Techniques for Dynamic Dialog, Voice, and Lip Sync for Character Encounters

James Tiller, Robert Hubal
UNC Eshelman School of Pharmacy

East Coast Gaming Conference 18 April 2018

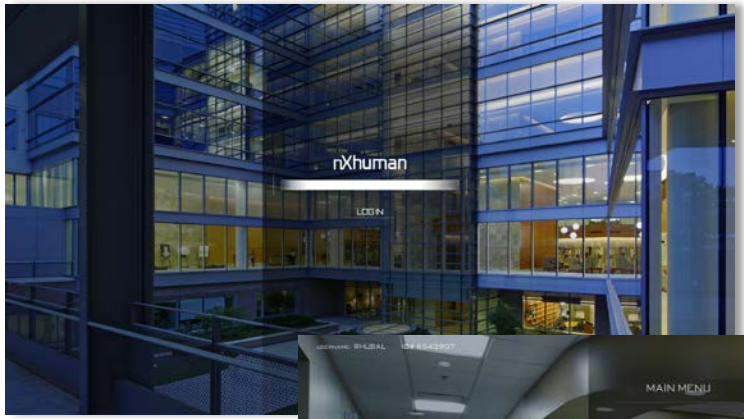
Robert Hubal

- Research interests center on the **intelligent use of technology** (e.g., simulation, natural language, sensors) to better teach and assess complex knowledge and evolving skills
  - Developing increasingly realistic virtual patients
  - Studying cost-effective methods for teaching and assessment of technical as well as sociocognitive skills
  - Studying cost-effective methods for improving technical and sociocognitive skills within clinical practice

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

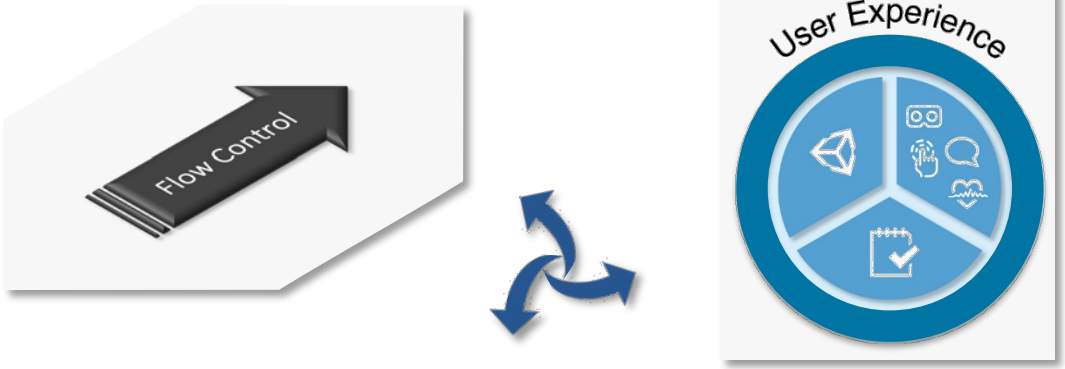
nXhuman Project Purpose

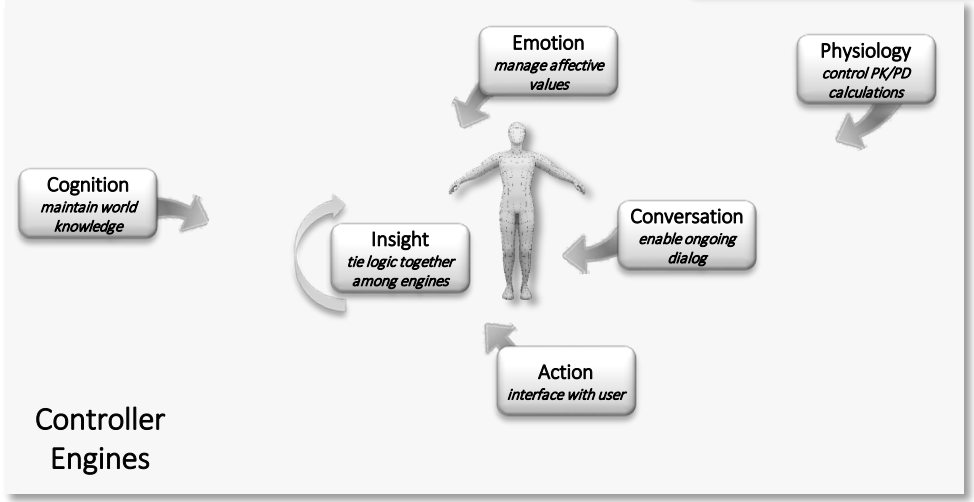
- Repeated practice in clinical decision making
- Prepare students prior to seeing first patients
- Exercise ‘process of care’



- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

nXhuman Underlying Components





- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

James Tiller

- Lead Software Engineer
- Previously: Network Engineer

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Overview

- Core components
  - Dynamic dialog
  - Procedural lip sync
- Sub-topics
  - Speech-to-text
  - Voice cloning
  - Gestures

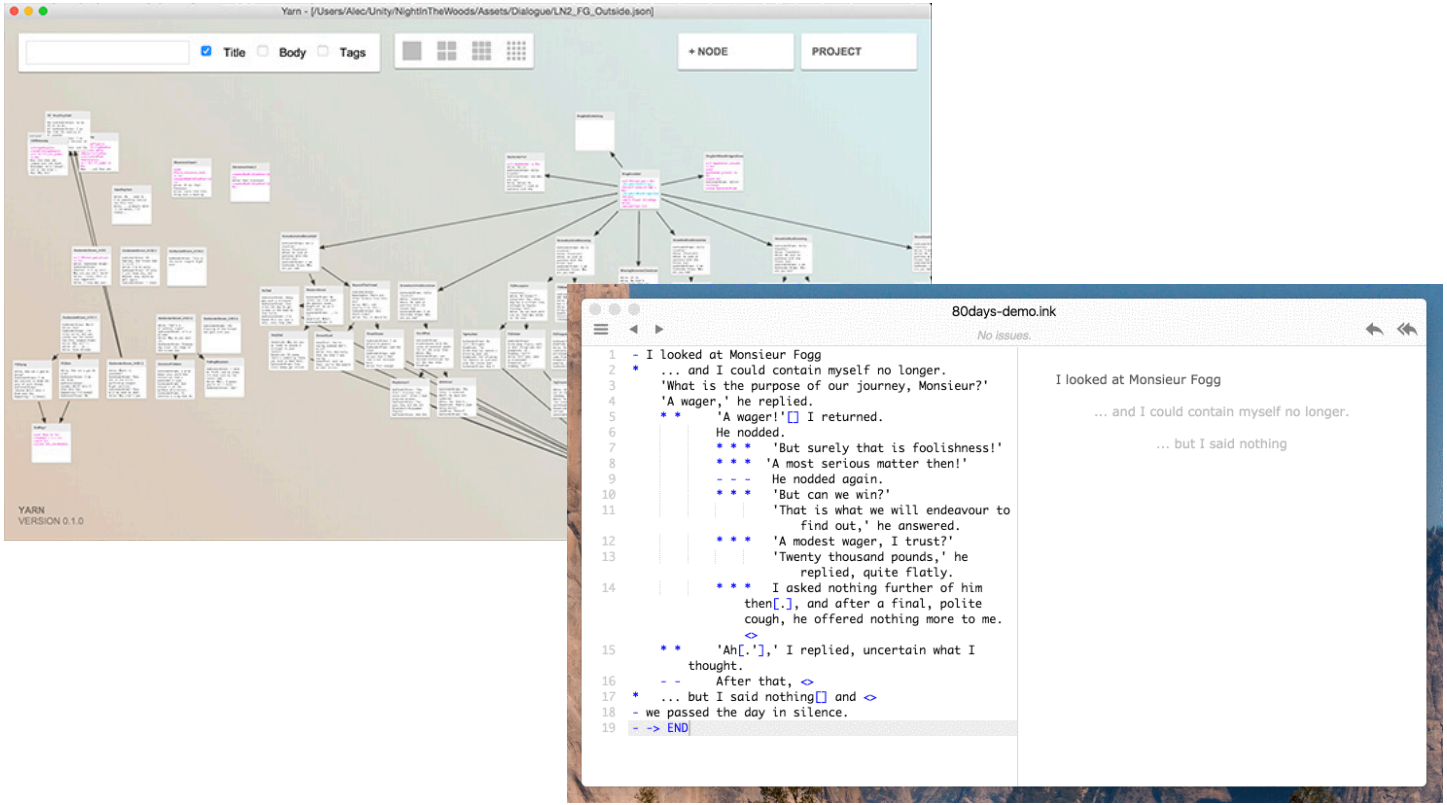
- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Dialog

