3D PowerPoint

CSE379: Senior Project

Matthew Raporte • Greg Potter • Professor Xiaolei Huang

Why?

- Rapidly advancing AR, VR, and 3D input technology.
- However, commercially unsuccessful and not a “household” technology.
- We wanted to gain experience in this field and demonstrate a practical application of new technology.

What?

Used affordable 3D input technology to detect 6 hand gestures and translate them into MS PowerPoint actions, creating a hands-free and natural way to give PowerPoint presentations

How?

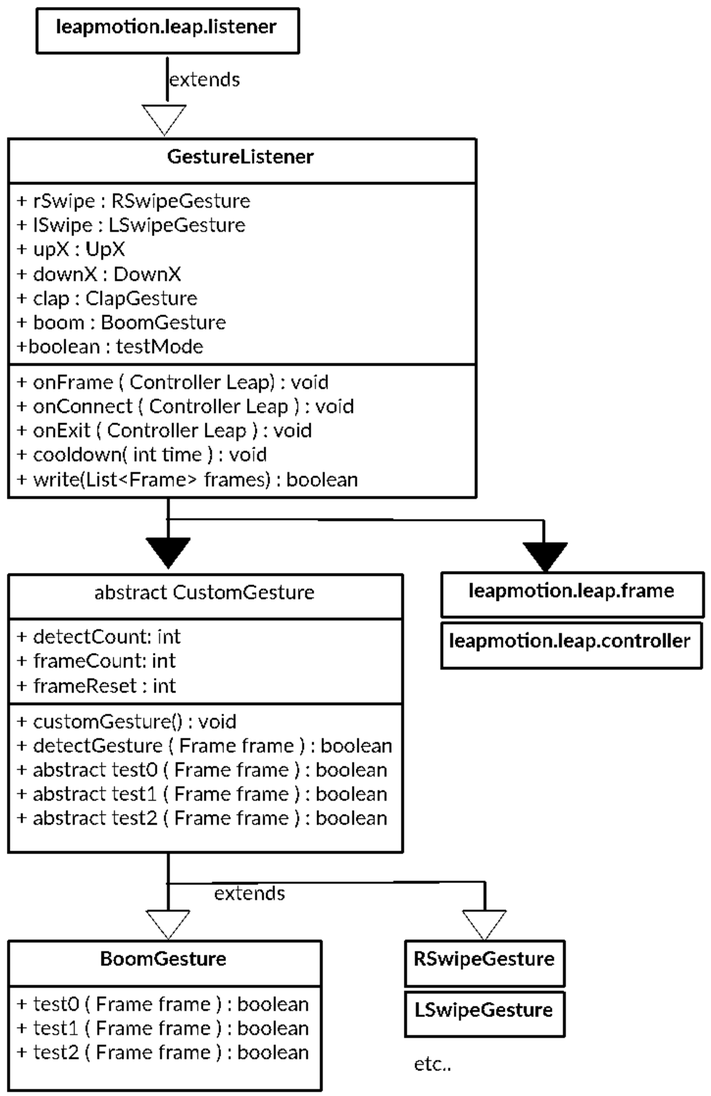
Split our project into 3 components:

① **Detection**
② **Translation**
③ **Action**

This allowed us to build, test, and optimize each piece independantly.

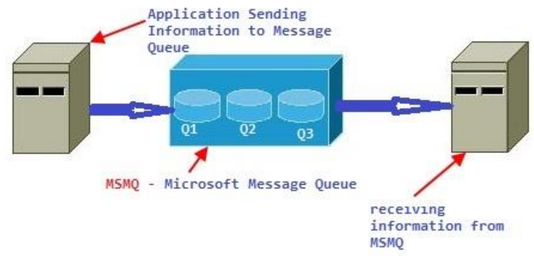
Detection

- Implements customized Listener object to test each detected Frame for gestures in prioritized order using detectGesture calls.
- detectGesture calls the gesture’s start, midpoint and end condition tests according to detectCount, and increments it when a test is passed.



- Each gesture overrides CustomGesture test methods and tests frame data to verify start, midpoint and endpoint conditions.

Translation



Uses MSMQ to relay commands from detection to action without interference.

Action

Standalone C# application controls PowerPoint using the PowerPoint Object Model with or without queue.

Final Product

Gesture	Action
Right Swipe	Next Slide
Left Swipe	Previous Slide
"X" Finger Swipe Up	Advance "X" slides
X Finger Swipe Down	Go back "X" slides
"Boom" Gesture	Activate Hyperlink/Media
Clap	Start/End presentation

Extensibility

- Our architecture allows for further expansion through extension and definition of new CustomGesture objects.
- Gestures may also be tested more robustly by adding more test methods to CustomGesture, and skipping over them by incrementing detectCount in simpler classes.
- Gestures can record select frame data to an Excel spreadsheet and calculate latency to facilitate development.