IEA

Intelligent Engineering Assistant

“AiYa! This is amazing~!” - IEA

Development Team

Ryan Kirkpatrick (Leader) - GUI Design and NLP

Dang Nguyen - Data Retrieval and Organization

Dali Xiao - Display Engine and Canvas Production

Min Jian Yang - Interpretation Engine and NLP

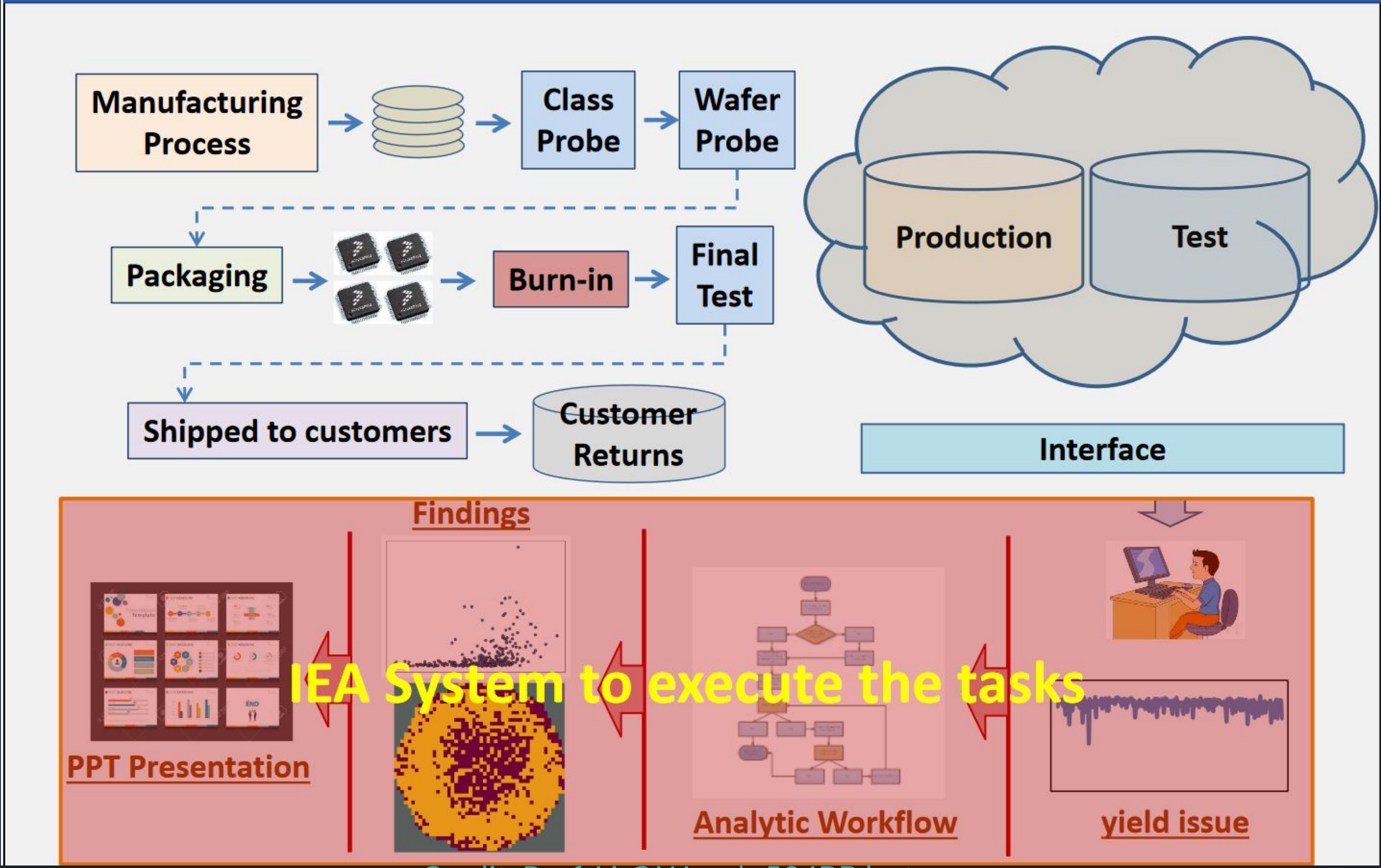
“Wow these guys are so cool” - IEA

Purpose

- Companies are allocating a lot of resources into the manufacturing process, especially hardware verification
- Engineers spend a lot of time looking over production data and trying to extrapolate meaningful results in order to improve yield
- This process is rather deterministic and can be automated
- This will help push products to the market faster as well as reduce company labor costs

