

Requirements Specification & Project Management Exercise

Team	Class	Week
Adaptive H.I.D.	CPE 495	9 Nov. 2015 – 14 Nov. 2015

Problem Statement

Computers have dropped in price, grown in capability, and leapt into consumer markets in the past few years. Adaptive Technologies (AT) which help persons living with disability interface with a computer have been left behind in this regard. Solutions available are expensive and difficult to obtain. There is a distinct need for human interface devices which meet many AT requirements and are available to wider audiences for more reasonable cost.

Project Objective Statement

Create and study affordable AT that can be crowdsourced and brought to people who currently can't access it.

Marketing Requirements

- **Open Source Design**
 - Make design conducive to materials and tools available to makerspaces.
 - Put extra effort towards making the device easy to customize and iterate upon.
 - Use low cost machinery and materials.
- **Easy to Operate**
 - Must provide as good or better functionality than commercial equivalent.
 - Should be decently portable.
- **Reliable**
 - Must accept all possible input.
 - Must have a robust mechanical design.

Objective Tree or Affinity Diagram

Engineering Requirements

Sliding and Braille Keyboards

- **Hardware bounce time during activation:** <6ms
- **10 Million cycle lifetime.**
- **User Operation**
 - 30 Words per minute
 - 90% Typing accuracy
- **Materials**
 - Thermoplastics – Case, Mounts
 - Stainless Steel – Springs, Case
 - Gold Alloy – Switch Contacts
- **PCB**
 - PCB should be at most 2-layer.
 - PCB should be dispersed enough in design to be easily fabricated with laserjet/home solutions.

Sliding Keyboard

- **Keymap**
 - Without layering: 36 possible characters.
 - Holds [A-Z] and special characters.
 - With layering: 180 possible characters.
 - Holds all of ASCII.

Braille Keyboard

- **Keymap**
 - 8 Keys for extended Braille.

Constraints and Standards

Constraints

- **Target Price Point:** <\$150
- **Portability**

- Should be straightforward to move.
- **2-Layer PCB Designs**
 - More complicated implementations remove the prospect of home manufacturing.
- **Software**
 - Must work on low power ARM SoCs.

Standards

- **USB HID Profiling**
 - Must be recognized as an interface device.
 - USB standard 1.1 or 2.0.
- **3.5mm Mono Jack**
 - Used in a multitude of existing AT interface systems.

Work Breakdown Structure

	Activity	Description	Deliverables/Checkpoints	Duration (days)	People	Resources	Predecessors		
1	Hardware Design								
1.1	Plan Switch Configuration	"Design a basic plan for the keyboard, specifically the placement of the switches."	Initial product design sketch	7	Chris				
1.2	Design Circuitry	Design the circuit necessary in order implement the planned switch placement	Circuit schematic	1	Chris		1.1		
1.3	Purchase Components			7	Chris		1.1		
1.4	Construct Prototype	Build and test our prototype.							
1.4.1	Construct a base	"Construct a platform which holds our microprocessor and switches, the switches are to be aligned based on our initial product sketch."		7	"Chris, Trey"		1.3		
1.4.2	Apply the Circuit	Connect the switches to the microprocessor through use of the schematic designed earlier.		2	Trey		1.4.1		
1.4.3	Provide a method of activating each switch	Add a device between the switches which allows each one to be activated.		5	"Chris, Trey"		1.4.1		

1.5	Program Prototype								
1.5.1	Establish Input	Read input from each switch on the microcontroller	Test data	1	"Trey, Bryant"		1.4.2		
1.5.2	Interpret Input	Program the character representation for each input from the keyboard.	Test data	2	"Trey, Mike, Bryant"		1.5.1		
1.5.3	Configure Output	Output the character last input from the keyboard to the client.	Test data	5	"Trey, Bryant"				
2	Software Design								
2.1	Research Software	Research available opensource software for both text prediction and text to speech applications		14	Mike				
2.2	Software Planning								
2.2.1	Plan Text Prediction	Either plan a new text prediction program from scratch or plan how to reimplement existing opensource software.	Flowchart and program description for text prediction				2.1		

Schedule Gantt Chart

