FALL 2016

MAESTRO

Project Update 3

October 11, 2016

Charu Dwivedi
Da Fang
Wil Kacsur
Emily Kirven
Fidelia Lam
Nilay Muchhala
J. Nick Smith
Daphna Raz

SCHOOL OF MUSIC, THEATRE & DANCE
UNIVERSITY OF MICHIGAN

MULTIDISCIPLINARY DESIGN PROGRAM

Presentation Overview

Project Introduction
Design Description and Subsystem Integration
Validation Methodology and Preliminary Results
Project Plan and Management

PROJECT INTRODUCTION

Context

- Beginning conductor class pedagogy
- Curriculum based on Michael Haithcock
- Problem: Cannot practice without live musicians
- Maestro 1.0

Stakeholder Objectives

Maestro 2.0 will assist beginning conductors in the following ways:

- Train the body in principled movement
- Understand and utilize gestural tools for communication
- Develop and reinforce basic conducting techniques
- Help students grow as a musician and ensemble leader
- Help define 5 types of articulations: standard, staccato, legato, marcato, and tenuto
- Delineate between various dynamics from piano to forte

Data Collection

Gathered in Summer

- -13 Conductors:
  - -Standard, Staccato, Marcato, Legato, Tenuto gestures
- -Kinect Recorder Application

Data Processing

- -MDP Conducting Team labeled “action points”
- -Good vs. bad data classification

VALIDATION METHODOLOGY & PRELIMINARY RESULTS

Validation Overview

UI/UX validation

Algorithm validation

End-to-end validation

UI/UX Validation

Engineering Requirement and Units	Validation Method Title	Type of Method	Detailed Information
Qualitative method	Focus Group Assessment	Student Developed	Page 4
Mean satisfaction score of > 4	Stakeholder satisfaction survey (UX/UI and sound synthesis)- 5 point Likert scale	Student Developed	Page 5

UI/UX Validation: Focus Group Assessment

Focus Group Assessment
Student Developed Protocol

Equipment:

- xBox Kinect v2 with USB adaptor
- Laptop with Maestro Application

Protocol

- Ask group to complete specific tasks (login, select mode, etc.)
- Play samples of sound synthesis
- Follow up with round-table discussion

UI/UX Validation: Focus Group Assessment cont.

Focus Group Assessment
Student Developed Protocol

Status: In-Progress
Sponsor has approved the method

Timeline:
Initial group completed by October 14
Secondary group by October 28
Estimated duration: 2 hours

UI/UX Validation: Focus Group Assessment cont.

Focus Group Assessment
Student Developed Protocol

Impact of Failure:

System will have to be redesigned as suggested by stakeholders. Window of one month available for redesign.

UI/UX Validation: User Satisfaction Survey

User Satisfaction Survey

(part of the secondary focus group mentioned previously)

Student Developed Protocol

Equipment:

- xBox Kinect v2 with USB adaptor
- Laptop with Maestro Application
- Printed Survey (based on 5-point Likert scale)

Protocol

- Sample of 8 students
- Ask students to complete specific tasks
- Play examples of sound synthesis
- Follow up with formal Likert-scale survey

UI/UX Validation: User Satisfaction Survey cont.

User Satisfaction Survey
Student Developed Protocol

Analysis of Results

Evaluate survey responses
Pass: 80% of participants rate each question > 4

Status: In-Progress

Sponsor has approved the method

Timeline:

To be completed by October 28
Estimated duration: 90 minutes

UI/UX Validation: User Satisfaction Survey cont.

User Satisfaction Survey
Student Developed Protocol

Impact of Failure:

Redesign the system as suggested by Likert scores. If Focus group is not delayed, there is a window of three weeks to make changes.

Algorithm Validation

Engineering Requirement and Units	Validation Method Title	Type of Method	Detailed Information
90% detection rate of Action Point	Action Point Detection	Student Developed	Page 7
Accurate detection rate of Action Point within 3 point window	Action Point Accuracy	Student Developed	Page 8
85% of Gestures correctly predicted	Gesture Detection	Student Developed	Page 9

Algorithm Validation

Action Point Detection

Student Developed Standard - Requirement: 90% of Action Points Detected

Apparatus

Microsoft Kinect - for recording input data

Python Testbench - for automatically running our Algorithm on multiple files

Procedure

Record many datasets from multiple conductors on each of the five types of gestures

Run the latest copy of the Algorithm (translated into Python) against all recorded data through automated test bench

Determine if a sound was produced (i.e. was the action point caught)

Status: In-Progress

Sponsor has approved the method

Impact of Failure:

Project goals will not have been met this semester

Goal will have to be passed on to next semester’s team

Algorithm Validation

Action Point Accuracy

Student Developed Standard - Requirement: Action Points Accurate w/i 3 point window

Apparatus

Microsoft Kinect - for recording input data

Python Regression Framework - for automated regression testing

Procedure

Record many datasets from multiple conductors on each of the five types of gestures

Run the latest copy of the Algorithm (translated into Python) against all recorded data through regression testing framework

View output analytics (% of action points caught, standard deviation, etc)

Status: In-Progress

Sponsor has approved the method

Impact of Failure:

