we needed to watch our schedule very carefully. Many times I was faced with the temptation to attend to the immediate homework or study of other courses rather than doing 340. I am happy to say that I have learned a great deal about weighing these temptations and organizing my time properly.

Another key difference between 340 and other "shaft" courses is that there was a fair bit of self-learning, where problems would have to be solved by myself or with the help of other group members. No one would be there to help solve a bizarre compiling or linking error. I am now more confident in myself to solve future problems.

Finally, I felt that the teamwork in general was smooth and it was great fun working with the team. Team meetings and discussions were always useful and interesting, as well as entertaining. There is a lot of talent in this group, and I am very pleased that I had a chance to work with them.

## 6.5  Janice Wong

In this project, I was responsible for creating a graphical user interface. We used different Windows programming tools such as Visual Basic and Visual C++. Although I have modified old code and built new features on an existing program during my previous coop work terms, this is the first time developing a program from scratch.

The main challenge with the user interface program was to incorporate all the subprograms in the ATS. This requires understanding of the subprograms written by the other group members. We once considered using dynamic linked libraries (DLLs) to call functions written in other languages but finally we decided that calling executable programs of other subsystems would be the most appropriate solution. These are all new experiences for me. Also, working with and integrating large scale designs would be a very important skill out in the industry.

Since our product is so innovative, we had to do a wide range of research to determine the most cost-effective and efficient approach to our design.  The major obstacles we faced were finding a low cost yet effective sensor to detect the ball positions and colors.  Possibilities we considered included tactile, capacitance and touch sensors, but after extensive research and consultation from faculty members, we found that image processing using a digital camera was the best choice in all aspects.

In addition to the technical aspects, I have also improved my interpersonal and budgeting, and especially pool skills; because in the past four months, we all have received hard-core training in the strategies of pool!