Next Steps for Your Final Project

Reflect on your midterm project. You may expand your midterm project, or take a new approach to a Tangible User Interface that takes advantage of human senses beyond traditional user interfaces.

You may continue to work as a group (maximum of 3 members) or as an individual. If you work in a group, be clear about each member’s role in the project.
Next Steps for Your Final Project

**10/30**
Finalize your group for project and post your proposal on the course website. Create a list of materials you need. (We may be able to help.)

**11/15 & 11/20**
In-class final project progress report and critique.

**12/9 & 12/11**
Final project exhibition. Present your prototype.

**12/15** Final write up due in the ACM SIGCHI Extended Abstract format (6-8pgs)
Recommendations

Do your background research
Towards “publishable quality” work
- Avoid making redundant effort
- Discuss rationale for your design and your contributions

Engage in many design critiques, early
Iterate! Share your idea with others and test its potential, early.

Use the resources

Kimiko: Tuesdays 1-2pm at South Hall 314,
Thursdays 3:30-4:30 at South Hall 110, and by appointment.
Patrick: Wednesdays 1-2pm at Stanley Hall B144, and by appointment.
Reza: by appointment
Liz: by appointment
<table>
<thead>
<tr>
<th>Week</th>
<th>Tuesday LECTURE</th>
<th>Thursday LAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td></td>
<td>08/28 Introduction</td>
</tr>
<tr>
<td>Week 2</td>
<td>09/02 Activity Theory and HCI</td>
<td>09/04 Introduction to Physical Computing</td>
</tr>
<tr>
<td>Week 3</td>
<td>09/09 Tangible Bits</td>
<td>09/11 Digital I/O with Arduino Boards</td>
</tr>
<tr>
<td>Week 4</td>
<td>09/16 Containers, Tools, and Token: Taxonomy of TUIs</td>
<td>09/18 Sensing 1: Potentiometers</td>
</tr>
<tr>
<td>Week 5</td>
<td>09/23 Calm Computing and Ambient Media</td>
<td>09/25 Sensing 2: Force sensors and photocells</td>
</tr>
<tr>
<td>Week 6</td>
<td>10/03 Human Centered Design</td>
<td>10/02 Output 1: Piezo speakers</td>
</tr>
<tr>
<td>Week 7</td>
<td>10/07 Design and Innovation</td>
<td>10/09 Output 2: DC motors</td>
</tr>
<tr>
<td>Week 8</td>
<td>10/14 Midterm Project Review</td>
<td>10/16 Output 3: Servo motors</td>
</tr>
<tr>
<td>Week 9</td>
<td>10/21 Guest Lecture by Hayes Raffle</td>
<td>10/23 Output 4: Simple Mechanics</td>
</tr>
<tr>
<td>Week 10</td>
<td>10/28 Guest Lecture by Wendy Ju</td>
<td>10/30 Synthesis 1: Invent a music instrument (group work)</td>
</tr>
<tr>
<td>Week 11</td>
<td>11/04 Guest Lecture by Liz Goodman</td>
<td>11/06 Synthesis 2: Invent a music instrument (group work)</td>
</tr>
<tr>
<td>Week 12</td>
<td>11/11 No class: Holiday</td>
<td>11/13 Guest Lecture by Mike Kuniavsky (ThingM)</td>
</tr>
<tr>
<td>Week 13</td>
<td>11/18 Final Project Progress Report and Critique</td>
<td>11/20 Final Project Progress Report and Critique</td>
</tr>
<tr>
<td>Week 14</td>
<td>11/25 Evaluating TUIs</td>
<td>11/27 No class: Thanksgiving holiday</td>
</tr>
<tr>
<td>Week 15</td>
<td>12/02 Guest Lecture by Patrick Goodwill</td>
<td>12/04 Summary</td>
</tr>
<tr>
<td>Week 16</td>
<td>12/09 Final Project Exhibition (Part I)</td>
<td>12/11 Final Project Exhibition (Part II)</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Tuesday LECTURE</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Week 1</td>
<td>08/28</td>
<td>Introduction</td>
</tr>
<tr>
<td>Week 2</td>
<td>09/02</td>
<td>Activity Theory and HCI</td>
</tr>
<tr>
<td>Week 3</td>
<td>09/09</td>
<td>Tangible Bits</td>
</tr>
<tr>
<td>Week 4</td>
<td>09/16</td>
<td>Containers, Tools, and Token: Taxonomy of TUIs</td>
</tr>
<tr>
<td>Week 5</td>
<td>09/23</td>
<td>Calm Computing and Ambient Media</td>
</tr>
<tr>
<td>Week 6</td>
<td>10/02</td>
<td>Human Centered Design</td>
</tr>
<tr>
<td>Week 7</td>
<td>10/07</td>
<td>Design and Innovation</td>
</tr>
<tr>
<td>Week 8</td>
<td>10/14</td>
<td>Midterm Project Review</td>
</tr>
<tr>
<td>Week 9</td>
<td>10/21</td>
<td>Guest Lecture by Hayes Raffle</td>
</tr>
<tr>
<td>Week 10</td>
<td>10/28</td>
<td>Guest Lecture by Wendy Ju</td>
</tr>
<tr>
<td>Week 11</td>
<td>11/04</td>
<td>Guest Lecture by Liz Goodman</td>
</tr>
<tr>
<td>Week 12</td>
<td>11/11</td>
<td>No class: Holiday</td>
</tr>
<tr>
<td>Week 13</td>
<td>11/18</td>
<td>Final Project Progress Report and Critique</td>
</tr>
<tr>
<td>Week 14</td>
<td>11/25</td>
<td>Evaluating TUIs</td>
</tr>
<tr>
<td>Week 15</td>
<td>12/02</td>
<td>Guest Lecture by Patrick Goodwill</td>
</tr>
<tr>
<td>Week 16</td>
<td>12/09</td>
<td>Final Project Exhibition (Part I)</td>
</tr>
</tbody>
</table>
Output 3: Servo Motors
Making motions with servo motors
Servo Motor
Servo Motor