Project goals will not have been met this semester

Goal will have to be passed on to next semester's team

Algorithm Validation

Gesture Detection

Student Developed Standard - Requirement: 85% of Gestures correctly predicted

Apparatus

Maestro Gesture Recognition Analytics Tool

Recorded Conducting Gesture Data

Gesture Recognition Algorithm

Procedure

Load Gesture Recognition Algorithm into Analytics Tool

Run Analytics Tool

Observe analytics generated by the Analytics Tool

Status: In-Progress

Sponsor has approved the method

Waiting to reach previous validation methodology goals before beginning with this one

Impact of Failure:

Project goals will not have been met this semester

Goal will have to be passed on to next semester’s team

End to End Validation

Engineering Requirement and Units	Validation Method Title	Type of Method	Detailed Information
UI and Feedback Functions	Horizontal Testing	Student Developed	Page 10
Feedback of Different Conditions Obtained	Visual and Sound Feedback Test	Student Developed	Page 11
User and Expert Satisfaction	Experts and Stakeholder Satisfaction	Student Developed	Page 12

End to End Validation: UI and Feedback Functions

Status: In Future

Equipment:

xBox Kinect v2.0
Laptop with Maestro application

Impact of Failure:

Check hardware connection and surrounding environment
Debug UI design flow

Protocol:

Select each mode and assignment and return to the main menu
Make sure the UI and the Sound Synthesis Subsystem functions
Check the feedback latency for each assignment

End to End Validation: Feedback of Different Conditions Obtained

Status: In Future

Equipment:

xBox Kinect v2.0
Laptop with Maestro application

Impact of Failure:

Check integration of the sound synthesis module and tracking algorithm
Debug UI visual feedback

Protocol:

Trigger each designed visual and sound feedback response
Make sure each feedback response behaves as expected
Record the latency for each feedback response

End to End Validation: User and Expert Satisfaction

Status: In Future

Equipment:

xBox Kinect v2.0
Laptop with Maestro application

Impact of Failure:

Debug the integrated gesture tracking algorithm
Check for the UI friendliness

Protocol:

Sample 8 beginning conducting students
Ask them to use the software themselves after necessary instruction
(Expert) Evaluate system's feedback and latency to gestures

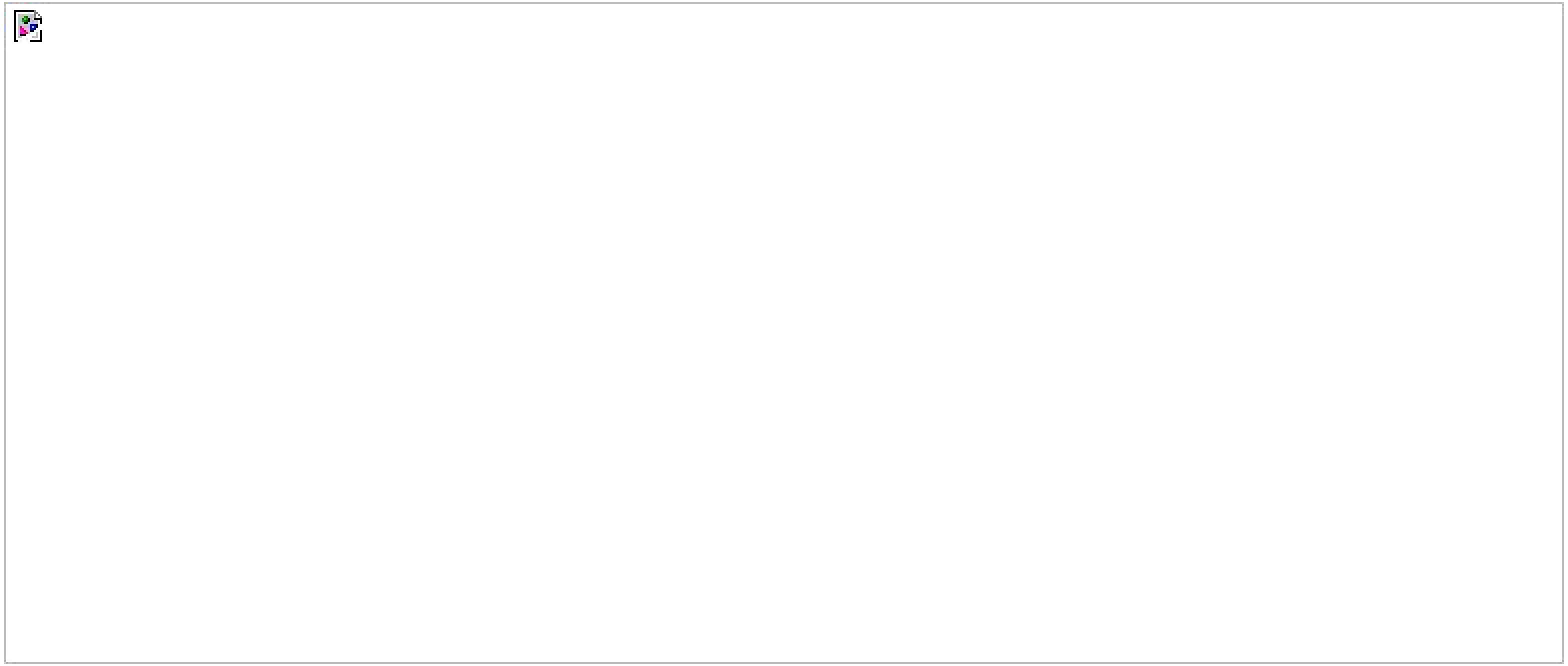
DESIGN DESCRIPTION AND SUBSYSTEM INTEGRATION

