Input Output modalities for visually challenged users. -Mini Design

Sanket Kamble

Unais Sait

Guided by

Pradipta Biswas

### Problem & List of Requirements

Difficulties faced by visually challenged person (completely blind) in using modern technologies (smart phones, computers ) in turn effects their daily routine

Requirements	Weightage		
Affordability	5		
Easier selection process	3.5		
Delay free interface	3		
Error Handling system	3		
Secure transaction system	2.5		
Haptic sensing units	1.5		
Navigation solutions	1.5		

# Objectives

- To make the selection process less sequential
- To provide delay free feedback
- Affordable technology
- Error Handling system

## Technologies



### Haptic Feedback

- Helps to realize the boundary of the screen while using mouse
- Delay free feedback

Mouse Cursor at X = 963, Y = 476 Cursor Position





# Haptic Feedback

• Not easy to distinguish vibration



Realization of intensity of vibration



### Haptic Feedback



Wearable Haptic Gloves

- Same technology
- Better sense of haptics
- Wearable

# Test Setup



## Results





#### Haptic sensation

### Gesture Recognition

- Two cameras and three infrared LEDs
- Track infrared light with a wavelength of 850 nanometers, which is outside the visible light spectrum.
- Capturing speed :100fps
- Volume captures 500x500x500mm3
- Resolution = 0.01mm
- Viewing range is limited to roughly 2.6 feet (80 cm) above the device



### Gesture Recognition Testing



- Positions of cursor is difficult to visualize
- Difficult to understand boundary

# Speech Recognition

#### **Current Option:**

- Google search
- Samsung S voice, BIXBY
- Siri
- Cortana

#### Concept :

Lancher with Speech Recognition.

#### **Observation:**

- Required to go to the speech recognizing app or
- Required to enable the option, within the OS-launcher

# Google Speech API

#### CLOUD SPEECH API

Speech to text conversion powered by machine learning





# Speech Recognition

(in a launcher)



# **Test Setup**



# Experiment

### Task: Select the App from the list



# Application



# Working

Catch Uncaught Exception and tell the system what to do. To implement this, the below steps were followed:-

- 1. Create a class extending Application Class
- 2. Handle uncaught Exception in your Application subclass.
- 3. In your launcher Activity, call your Application class.
- 4. After catching an Exception, start your main Activity (as per your requirement).

```
private void handleUncaughtException (Thread thread, Throwable e)
{
Intent intent = new Intent (getApplicationContext(), MainActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(intent);
}
```

# Application



- Takes 5-6 seconds to get back to the same activity
- Speech assistance to

# Comparison of three Technologies

Haptic sensation + Text To Speech

- Avoids sequential selection process
- Haptics allows the user to visualize the window faster

Speech recognition in Launcher + Text to Speech

Self resolving Application

- Avoids sequential selection process
- Affordable technology
- Operations are less
- Voice recognition with AI algorithm - google

- Error handling system
- Voice notification during errors

### Suitability of the Technology

Requirements	Delay free interface	Error Handling system	Secure transaction system	Haptic sensing units	Navigation solutions	Affordabi lity	Easier selection process	Total
Haptic sensation	1	3	1	4	2	3	3	17
Speech recognition	1	3	3	1	4	5	4	21
Application	1	5	3	1	2	5	2	19

\*Scale out of 5

# Thank You