“What’s the point?” - IEA

PRODUCTION LINE



Manufacturing Process

Class Probe

Wafer Probe

Packaging

Burn-in

Final Test

Shipped to customers

Customer Returns

Production

Test

Interface

Findings

PPT Presentation

Analytic Workflow

yield issue

IEA System to execute the tasks

Credit: Prof. Li-C Wang’s 594BB lecture

Product Description

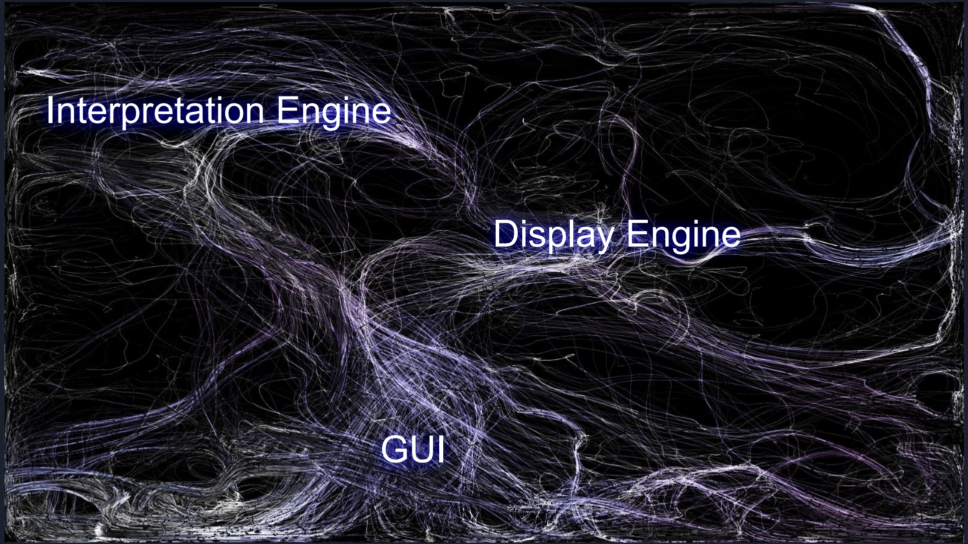
- Querying System
  - Interacts with the user through audible AI
  - Retrieves and caches data for queries
  - Understands the context of what data is available and how to conversation is developing
  - Displays essential data through charts or graphs

“

“What I do~” - IEA

Control & Threading

- The querying system consists of a GUI, an interpretation engine and a display engine.
- To ensure these subsystems run smoothly, a control system is used to handle signaling between subsystems.
- Signalling lets subsystems communicate across threads, and threading will be done on a functional basis.
- Threading helps with:
  - avoiding interference, running in-sync, scheduling processes
- Signal using pyqtSignals and pyqtSlots

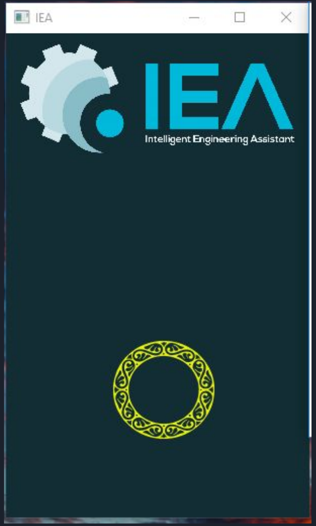


Error: The program has stopped responding...