Holistic Overview

End-to-End System has 4 parts:

User Interface Input
Algorithmic Backend
Sound Synthesis
User Interface Output

Software Flow



Subsystem Design Description

Gesture analysis on recorded files (python)

Real time gesture tracking algorithms (C#)

UI (C#)

Sound synthesis system (MAX)

System Integration: Current Status

Most current version of gesture tracking algorithm in python

Multiple benefits for developers working in python

- Speed and ease of development
- Open source tools for analytics
- Open source tools for Machine learning
- Potential for cross platform functionality

System Integration: Current Status

Second most current version of algorithm is always in C# and integrated into project

TO DO: standardize the transition from python to C# for better workflow

- Function definitions, variable names
- C# vs python syntax (class declaration, list structures)

Gesture Analysis (Python)

Uses data acquired by conducting team over summer

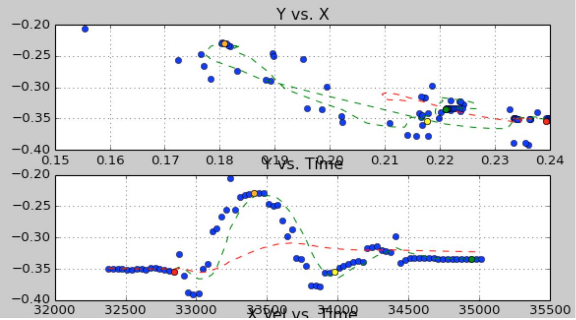
Allows us to visually represent the data we acquire

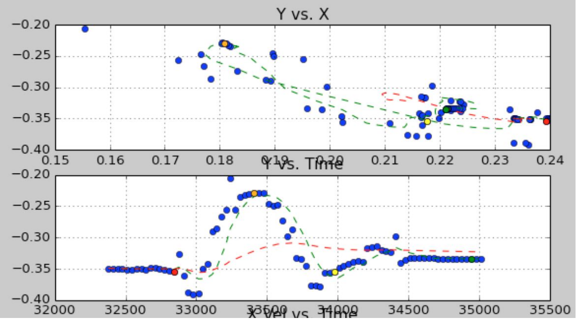
Simulators run our algorithm on all of our data files in python

Simulator compares algorithm prediction of action point to action points that our conductors hand-selected

Returns a text file of analytics that record:

- Accuracy of action point detection
- Percentages of action points found
- Percentages of accurate classifications





Easy to add more analytics because of data format

- JSON Format

Gesture Algorithm (C#)

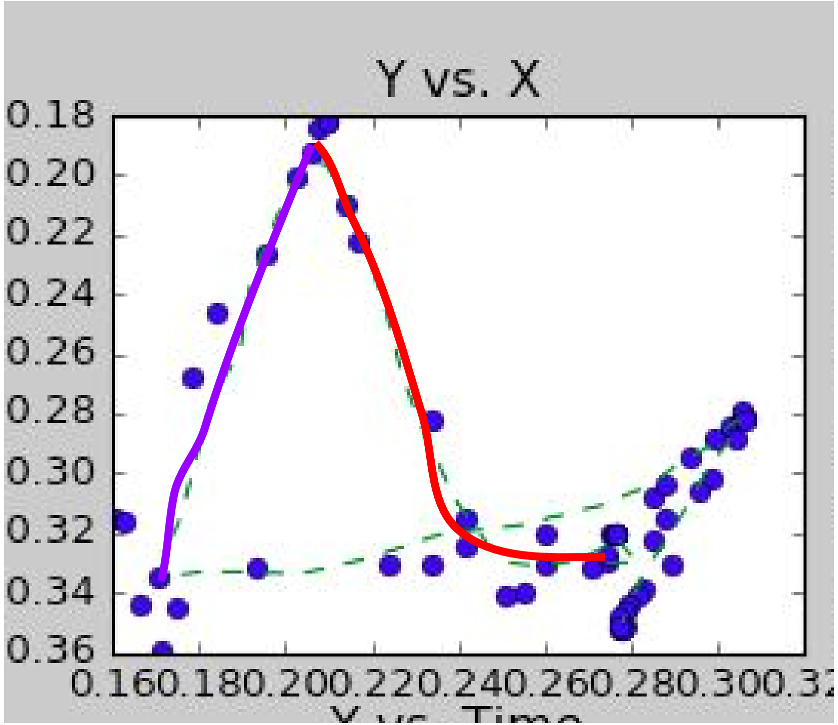
Math based analysis of 2D Coordinate System

Algorithm is based off of three critical points:
Start Point, High Point, and Action Point

Two important “legs” of gesture (purple and red line):
[start → high] and [high → action]

