A. Digital Camera (Canon 850 IS). This interface is used in many contexts. It is used to capture moments and to help preserve memories. Because this item is one that is useful to everyone, it should be designed in a way that is very simple to understand and use. One of the strengths that this camera has is that it has a very large viewing screen. I think that this is a good design because this is where you view what you are doing on this camera. Another strength that this camera has is that it has a very large button to press to take the picture. This is a good design because it is easy to find and press even if you are not looking directly at the button. This is important because the person taking the picture cannot be looking at the button when it needs to be used. A last strength of this camera is that the on/off button is small and not near any of the commonly used buttons on this camera. This is a good idea because the user won’t accidentally turn the camera off while using it. One of the weaknesses of this camera is that the special features of only showing one color in the picture is hard to find. In fact, I still do not know how to use it. This is a bad design because people cannot easily figure out how to use this! Basically an easier design is one where there is a simple button to find it. Another weakness is the circular button in the back to choose different menu options. The circle design makes it hard to know exactly what menu choice you will be choosing because the boundaries for where you should press this button are not clearly defined. I think a different shape button is better. Or four separate buttons that are arranged in a circular shape. A last weakness of this camera is after a few moments of not doing anything, the screen “sleeps” when you are about to take the picture. This is not good because people do not know what to do once this happens. One way to improve this is to have the camera make a beeping noise before it sleeps so at least the user knows that something is about to happen.

B. Cell Phone (Latest Verizon Chocolate). This interface is used all the time by all different types of people. Because this item is used by everyone, it should be designed in a way that is simple to understand and use. One of the strengths of this phone is that it has a “spin wheel” to scroll through the list you are looking at. This is a good idea because it takes much less time and energy to get to the list item you are searching for. Another strength that this phone has is that it is a “slide” design. Half of the phone slides up to reveal the buttons underneath. This is a good design because the buttons are protected and will not be accidentally pressed when not in use. A last strength is their sound design. There is an option to have all sound off except for your alarm. This is a great idea because then you can sleep without people waking you up but you will still get up at the right time because the alarm is on. One of the weaknesses of this phone is that there is no way to look through your contact list while you are on the phone. This is not a good design because when you want to give someone somebody else’s number, you can’t. One way to fix this is that while you press the “contacts” button, the phone call is not affected in any way. Another weakness is when you call your voicemail and press buttons to pick your menu choice, the phone immediately goes to an “Add Contact” menu. This is not only a bad design but very annoying as well because you need to constantly try to get out of this menu in order to continue
with your voicemail procedures. The simple way to fix this is to just not have every number button automatically go to that menu. A last weakness of this phone is the speaker is not very good. When calling other people, it is hard for them to hear me. This is not a good design because the whole point of the phone is to communicate with others. One way to fix this is to maybe make the speaker area larger.

C. MP3 Player (SanDisk). This interface is also used all the time by all different types of people. Because this item is used by everyone, it should be designed in a way that is simple to understand and use. One of the strengths of this MP3 player is that it holds a lot of memory. You can put a lot of songs on this mp3 player. This is a good design so that you can have a large variety of songs and not need to interchange music that often. Another strength for this mp3 player is that it is very small so you can easily bring it with you wherever you go. This is a good design because that way the user can use it whenever they want to and they can bring it with them all the time. A last strength of this mp3 player is that the turn wheel to go traverse through the song choices. This is a good design because it saves time and energy when searching for songs. One of the weaknesses of this design is that it is hard to toggle back and forth between the songs and the playlist. One way to fix this is to clarify on the screen exactly where you want to go to. Another weakness is the spin wheel is bulky so if you cannot really put it in your pocket or it is hard to fit into small places. A way to fix this is to simply make the spin wheel thinner. A last weakness is that the on/off button is very hard to press. This is just an inconvenience. There is already a “hold” function so there is no need to make this button extremely difficult to use. A way to fix this problem would be to change the button.

2. While conducting interviews with 3 different people, many of the problems that they described were the same. One of the problems that was discussed was the extreme lag time with certain programs on the computer. This could be caused by the limits of the OS or the program’s creators did not prioritize performance. One way to alleviate this problem could be to upgrade the hardware or the OS. A better way to go about it is for the creator’s of the program to value productivity and efficiency just as highly as introducing new features. Another problem that was brought up was the issue of having to constantly switch between keyboard and mouse. This is a problem because it is inefficient and it kills productivity due to all of the times that you need to switch between the two. This is due to the fact that most programs require two input devices to operate. One way to alleviate this is to allow the keyboard to access more functionality so that the use of the mouse would decrease. Furthermore, users also complained that programs are hard to learn because of feature bloat or simply too many options presented. This can be addressed by program creators paying more attention to interactive design and removing options that are unlikely to be used. Finally, some people complained about programs scaring them away from doing something. For example, after performing a certain operation the computer will prompt them with a message similar to “Are you sure you want to do this?” If the user is new to the program and does not fully understand the consequences of their actions they will be prone to “cancel.” Instead of asking those questions, the computer should simply continue with the action and allow for the user to undo if they wish to. This is not always a feasible option but should be done as much as possible.