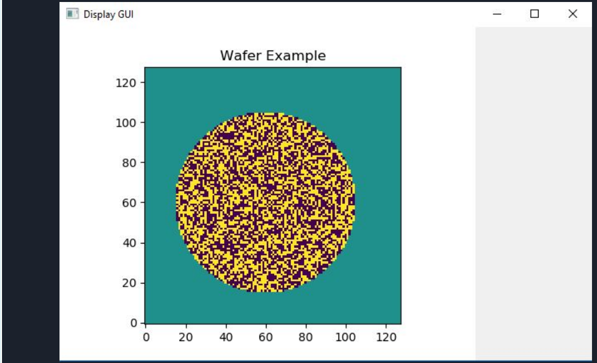
GUI

- **Main GUI:**
  - IEA Response: A textual response from IEA that reads the same as the audio output of the interpretation engine
  - Button to tell interpretation engine to start listening to the user
- **Display GUI:**
  - Display window for showing matplotlib graphs generated by the display engine
  - Excess area for labels to be given from display engine
  - Understanding of plot type, and ways to represent different structures/ display objects

Main UI:



Display UI:

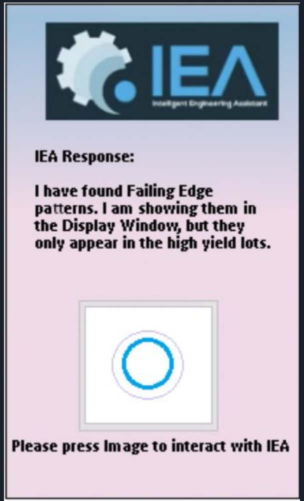


“Does this GUI make me look fat?” - IEA

GUI - Spec

- Input:
  - From User:
    - Clicks to toggle audio
  - From Interpretation Engine:
    - Response Text
  - From Display Engine:
    - Display data and format type
- Output:
  - To User:
    - Displayed response text and graphs
    - Final output of PPT Presentation
  - To Interpretation Engine:
    - Signal to start listening to user
- What's Next:
  - More interaction on the display UI and more options for charts and graphs

Main UI:

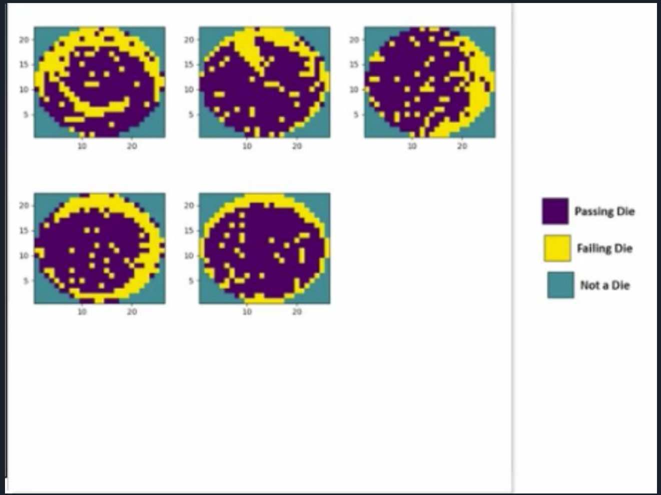


IEA Response:

I have found Failing Edge patterns. I am showing them in the Display Window, but they only appear in the high yield lots.

Please press Image to interact with IEA

Display UI:



"Does this GUI make me look fat?" - IEA

Interpretation Engine

- Queries the user by using Amazon Web Service's Text-to-Speech API to convert text string to audio
- Uses Google Speech-to-Text API to convert user's query into text string
- Passes the text string into an Natural Language Processor to extract semantic meaning
  - Part of speech (POS) parser, state machine, NLP algorithm (LSTM, GRU, etc...)
- Maps the semantic meaning to corresponding function and pass it into the Display Engine

"Sorry, could you say that again?" - IEA

Interpretation Engine - Spec

- Stage 1: Keyword Identifier
  - Mapping user command to specific keywords -> display engine commands
  - Small finite state machine keep track of the context

- Stage 2: Part-of-Speech
  - Extract semantic meaning by breaking the sentence up with POS parser (features)
  - Combine the POS parser with keyword identifier and distance measure
  - Complex state machine with more descriptive states

- Stage 3: To be determined...
  - Insufficient data to train right now
  - Neural Network?
  - Hybrid Network?