Speed and relative distance covered will dictate which gesture is predicted

Met with conductors to come up with formula that maps the speed and relative distance of these legs to different classifications of gestures



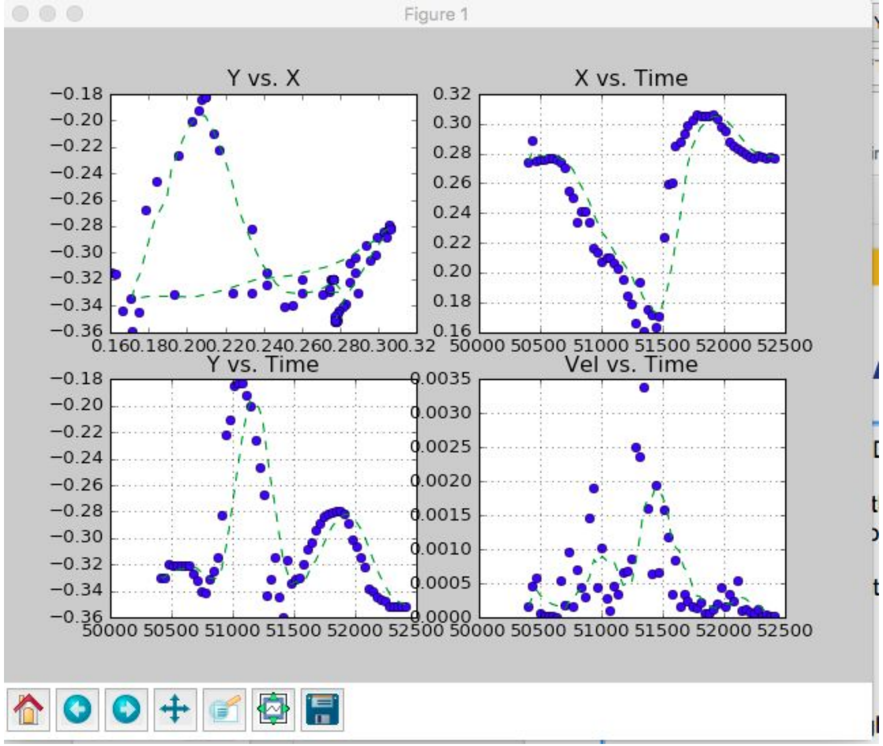
Gesture Algorithm cont

Smoothing Algorithms

Dotted lines in plot showcase our smoothing algorithms

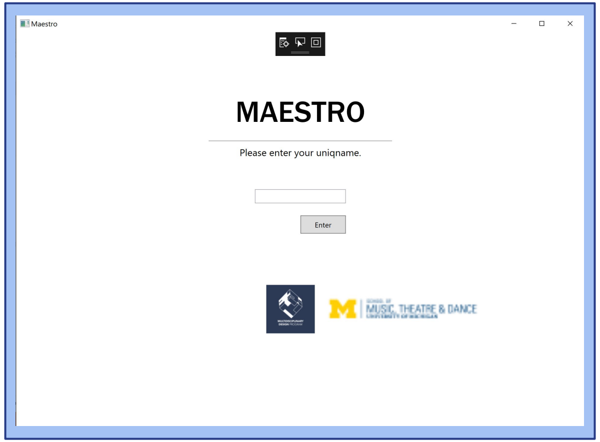
Use moving averages to smooth out data in order to get accurate results

