

## **Project ACE (Accessible Computing Education)**

**Reza Hammond**

Undergraduate Student – University of Delaware

[Rkweku@udel.edu](mailto:Rkweku@udel.edu)

**Dr. Stephanie Ludi**

Project Coordinator – Professor at Rochester Institute of Technology

<mailto:salvse@rit.edu>

## Table of Contents

1. Abstract	-----	3
2. Introduction	-----	3
3. Methodology		
3.1. The Language	-----	3
3.2. The Software	-----	3
3.3. Sensors	-----	4
3.4. The Activities	-----	4
4. Limitations	-----	5
5. Future Work	-----	5
6. Conclusion	-----	6
7. References	-----	6

## **1. Abstract**

As the world becomes more dependent on computers, those who are visually impaired are left behind as computers have been engineered to be used only by those who can see. Visually impaired students who are interested in computers may choose career paths that involve minimal interaction with computers because the challenge of using a computer proves too difficult without proper training and efficient software. This can be a loss of great talent as many of these students may have great potential in the field of computing. The goal of this project, therefore, is to spark an interest in computing for visually impaired students. This paper will present ideas for how this may be conducted with several activities designed for Lego Mindstorm robots during the summer of 2009.

## **2. Introduction**

Computers are becoming exponentially faster and better all the time. Computers are capable of performing tasks that would take people months, or even years, in seconds. In fact, computers today are capable of performing tasks that would have taken computers twenty or thirty years ago months, or even years, in seconds. These advancements are made possible by the hard work of all types of people: men, women, white, black, deaf, blind, etc. Computing, or anything for that matter, should not be limited to one race, color, gender, or to people without disabilities, as a person from any group may have the next best idea for something. That is where Project ACE comes into play. People work hard, appropriately so, for the advancement of computing for minorities and women, however, not many people are pressing for advancement in the field of computing for students with disabilities. The goal of project ACE is reach out to students who are visually impaired and try to spark an interest in programming and to let them know that the resources needed to pursue a career in computing are available. This will be goal will be attempted to be achieved with programming challenges using Lego Mindstorm robots. This project is a vision of Dr. Stephanie Ludi of Rochester Institute of Technology. She is a living example of the type of person that may find a career in the field of computers through this project.

## **3. Methodology**

### **3.1 The Language**

A very important part of this project is the language. One popular and fairly easy to use language for programming Lego Mindstorms is NXC (Not eXactly C). Several key techniques seen in other languages like proper variable naming, proper use of comments and even the proper use of data abstraction can be taught with while using NXC. It is believed that NXC provides a good outlook for how programming is done in more advanced languages like C and Java. This language will not only help give students an idea of what programming is truly like, but it gives the programmer a lot of power over the robots.

### **3.2 The Software**

#### **3.2.1 Programming**

The software used for programming the robots is BrcixCC. This program will allow not only the programming of NXC, but it will also be the direct link between the NXT brick (the "brain" of the robots) and the computer. In short, the code will be written and loaded to the bricks all through this program.

#### **3.2.2 Designing and building the robots**

All of the robots are designed in Lego Digital Designer, which is available for free to download at <http://ldd.lego.com/>. This program allows for virtual models to be built, and then step-by-step instructions to build that model from scratch are then provided. This program will help the students build the robots that are needed for each activity.

### 3.3 Sensors

There are several different types of sensors that are made for Lego Mindstorms. Only three come with a Mindstorm set, a light, ultrasonic, and two touch sensors, however, there is a plethora of sensors available from HiTechnic. The sensors that are used in the activities discussed in this paper are listed in table 1.

Light Sensor	The light sensor will emit a small amount of light from an attached LED. The amount of light reflected back into the light sensor is what the light sensor will return. The returned value will be anything between 0 and 100.
Ultrasonic Sensor	The ultrasonic sensor measures the distance of an object directly in front of the sensor. The manufacturer lists its range from 0 to 255 cm.
Color Sensor	The color sensor detects the color of an object directly in front of the sensor. It will return a value between 0 and 17, where each integer has a specific color assigned to it.
Touch Sensor	The touch sensor will return either 0 if the sensor is not pressed, or 1 if it is pressed.
Infrared Sensor	The infrared sensor detects whether or not there is an infrared light source in a 240-degree range. If there is an object, the infrared sensor returns a value between 1 and 9, where 1 is the very left, and 9 is the very right.

Table 1: Information on the sensors used in the activities

### 3.4 The Activities

#### 3.4.1 Forklift

The forklift activity was designed to have three parts to the activity: line following, raising and lowering an object when a platform is seen, and detecting a color and performing an action at a specific color. For the line following activity, a light sensor is used to detect whether or not the forklift is on a path. For the raising and lowering of an object, an ultrasonic sensor is used to detect the distance that a platform is away from the forklift so that the forklift may lift the object and then lower it onto the platform when it is close enough. For the final activity, all tasks are combined, and the incorporation of a color sensor is used to be able to determine if the forklift should turn left or right to find the platform.