"Sorry, could you say that again?" - IEA

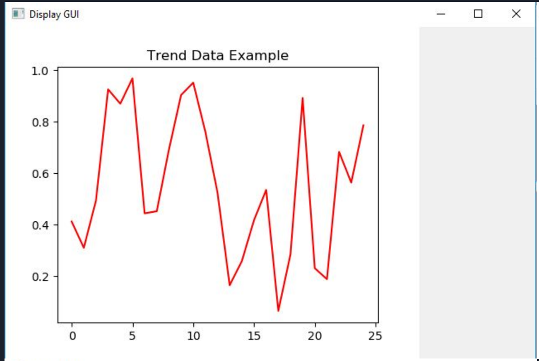
Display Engine

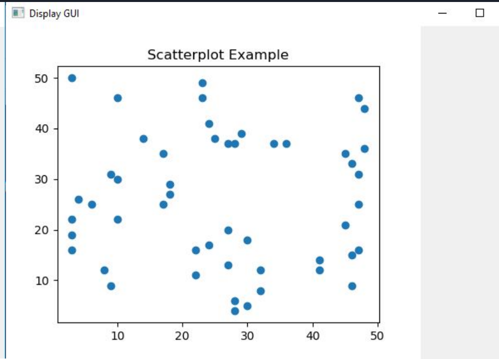
- Depending on the user's query, IEA will try to output a graph that is appropriate for the data

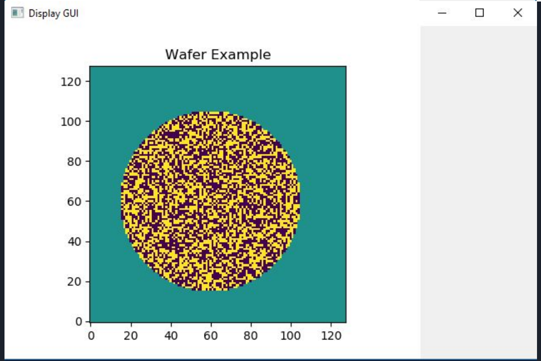
- The query has specific user intents that can be mapped to display functions

- External data is initially cached, then used for lookup

- Different kinds of graphs can be supported
  - Trend, Scatterplots, Wafer plots, etc.
  - Additional graphs can be implemented