0 degree

180 degrees
Servo Motor

1. Gears
2. Potentiometer
3. Motor
4. Electronics
Parallel Motion Linkage

Mechanism Inside a Toolbox

http://www.technologystudent.com/cams/link1.htm
4–6g

37.2g (1.3oz)
180°

Ground (0V)
Power (+5V)
Control (PWM)
0 degrees

45 degrees

180 degrees

- 0 degrees: high 1000 microseconds, low
- 45 degrees: high 1250 microseconds, low
- 180 degrees: high 2000 microseconds, low
Servo Motor

1. Gears
2. Potentiometer
3. Motor
4. Electronics
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
```cpp
void loop() {
  val = Serial.read(); // read the serial port
  if (val >= '1' && val <= '9') {
    val = val - '0'; // convert val from character variable to number variable
    val = val - 1; // make val go from 0-8
    pulseWidth = (val * (maxPulse-minPulse) / 8) + minPulse; // convert val to microseconds
    Serial.print("Moving servo to position ");
    Serial.println(pulseWidth,DEC);
  }
  updateServo(); // update servo position
}

// called every loop().
// uses global variables servoPin, pulseWidth, lastPulse, & refreshTime
void updateServo() {
  // pulse the servo again if the refresh time (20 ms) have passed:
  if (millis()) - lastPulse >= refreshTime) {
    digitalWrite(servoPin, HIGH); // Turn the motor on
    delayMicroseconds(pulseWidth); // Length of the pulse sets the motor position
    digitalWrite(servoPin, LOW); // Turn the motor off
    lastPulse = millis(); // save the time of the last pulse
  }
}
```

---

Serial message: 3
servo_serial_simple ready
moving servo to 60

---

27
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
Assignment

Post descriptions and photo(s) of your crawler on the course website. Once you get your crawler to move forward, perhaps you would want to generate movement from your program and use your potentiometer to control the speed of the movement. You may also team up with a friend and use two servos instead of one.
Thanks!