Cass Pangell
Jeremy Norman
Nathaniel Cockrell
Justin Guerra

BPC Tech Club

Abstract

The goal of this project is to try and create interest in computer science among female high school students. The project creates fun and interesting activities that will introduce students to computer science in an after-school technology club. The project actually has two separate aspects to it: the first is a small animatronic creature and the second involves a small robot and wireless sensor. The animatronic creature is small, inexpensive, and remote controlled. The creature is able to move forward and backward, has movable arms, and a moveable head. After the robots completion, we created tutorials on how to use Blender, an open source 3D modeling/animation program. There are tutorials on how to build the creature and use a green screen. The second project uses a Scribbler robot, a small programmable robot, and a Sun Spot. The Scribbler is a relatively inexpensive programmable robot with a variety of sensors. Sun Spots are small wireless sensors that can be coded to perform a variety of tasks. The Scribbler comes with eight factory demos loaded by default, these range from simple tests of its sensors to more advanced programs like room exploration. The demos were re-created in Scribbler's provided graphical programming language and we wrote tutorials on the process. Each is a starting point for students to write more advanced programs on the Scribbler. The final component of this project combines the Scribbler and Sun Spot. The Scribbler explores a room while the Sun Spot tracks the temperature. A Java applet on a nearby computer displays this information graphically. There is a tutorial on how to run this applet.

Project 1:Animatronic Creature

The animatronic robot (AR) is loosely based off the main character from the upcoming PIXAR movie “Wall-e.” Specifications require that the AR be no taller than 24” and is made from easily obtainable parts within a budget of $100. The AR is able to move forward and backward, move its arms, and has a movable head. This allows the AR to seem more realistic. Using Blender, the AR is captured into a short film to provide a demonstration of the uses of a green screen and 3D animation software.

Building Materials

All building materials were easily obtainable from Home Depot, Target, hobby stores, and online sources. The frame of the AR is constructed with Styrofoam board because of its rigidity and low weight. Hot glue is used as the main connection between pieces of Styrofoam board and the motors.

Body

The completed body of the AR was redesigned from an initial PVC cube to be composed of 3/4” thick polystyrene board that is coated with a thin plastic film. The Styrofoam board was cut into several sections, forming a 1' cube (figure 1). The top and bottom sections are cut at 1'x1', the front and back are cut at 1'x10.5", and the sides are cut at 10.5"x10.5", fitting together to form a complete cube. Each of the side and front pieces were glued together using hot glue and then connected to the bottom section. The top of the cube is removable to allow access to the components within and therefore sits on two #8 2.5" screws that prevent it from sliding. These two screws are carefully drilled into the front at back pieces exactly halfway along the upper edge; the top piece has two small holes cut in the under side where the heads of the screws sit to prevent movement.

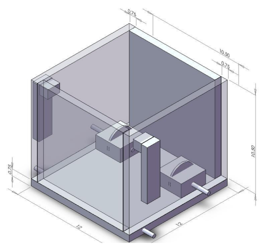

Figure 1

Arms

Each arm of the AR was constructed from 6"x1.6"x3/4" sections of Styrofoam board. Each arm is connected directly to the shaft of a small motor that was removed from remote control (R/C) cars. The motors are mounted directly into the sides of the Styrofoam cube. A small hole about 1.5"x1" (just large enough for the motor to fit through) is cut in each side board. The holes center was measured at 3 inches in from the front, and 2.5 inches down from the top, putting the arms near the front of the AR. Each of these motors is mounted permanently with hot glue in order to prevent tipping and to provide excellent stability.