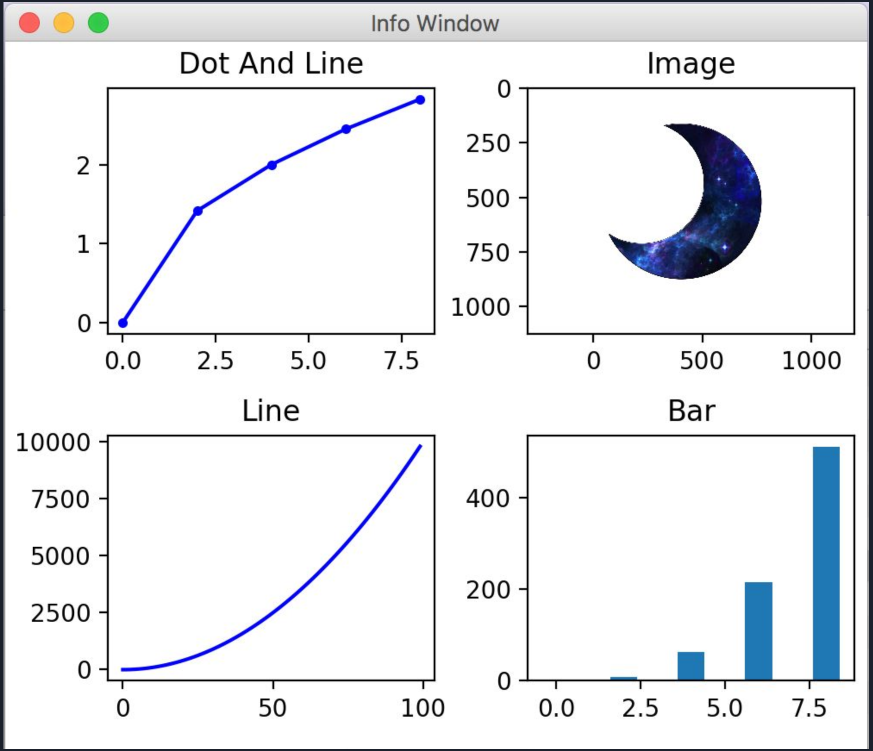




"Here are some graphs displaying useless data." - IEA

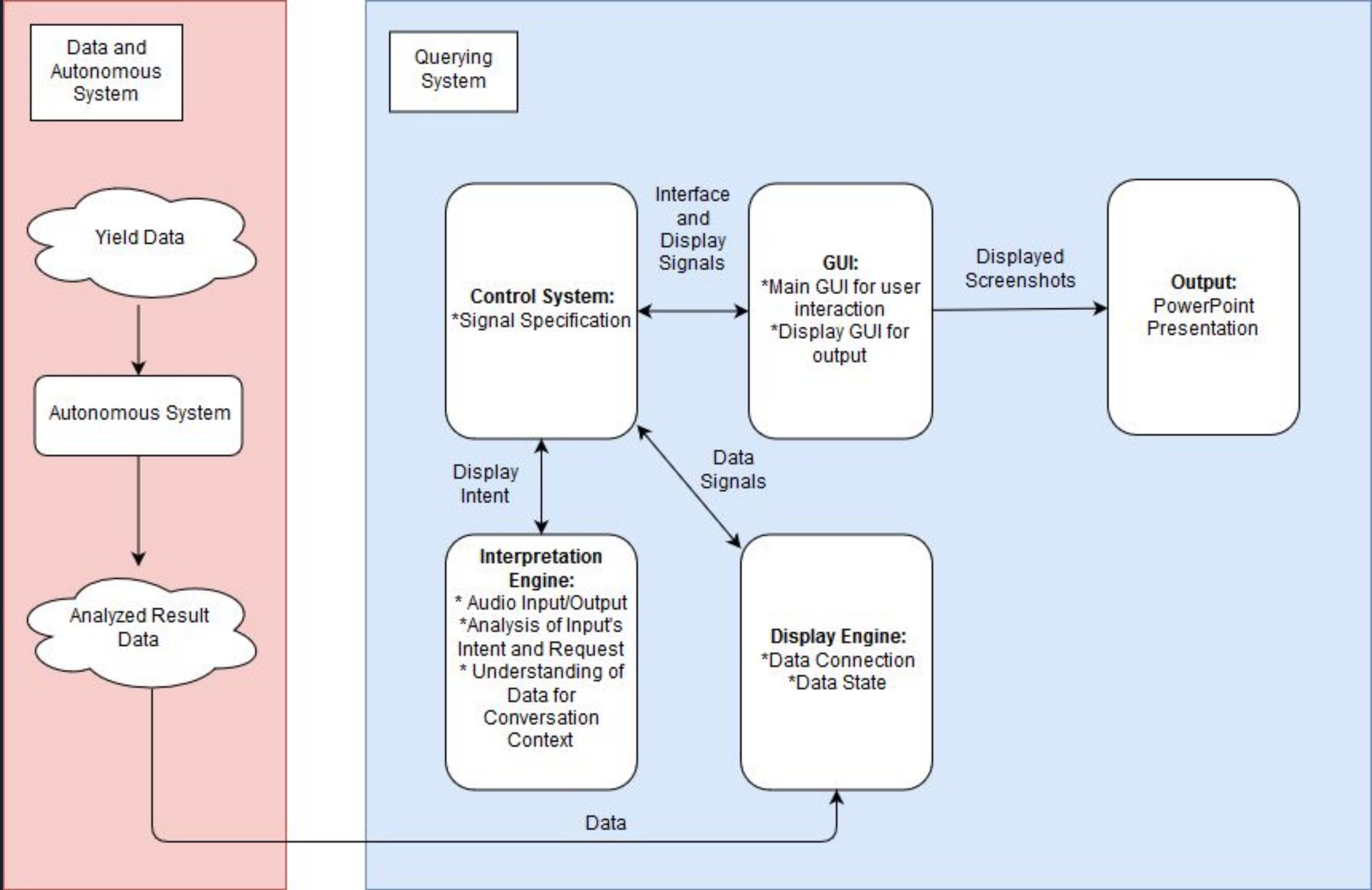
Display Engine - Spec

- Graphs have corresponding signals linked to Interpretation engine
- Keeps track of current intent / state
- Output - organized data for specific graph to GUI
- What's Next:
  - More charts and graph types
  - Actually interact with real data
  - More threading to prevent hanging