Allows us to work with clean data and define the “legs” of the gestures better

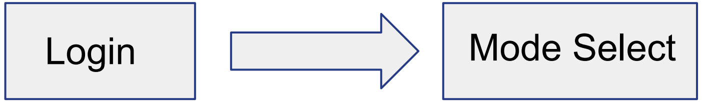


UI use case

Login

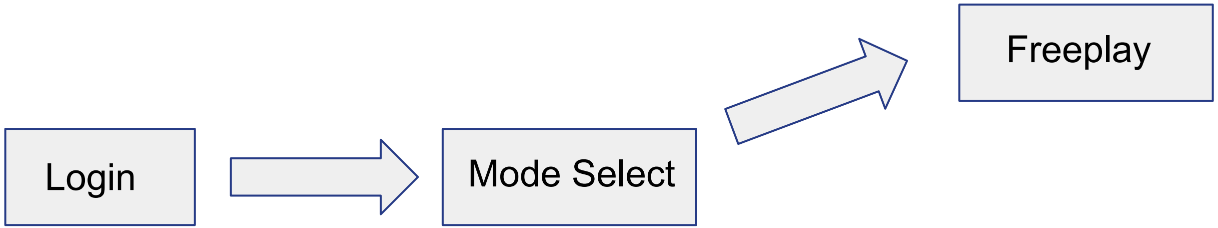


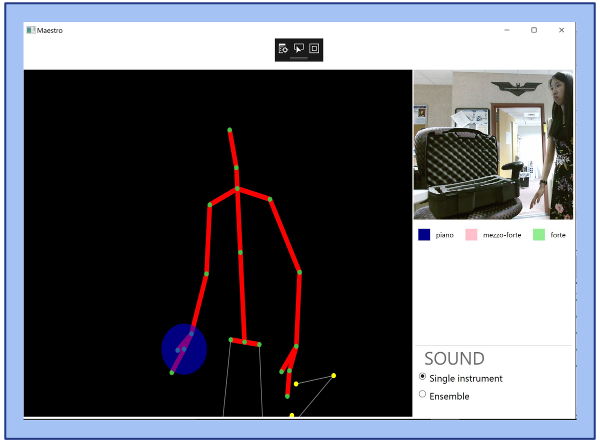
UI use case


Login
Mode Select

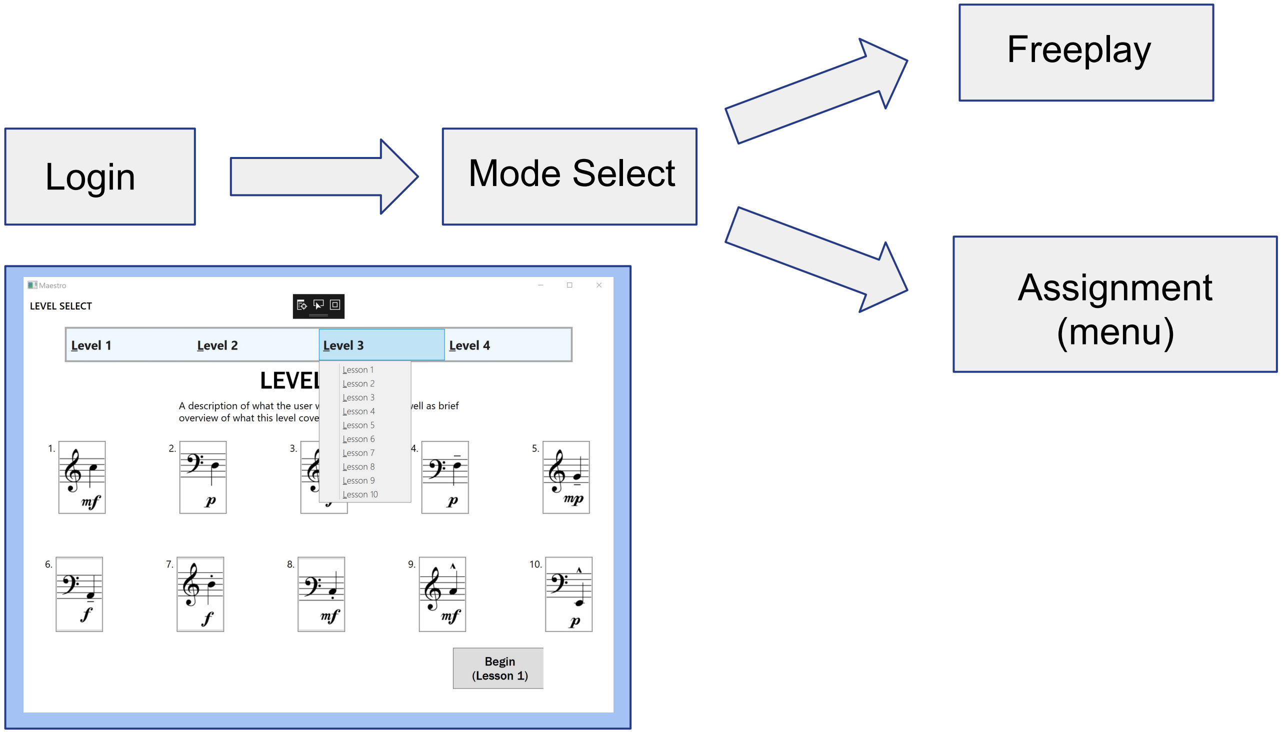

Maestro
MAESTRO
Mode Select
Freeplay
Assignment

UI use case

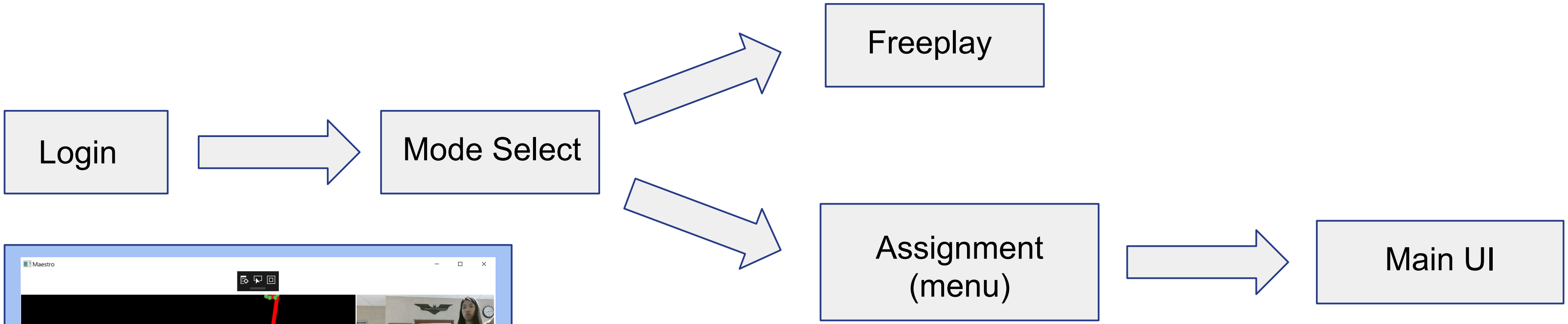

Login
Mode Select