Wheels

The wheels of the AR are designed to look like tank treads in an attempt to mimic the look of Wall-e. Each side of the AR has two wheels, taken from the R/C cars. The front wheels on each side are mounted directly into the body of the AR using #8 2.5" screws. The screw is drilled directly through the axel point of the wheel and into the base of the AR body at 2 inches from the front edge. If the wheel does not turn freely on the screw, the center of the wheel axel needs to be drilled until the hole is large enough to allow free turning along the wheel. The back wheels of the AR provide power. The entire assembly of the power unit, rear axel, and circuit board from each R/C car is removed and placed in the bottom of the AR cube. To provide the best stability, each of the motor units is mounted into the lower section of the AR using hot glue. A small section of the Styrofoam board will need to be carefully removed, allowing the motor and axel section to sit low within the Styrofoam board base. It is important that the motor unit sits as close to each side wall as possible to allow for the axel to be poked through the Styrofoam wall and still have enough room to mount the rear wheel. After each of the motor units has been fully mounted to the base of the AR, a rear wheel is glued on to the axel that protrudes from the lower rear Styrofoam wall. This axel protrusion will appear at approximately 2" from the back and 1" from the bottom of the Styrofoam cube.

Head

The head of the AR is made primarily from components from a small remote control car: the circuit board, wireless receiver, and battery pack were used. The main portion of the head is constructed from a foam noodle (a form of flotation device, figure 2). The head is able to rotate 360 degrees at the neck and is also able to move up and down. Movement is controlled using R/C controllers. An instruction manual for building the

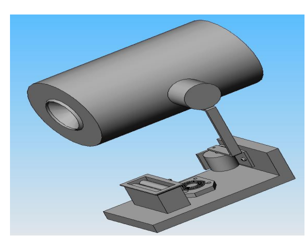

Figure 2

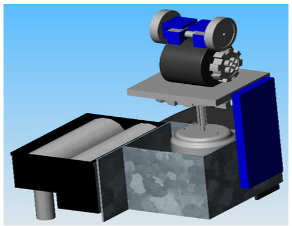

Figure 3

head of the AR is included. The neck and motor assembly was constructed using motors and LEGOs, in addition to the components from the R/C car (figure 3). We created a LEGO® kit that allows students to construct the neck relatively easily. The LEGO components are nice because they are available online for a low cost (around $1.50 plus shipping) and will be familiar to most students.

Mechanical Features

The AR is made with various components from two remote control (R/C) cars. An appropriate R/C car costs between $10 and $20. However, these parts can be purchased separately at a hobby store or in bulk for more flexible prices. The advantage of purchasing an R/C car is that it provides a working circuit board with a specific remote control frequency already set. A detailed instruction manual for disassembly and assembly is included. The car's motors, battery packs, circuit boards, wheels, and gears will be used in the AR. The motors that control the turning of the R/C car are generally smaller than those that move the car, therefore the turning motors are used for the arms of the AR, and the more powerful motors are used for the wheels of the AR. Some simple soldering is required.

Green Screen Implementation

Using Blender, the open source 3D animation software, the AR was operated in front of a green screen and integrated with examples of animation and visual effects. There is an included tutorial on how to use the green screen capability of Blender. The purpose of this section is to allow students to familiarize themselves with 3D animation software. The tutorials will allow students to create interesting products with minimal difficulty.

Conclusions

During the first two weeks of the project we became familiar with Blender and its capabilities. We produced tutorials on animation techniques and green screen implementation in this time period. In addition, we created an animated background in order to liven up the green screen demo. The third and fourth week was spent designing, redesigning, building, and rebuilding the animatronic robot. During week five, construction of the animatronic robot was finished, in addition, we also wrote the building guide. Week six was spent finalizing the AR, filming the AR in front of a green screen, and finalizing the reports.

Lessons Learned

Although it initially appeared fairly straightforward to produce an animatronic robot (AR), there were multiple difficulties that we encountered during the construction of the AR.

Initially, the body of the AR was built using small sections of PVC piping to provide excellent stability with minimal weight. Unfortunately, we found that the PVC piping was too heavy for the small R/C car motors. Therefore we needed to find an alternative way to build the frame. Reducing the body to ¾" Styrofoam board made it possible to use the R/C car motors to power the AR.

Another obstacle that was encountered was the frequency of the radio bands to control the AR.

The head, arms, and legs each are controlled by separate remote controllers, but we found that the frequency of the arm controller was identical to that of the leg controller. Therefore, an additional R/C car, with an adjustable frequency, had to be purchased to allow independent control of the arms and legs.

Concerning the circuit boards, the manufactured soldering proved to be minimal and the wires broke off easily. Adding a spot of hot glue over the connections turned out to be a good solution to this problem. Hot glue is an insulator and does not interfere with circuit board functionality.