"Here are some graphs displaying useless data." - IEA

BLOCK DIAGRAM



Data and Autonomous System

Yield Data

Autonomous System

Analyzed Result Data

Data

Querying System

Control System:
*Signal Specification

Interface and Display Signals

GUI:
*Main GUI for user interaction
*Display GUI for output

Displayed Screenshots

Output:
PowerPoint Presentation

Display Intent

Data Signals

Interpretation Engine:
* Audio Input/Output
*Analysis of Input's Intent and Request
* Understanding of Data for Conversation Context

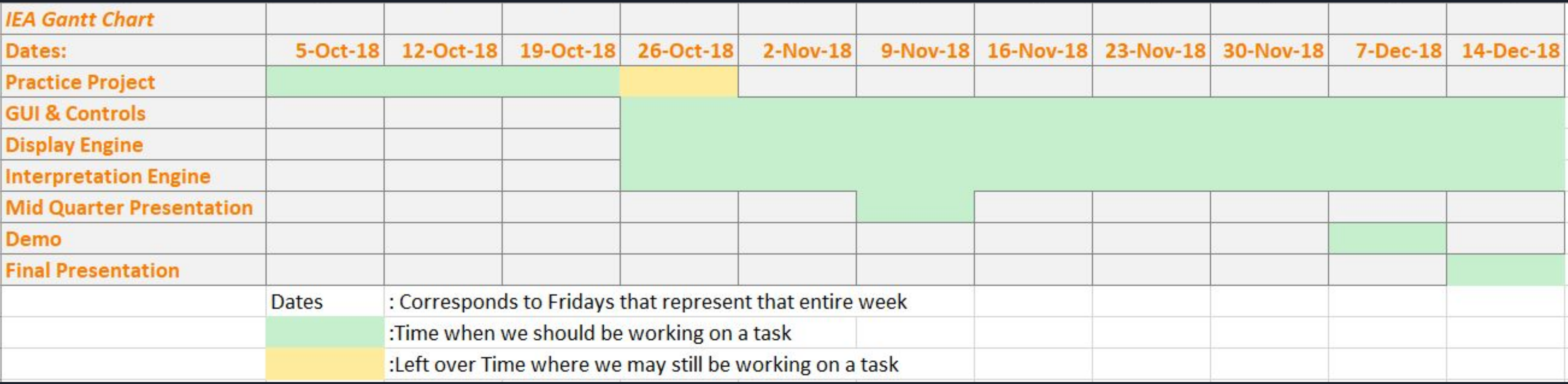
Display Engine:
*Data Connection
*Data State

Costs?

$0 !

Just kidding… We might need microphones to synchronize testing of audio input across different computers and reduce noise

Current Progress



IEA Gantt Chart											
Dates:	5-Oct-18	12-Oct-18	19-Oct-18	26-Oct-18	2-Nov-18	9-Nov-18	16-Nov-18	23-Nov-18	30-Nov-18	7-Dec-18	14-Dec-18
Practice Project	🟩	🟩	🟩	🟨							
GUI & Controls				🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Display Engine				🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Interpretation Engine				🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Mid Quarter Presentation						🟩					
Demo										🟩	
Final Presentation											🟩
	Dates	: Corresponds to Fridays that represent that entire week									
	🟩	:Time when we should be working on a task									
	🟨	:Left over Time where we may still be working on a task									

Goals

Features we want to implement moving forward:

- Additional graph support
- Passive listening
  - the program will know when to listen to the user based on conversation context
- Ability to export session data
  - Powerpoints, PDF, etc.
- Robust NLP so that the user can hold a natural conversation
- Generalize querying system (probably not happening)

THANK YOU !!!

Prof. Yoga

Prof. Li-C Wang

Chuanhe (Jay) Shan

Yueling (Jenny) Zeng

Thank you!

Any Questions?