Freeplay


Maestro
piano
mezzo-forte
forte
SOUND
Single instrument
Ensemble

UI use case

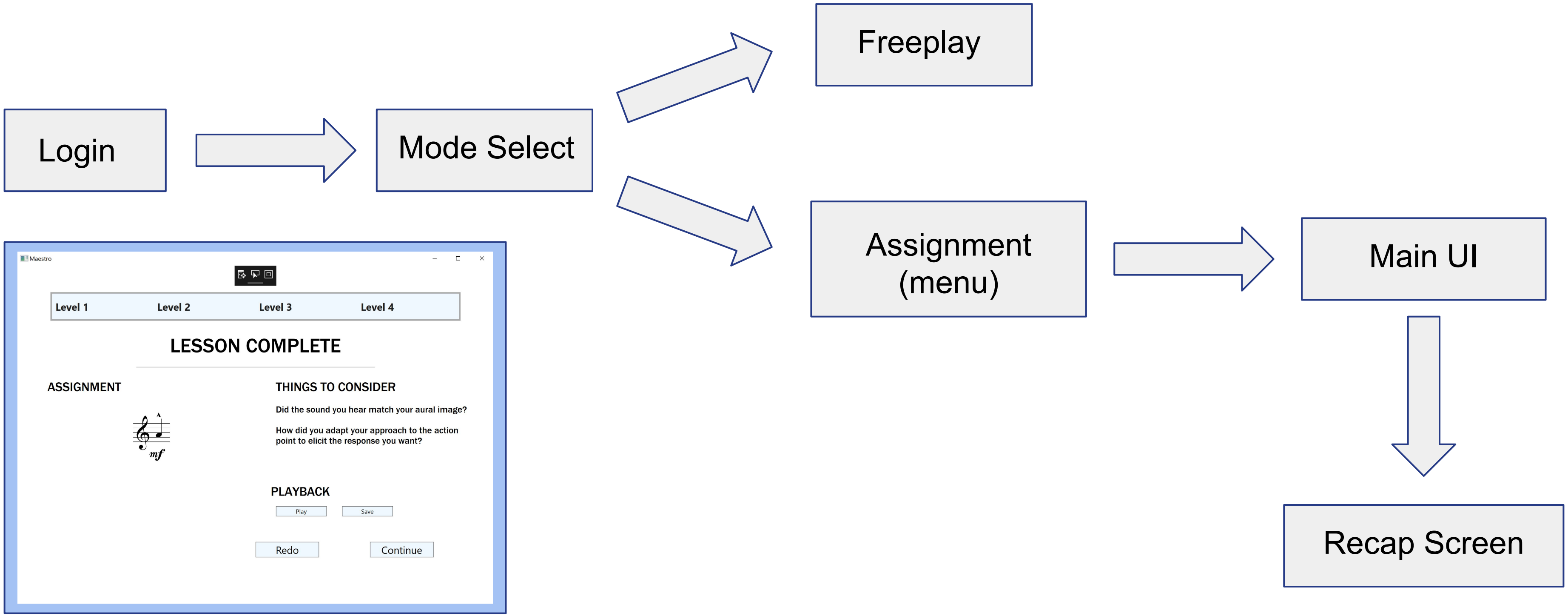

Login
Mode Select
Freeplay
Assignment (menu)
Maestro
LEVEL SELECT
Level 1
Level 2
Level 3
Level 4
Lesson 1
Lesson 2
Lesson 3
Lesson 4
Lesson 5
Lesson 6
Lesson 7
Lesson 8
Lesson 9
Lesson 10
LEVEL
A description of what the user w ... ell as brief
overview of what this level cove
Begin
(Lesson 1)