Powering the AR turned out to be another headache. The motors themselves did not provide enough torque to turn the head up/down or tilt side/side. The neck of the AR was then modified to allow at least some movement. In the future it would be better to calculate how much weight the motors can realistically handle as opposed to figuring it out with trial and error.

Weight was the principle setback in the creation of the AR. The inexpensive R/C car motors can only provide a limited amount of power. This could have been prevented by originally designing the AR with minimal weight in mind.

Parts for the AR can be purchased online or at local retailers. However, ordering parts online took time and slowed the construction phase of the project. Better planning could would prevent this problem.

Future Work

If additional time was available, we would have liked to integrate the Sun SPOT modules with the head of the AR to allow for a personality. This would allow for programmable emotions to be added to the robot. It would potentially be useful to produce the robot with a new set of stronger motors and attempt to a more solid body made of PVC. It would also be ideal to create more advanced Blender tutorials.

Project 2: The Scribbler and Sun SPOTs

The second project consists of three parts: a wireless sensor demonstration using SunSPOTs, a set of programmed robotics demonstrations using a small sensor-equipped robot, called the scribbler, and a culminating demonstration combining the two.

AirText

The wireless sensor demonstration uses a Sun SPOT to create an "Air Text" effect. In order to complete this portion of the project, the SPOT will need to use the on-board accelerometer and LEDs to create text in the air by waving the SPOT back and forth. (See figure 4). The text will read "BPC Tech Club." Finally, in order to complete this portion, a power point tutorial will be written to guide a new programmer to open the program in NetBeans, modify the displayed text, build the new program, and deploy it to a SPOT device.

Figure 4: Air Text Effect for first demonstration

Demonstrations

The scribbler portion of the project includes eight small demonstrations. Each demonstration includes an accompanying tutorial written in power point explaining how to recreate each demonstration. Specific goals of each demonstration are outlined below.

Demonstration 0

- This demonstration introduces the light sensors available on the Scribbler
- If light is being detected by one of the sensors, the corresponding LED will be turned on

Demonstration 1

- This demonstration drives around the room, seeking brightly lit areas
- A flashlight can be used at ground level to steer the Scribbler

Demonstration 2

- This demonstration makes use of the Scribbler's infrared sensors
- If an object is placed in front of the scribbler the infrared sensors detect it and turn on a corresponding LED

Demonstration 3

- This demonstration uses the infrared sensors to avoid objects while moving
- The scribbler drives in a straight line and if it encounters obstacles it takes appropriate evasive action

Demonstration 4

- This demonstration makes use of the Scribbler's line sensors
- If a line is detected by a sensor the corresponding LED is turned on

Demonstration 5

- This demonstration combines the line sensors with movement
- The Scribbler uses its line detecting sensors to try and follow a path, if the path is lost it searches for another to follow

Demonstration 6

- This demonstration is intended to be used with the Scribbler's built in scribbling abilities
- The Scribbler first draws a figure eight, then after a brief pause the Scribbler draws a square

Demonstration 7

- This final demonstration puts the Scribbler in ambulance mode
- Ambulance mode includes driving around while avoiding obstacles, playing lots of sounds, and flashing its LEDs

Collaborative Demonstration

Figure 5: Scribbler with Sun SPOT attached

The last portion of the second project, the "Synergy Demonstration," combines the Scribbler and the Sun SPOT with Velcro (see figure 5). The

SPOTs are used to record temperature readings as the Scribbler moves around the room. A graphical window on a nearby computer displays a virtual thermometer of the sensor readings as well as a line graph of previous readings. We also added the ability for the desktop application to make a thermal map of a room while the Scribbler explores the room by using the SPOT's accelerometer to estimate position.

Design

AirText and the Scribbler demonstrations did not require extensive designing. Both simply required that a small set of functional requirements were met.

The "synergy" demonstration requires software on both a wireless sensor as well as the desktop. In order to make them work together we designed a very simple networking protocol. In the protocol, the sensor will listen for commands from the computer and then respond appropriately. The sensor must calculate its position every 65 milliseconds to avoid the high precision timer from overflowing, the network protocol must work asynchronously to avoid packet delivery delaying the calculation.