3. See page 4 for table and graph, page 5+ for source code.
4. C. Limitation of applying Hick’s Law to real-world performance tasks is that although the formula may prove that one way of designing something is faster/better, the other one could actually be faster in real life. This is because of how the design of it the item is as well as how users usually look at things and if they are easier to figure out. An example of this is the pie menu versus the linear chart. The pie menu by the formula is the better menu however, in real life the linear menu is used because it is easier for users.
<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.7</td>
</tr>
<tr>
<td>2</td>
<td>5.5</td>
</tr>
<tr>
<td>3</td>
<td>9.7</td>
</tr>
<tr>
<td>4</td>
<td>7.4</td>
</tr>
<tr>
<td>5</td>
<td>5.6</td>
</tr>
<tr>
<td>6</td>
<td>4.1</td>
</tr>
<tr>
<td>7</td>
<td>5.1</td>
</tr>
<tr>
<td>8</td>
<td>4.6</td>
</tr>
<tr>
<td>9</td>
<td>5.4</td>
</tr>
<tr>
<td>10</td>
<td>5.4</td>
</tr>
<tr>
<td>11</td>
<td>4.2</td>
</tr>
<tr>
<td>12</td>
<td>4.0</td>
</tr>
<tr>
<td>13</td>
<td>4.0</td>
</tr>
<tr>
<td>14</td>
<td>4.0</td>
</tr>
<tr>
<td>15</td>
<td>3.8</td>
</tr>
<tr>
<td>16</td>
<td>4.8</td>
</tr>
<tr>
<td>17</td>
<td>4.4</td>
</tr>
<tr>
<td>18</td>
<td>3.2</td>
</tr>
<tr>
<td>19</td>
<td>3.3</td>
</tr>
<tr>
<td>20</td>
<td>3.5</td>
</tr>
</tbody>
</table>
<html>
<head>
  <script src="jquery-1.2.6.min.js"></script>
  
  var attemptNum = 1;
  var attempts = [ ];
  var startTime;

  $(document).ready( function() { 
    var $attemptNum = $('#attemptNum');
    var $userInput = $('#userInput');

    $("#start").click( function() { 
      startTime = new Date().valueOf();
      attemptNum = 1;
      attempts = [ ];

      $('#A').hide();
      $('#B').show();
      $('#C').hide();
      $attemptNum.html(attemptNum.toString());
      $userInput.val('').focus();
    });

    function saveData() { 
      var endTime = new Date().valueOf();
      var input = $userInput.val();
      attempts.push( {input: input, time: endTime - startTime} );

      // Clear input box and update attempt 
      $userInput.val('').focus();
      attemptNum++;
      $attemptNum.html(attemptNum.toString());

      // Update startTime
      startTime = endTime;
    } 

    function quit() { 
      $('#B').hide();
      $('#C').show();
      var correctVal = "zyxwvuts";
      var html = "<thead><tr><td></td><td>Input</td><td>Time (ms)</td></thead><tbody>";
      $.each( attempts, function(i, val){
        var isCorrect = ( val.input == correctVal );
        html += '<tr><td>' + (i+1).toString() + '</td><td class="' + ( isCorrect ? "correct" : "error") + '">' + val.input + '</td><td>' + val.time + '</td></tr>
        html += "</tbody>";
      });
      html += "</thead><tbody>";
      $("#daTable").html(html);
    }

    $userInput.keydown( function(e) { 

  });
</head>
</script>
console.log(e.keyCode);
if (e.keyCode == 13) { // Enter
    saveData();
} else if (e.keyCode == 27) { // Esc
    quit();
}};

$('#enter').click( saveData );
$('#quit').click( quit);

$('#retry').click( function() {
    $('#start').click();
});

</script>

<style type="text/css">
body {
    padding: 0 10%;
    font-family: Verdana, Arial, sans-serif;
}

h1 {
    text-align: center;
}

#B, #C {
    display: none;
}

td {
    padding: 3px 10px;
}

td.error {
    color: red;
}

td.correct {
    color: green;
}

</style>
</head>
<body>
<h1>Survey</h1>

Enter the last 8 letters of the alphabet backwards with one hand as fast as possible. <br/>
To begin, click "Start" and to submit, hit "Enter". To end the trial, press "Esc" or "Quit" to display the results.

<input id="start" type="button" value="Start"/>

<input id="userInput" type="text"/>
<input id="enter" type="button" value="Enter"/>