UI use case

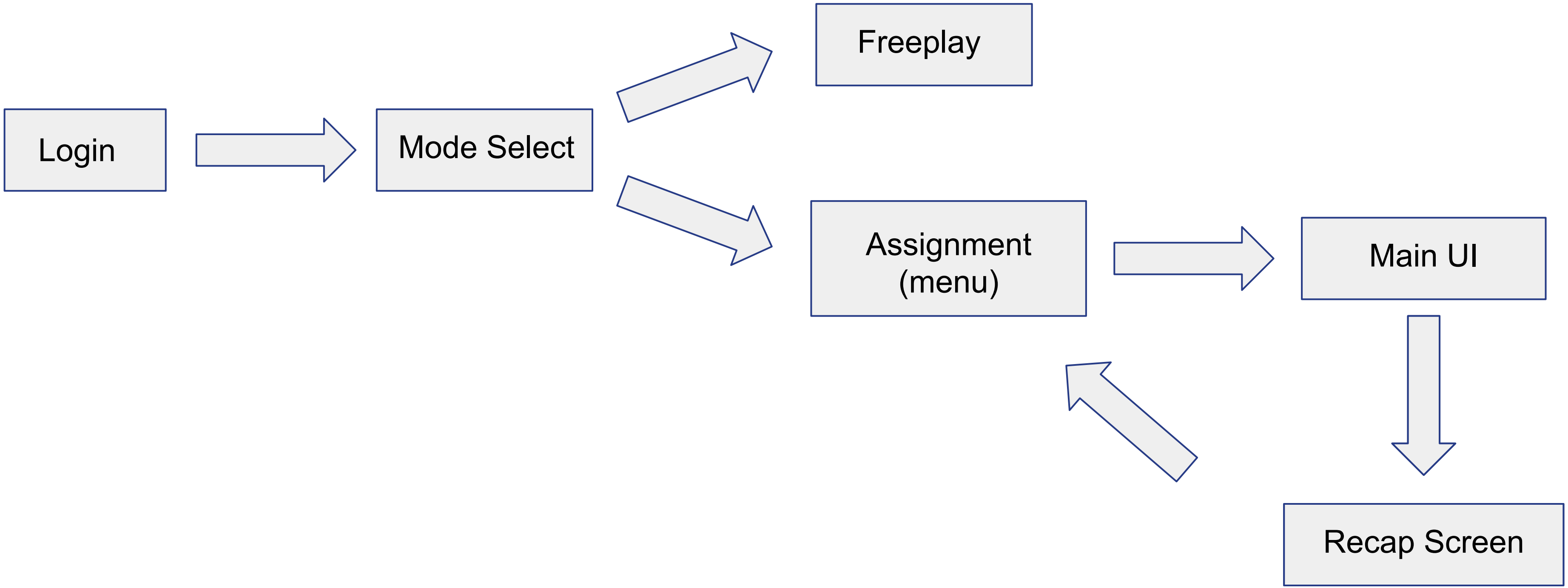

Login
Mode Select
Freeplay
Assignment
(menu)
Main UI


Maestro
piano
mezzo-forte
forte
SOUND
Single instrument
Ensemble
Kinect not available!

UI use case


Login
Mode Select
Freeplay
Assignment (menu)
Main UI
Recap Screen
Maestro
Level 1
Level 2
Level 3
Level 4
LESSON COMPLETE
ASSIGNMENT
THINGS TO CONSIDER
Did the sound you hear match your aural image?
How did you adapt your approach to the action point to elicit the response you want?
PLAYBACK
Play
Save
Redo
Continue

UI use case


Login
Mode Select
Freeplay
Assignment
(menu)
Main UI
Recap Screen

Sound Synthesis

Module based system in Max/MSP

OSC communication protocol

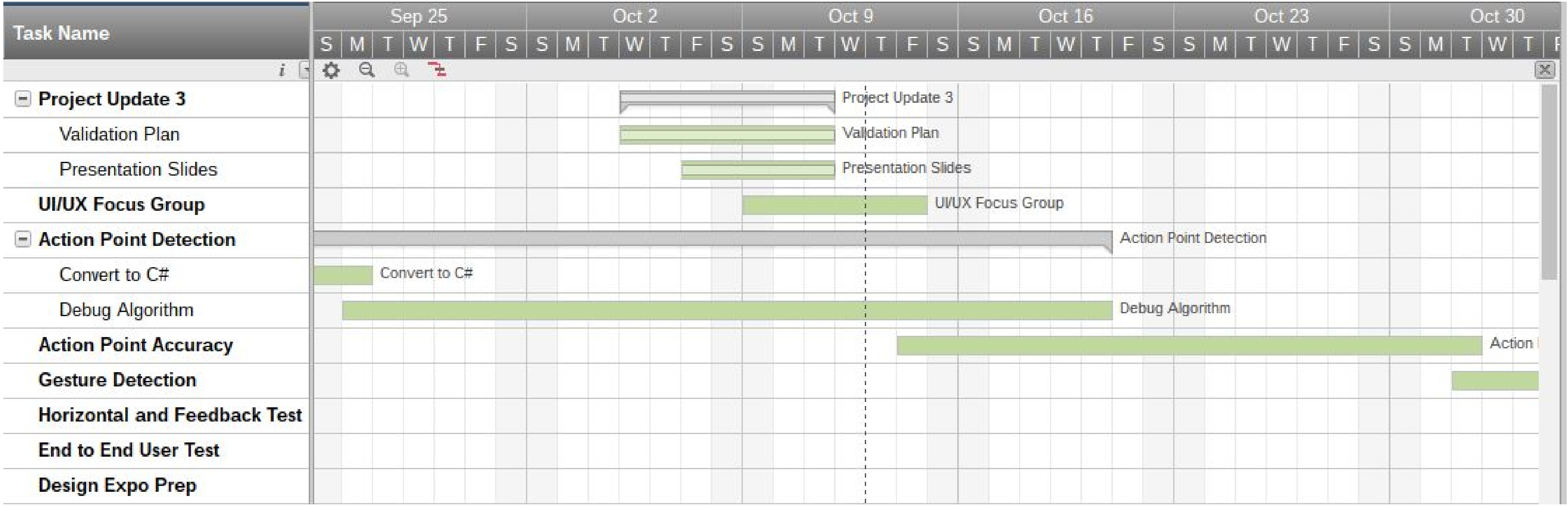
Main sound engine takes arguments for:

- Note frequency
- Attack type
- Dynamic level
- Duration

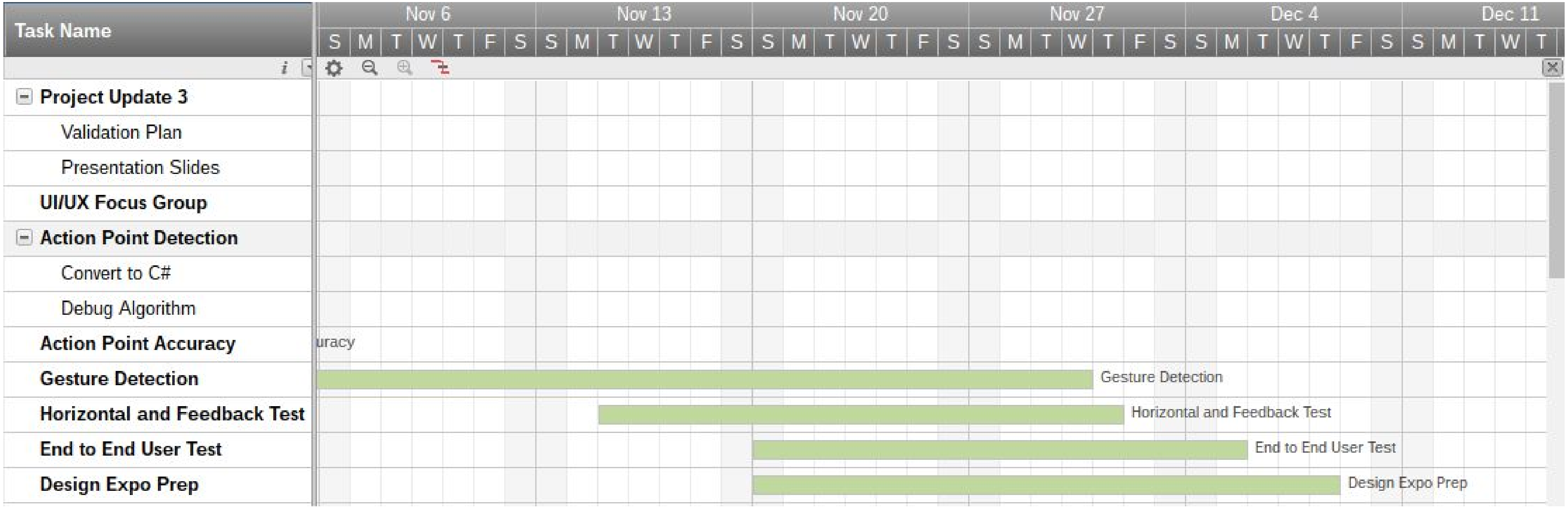
Options for solo instrument or ensemble

PROJECT PLAN & MANAGEMENT

Near Term Plans



Gantt Chart Cont.



Timeline

UI/UX Focus Group Assessment - October 14
Action Point Detection - October 20
User Satisfaction Survey - October 28
Action Point Accuracy - November 1
Gesture Detection - November 30
Horizontal and Feedback Test - December 1
End to End User Test - December 5
Design Expo - December 8

Likely Problems

Biggest Concern:

Failing to to reach Algorithmic Validation Requirements

Case 1: Action Point Accuracy is < 90% for actual test subjects

Unlikely to occur

Case 2: Gesture Prediction is < 85% for actual test subjects

Contingency Plan

- Failure to reach 90% action point detection rate
  - Cohesive groundwork laid for future work

- Failure to reach 85% gesture prediction rate
  - Action point detection rate will have been met
  - Begin laying groundwork for a fully cohesive next iteration
    - Dynamic control
    - Pattern detection
    - Extra time spent documenting

Q & A

Maestro 2.0 baseline specifications

System that allows users to shape a single sound

Properties we can change:

- Dynamics
- Articulation
- Length

External Constraints

- Each conductor has their own style
- Signal delivery + processing takes time

Stakeholder Requirement	Relative Priority	Specification	Measurement Methodology
Accurately detect beginning, middle, and end of gesture	1	Success rate of 80% or higher AND system is in agreement with expert opinion	Calculate success rate of each part of gesture based on multiple tests using Emily and Nick as sample. Will have a minimum of X gestures (in discussion with sponsor-Feb. 3, 2016)
Accurately detect across subjects	1	Success rate of 80% or higher AND system is in agreement with expert opinion	Calculate success rate of detection based on a sample consisting of Dr. Brown's COND 315 students
Informative audio feedback based on how gesture was executed by student	2	System response time of 30ms or less on average	Run multiple tests of our device using Emily and Nick and time the audio feedback lag using a timer function
Attractive audio feedback mapped to gestures	2	At least 75% respond with "attractive"	Survey Dr. Brown's COND 315 class: attractive / not attractive
Intuitive User Interface (UI) for all	3	Average of 3.5 on Likert scale	Survey Dr. Brown's COND 315 class using a Likert scale of 1-5 on intuitiveness and ease of use of UI