In our scheme, the desktop will only make requests to the sensor rather than the sensor periodically sending information. This is so that the sensor can focus on its position calculations, which can become computationally intense for such a small device, but also so that the data set resolution can be controlled by the user if necessary.

In order to complete the application, the desktop uses a very simple scheme outlined in figure 6. The display simply works by loading up a renderer which looks into the data source for information. The renderer returns an image to the display based on the previous data.

Protocol

The desktop and the sensor work together by using a simple asynchronous network protocol of context-free commands and responses. The desktop is able to send requests for data, and when the sensor hears these requests it responds with the temperature and location of the sensor. The context-free portion of the protocol allows the sensor the flexibility to skip a report when high priority calculations are en-route. Specifically, the position based on acceleration must be calculated at least every 65 milliseconds since the high resolution timer can overflow otherwise and lose the measurement as well as invalidate the internal position variables. Sending packets of even 32 bytes in size can take more than a full second to transmit with the limited radio power when in a reliable delivery mode.

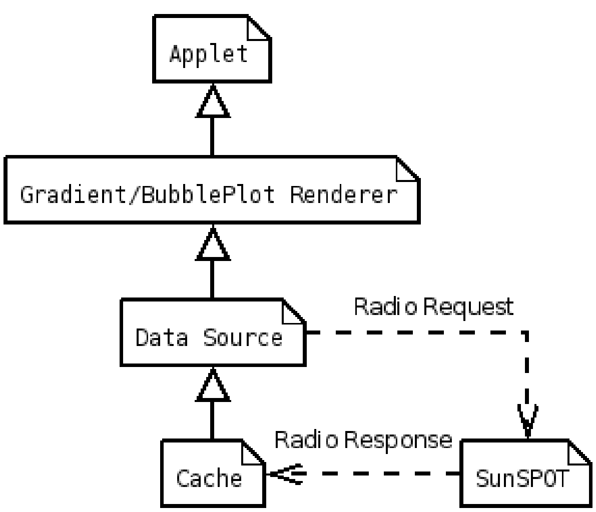

Figure 6

Desktop Software

The applet displays the location and temperature information in a bubble plot. A bubble plot displays data in circles of varying size. The area inside the circle represents potential values of the plotted point. Bubble plots are useful when you are uncertain about the accuracy of the data. Temperature is represented by coloring the bubbles differently. For example, the coldest temperature represented is pure blue and the hottest temperature is pure red. Temperatures between the two extremes smoothly transition from blue to red.

Sensor Software

The sensor's software is very simple. The software makes use of the multi-threaded environment to make each task a mini-process. Specifically, the network protocol handler, position calculator, and temperature reader will run in their own threads and only report to each other when a response is necessary. This makes it easy to decouple the different tasks and modify them individually.

Decoupling the position calculator into its own thread is particularly important, as mentioned before, to enforce task priority. It also makes it easier to rewrite the positioning code, which so far seems very error prone and unreliable, without needing to rewrite the rest of the program.