#### 3.4.2 Racing Robots

The racing robots is a fun activity that will require students to choose from several line following algorithms to successfully program a robot that will race around a track with several lanes while competing against one or more robots. There are a few good line following algorithms, each with its own perk over the other. After an algorithm is selected and programmed, the robot, since it is competing against other robots on the same track, must have the ability to change lanes when another robot must be passed or to get to the inside lane. This second activity had the robot change lanes to pass another robot, but it was not set up to change lanes safely. The final activity will be to implement safe lane changing. If another robot is in the lane that the robot wants to move to, then the robot should not move to that lane for it will crash into the other robot.

#### 3.4.3 Trash-Collecting Robot

The trash-collecting robot was a fun idea to use the ultrasonic sensor more in-depth than before. The first activity of this is to have the robot search for and approach a cup placed in an enclosed space. It was found that the best way for the robot to approach an object from a distance is to use a sort of line

following method because the ultrasonic sensor does not only see object directly in front of it, but slightly to the sides. The next activity is to have the robot grab the cup and then move toward the blue border. When the border is found, the robot should follow this line until the terminating color is found. The final part of the activity is to read the color of cup and place it in the designated spot for that color. The robot should then move back to the enclosed area and search for another cup. If another is not found, the program will end.

#### **3.4.4 Predator/Prey**

This challenge is a bit more open ended for the students, and also a bit more challenging than the other activities. The predator/prey activity was designed to have two robots work together to play a game of tag, or simulate predator and prey relationships in the wild. The “predator” robot is programmed according to, but not limited to, the specifications given in the challenge instructions. The “prey” robot is set up to be connected to a wireless Playstation 2 controller so a user can try to evade the predator robot. The predator and prey robots must be set up to communicate with one another so that when the predator touches the prey, the prey must know that it has been captured and the programmed will then be killed. There will be four species programmed as predators and preys, each with limitations to sensors and movements. These limitations are incorporated to encourage the students to think about creative ways to program each predator’s artificial intelligence. There are four species in this activity, a leaf, a grasshopper, a snake, and a hawk. As stated before, each species will have limitations to sensor usability and its movements. The preys that must be programmed are the leaf, grasshopper, and snake, and the predators that must be programmed are the grasshopper, snake, and hawk. The hawk activity is very open-ended and requires students to think outside of the box to develop a creative solution to the problem. As much of this activity is open ended, functions that were required in previous challenges are more optional in some cases. The hawk has several functions that can be added into the code to make it better than what is described. This makes this challenge leaves a lot in the students’ hands, which was the goal of this activity. It was designed to spark innovation in the students, much as what is needed everyday in computing.

#### **4. Limitations**

As stated before, this purpose of this project is to develop challenges that will spark an interest in computing for visually impaired students. Only one of the students that was working on this project is visually impaired, so predicting whether or not a task that is visually demanding would be doable was tough. Also, many problems aroused with the documentation of the robots in *Lego Digital Designer*. There were times when certain pieces were fitted together fine in real life, however, the program restricted the connection of two pieces. This problem was never corrected without either changing the model or adding pictures and videos; however, pictures and videos cause problems because the student may not realize that they must stop at a certain step in the building process to check for a picture or video. Finally, the last problem that was encountered was continuing the projects students had been working on before the summer of 2009. Without proper documentation from the previous students, knowing where they were going with their code, or even what they wanted to do with their robot, became a challenge in itself.

#### **5. Future Work**

As the work being done in this paper is over a year old and is still an ongoing project, there exists much room for improvement and more work. The following are some possibilities for future work:

- Remove punctuation, i.e. semicolons at the end of lines in the code to allow for screen readers to properly read the code.
- Fix the bugs in Lego Digital Designer so additional photographs and videos do not need to be appended to the files, as these may be confusing and hard to follow.

- Fix bugs within the completed activities so what is being asked is clear and doable.
- Perform more tests and workshops with students so the activities may be completed flawlessly without the supervision of an instructor trained by those who designed the activities.

## **6. Conclusion**

As there is much work to be done, the conclusion of this paper hardly indicates the conclusion of this project. By the end of the summer, one workshop had been completed with the use of the forklift activity. Although there is no actual data to display in this paper from that workshop, it can be said that many of the students had a very good time performing these challenges. There were hard parts, and there were some easy. It was a learning experience for the students as well as for everybody working on this project. While the students learned about the wonderful world of programming, those working on this project learned of some good things about the challenges, and some bad things about the challenges. What can be said is that the challenges designed will provide as a useful tool in the continuation of this project.

## **7. References**

Yocum, Dale. *NXT Tutorial*. NXT Tutorial. 2007. [http://www.ortop.org/NXT\\_Tutorial/index.html](http://www.ortop.org/NXT_Tutorial/index.html).