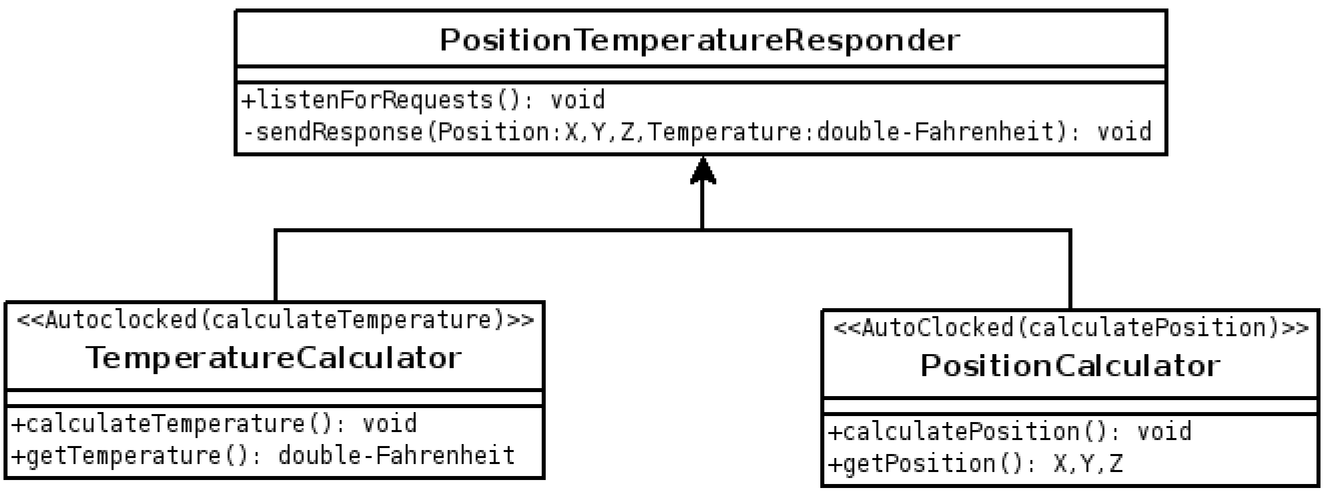

Figure 7

Conclusions

All of the outlined demonstrations for both the Scribbler and the Sun Spot and their respective tutorials were completed by week 4 with exception of the thermal map. The thermal map posed some special problems because sampling the accelerometer so often caused the accelerometer chip to heat up. This would normally be fine, except it turns out the thermometer and the accelerometer sit on the same chip. This caused the temperature readings to read a constant temperature that did not respond to room conditions.

We had enough time left over to work on other demonstrations for our client. Specifically, we started working on making a robotic controller which used one Sun Spot’s acceleration data to transmit control information to a controlling spot. The controlling spot then uses some extra circuitry we designed to drive motors. These demonstrations include a small wooden robot and a plan for a blimp. We also designed a simple circuit that controlled three external LEDs. These LEDs could then be used to control the Scribbler.

We were unable to complete the blimp because of time, but we did finish the design and the outline of the software. We simply did not have enough time to build the blimp.

One other application we wrote was a small program for the Sun Spot that turned on one of three external LEDs based on whether the temperature was increasing, decreasing, or staying constant. This is a "loose end" for allowing the club members to play with programming the Scribbler and using temperature information, which the Scribbler does not normally have.

Lessons Learned

We learned quite a few small lessons during our work. Most of our lessons revolve around the Sun Spot. One problem that arose during use of the Sun Spots was difficulties integrating outside hardware with the Spot. For example, the Sun Spots come with on board I/O pins, but they turned out to be somewhat difficult to access. The headers provided by Sun were fairly useless and we had to fabricate our own. This was both time consuming and it was more difficult to get responses than expected.

The other main problem related to Spots occurred when a program would freeze up. The only solution we were able to find was to re-flash the Spot. Re-flashing takes from five to ten minutes and really slowed our development.

The last lesson learned for us was that ordering parts online can take an unexpectedly long time. It's important to have a plan of what components you want in advance. We dealt with some of these supply problems by buying 200 NPN Transistors and 200 1 kilo-ohm Transistors. These allowed us to easily create missing gates by hand if we missed any. This cost us $15, but turned out to be useful in the end since it allowed us more flexibility.

Future Work

The work we did allows a lot of room for future work. During our project, we designed two interfaces to allow easy expansion.

The first interface we designed provides a DB-25 (standard LPT printer port) connector for connecting to the Sun Spot. This allows for a header that plugs into the Sun Spot to be reused with different circuits without wasting time on creating a Spot Header (which turned out to be very time consuming). This makes adding new sensors as easy as connecting a circuit to a standard male DB-25 connector and plugging them into the Sun Spot header (we also are supplying a few extra connectors for this purpose).

The second interface we designed was a controller connector. This is again a DB-25 connector, but with a different pin layout. This connector allows the high current control from the controller board to be connected to different physical motors. We implemented this interface for two different demonstrations. One is a small robot with mobile arms, legs, a rotating head, and blinking eyes. We also implemented the motors for the blimp controller.