- Traditional dialog trees
  - Multiple choice
  - Static interactions
- “Broad” dialog trees
  - Symbolization
  - Response generation

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Traditional Dialog Trees



- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

“Broad” Dialog Trees

- Open-ended dialog, with infinite(ish) branches from every node
- All dialog options are available at any time, and response is controlled by context

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Symbolization

Example 1

#greeting

Hello, my name is Dr. Tiller, I'm your Pharmacist today.

@speaker_id:Name @speaker_id:Job

@subject:Patient @datetime:Value

Example 2

#greeting

Yo, I'm Dr. Tiller, I guess I'm a Pharmacist or whatever.

@speaker_id:Name @speaker_id:Job

@subject:Unknown

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Machine Learning

- Few examples => Large coverage

Good afternoon, sir.
Good morning, ma'am.
Hello.
hey
I am a student pharmacist here at the clinic.
I'm a student pharmacist working on your medical team.
My name is Dr. Hubal.
Welcome, Simone.

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Intents

- Similar to verbs, but more abstract
- Can be stacked and scored

```
{
  "intent": {
    "intentName": "greeting",
    "probability": 0.95
  },
  "intent": {
    "intentName": "introduction",
    "probability": 0.90
  },
  "slots": [
    {
      "value": "Dr. Tiller",
      "entity": "speaker_id",
      "slotName": "Name"
    },
    {
      "value": "Pharmacist",
      "entity": "speaker_id",
      "slotName": "Job"
    },
    {
      "value": "Simone",
      "entity": "subject",
      "slotName": "Patient"
    },
    {
      "value": {
        "kind": "InstantTime",
        "value": "2018-02-08 20:00:00 +00:00"
      },
      "entity": "datetime",
      "slotName": "Value"
    }
  ]
}
```


- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

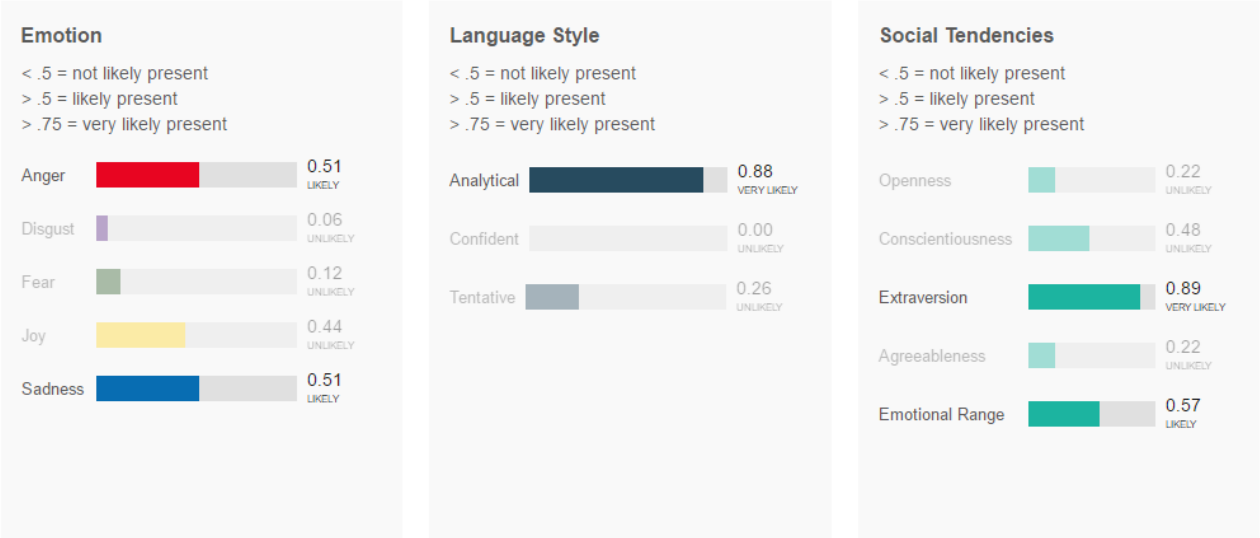
Entities/Slots

- Parsed from the text using Regex
- Usually needs some sort of rule system for extraction

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Sentiment Analysis

- Attributes of overall speech can be extracted



- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Response Generation

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

#inquire_purpose

What brings you in today?

@subject:Patient

@datetime:Today

On #inquire_purpose and @subject:Patient:

If current or previous context contains:

@subject:Patient @speaker_id:Job → I'm here for a checkup.

@subject:Unknown @speaker_id:Job → I'm uncomfortable telling you that...

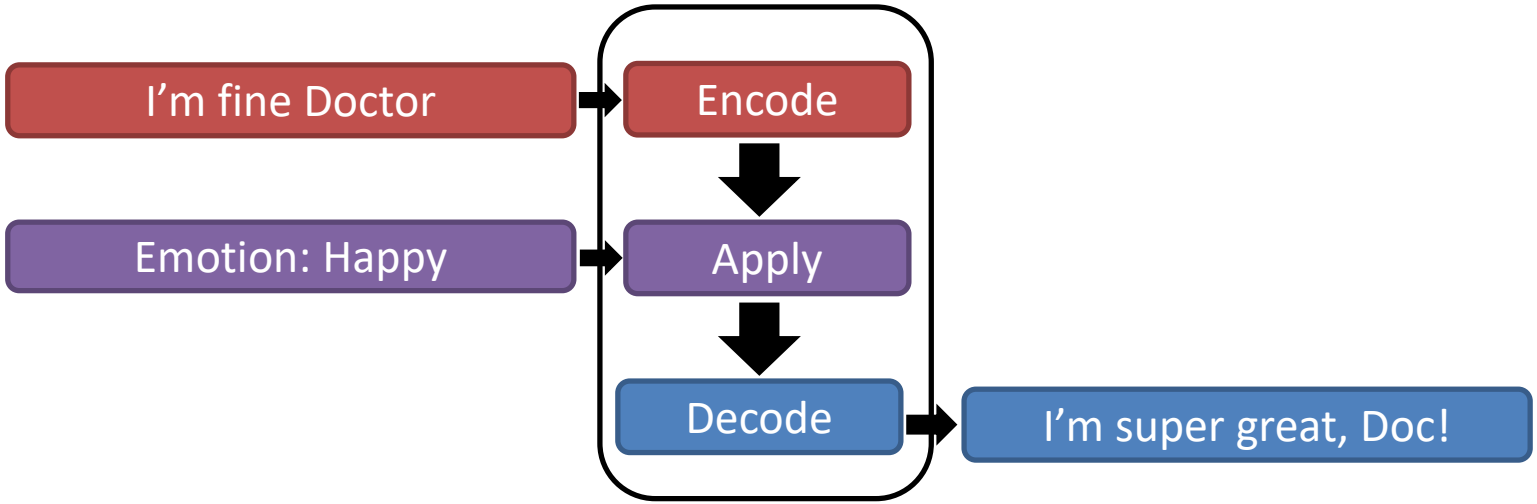
If current or previous context does not contain:

@subject → Are you my Doctor?

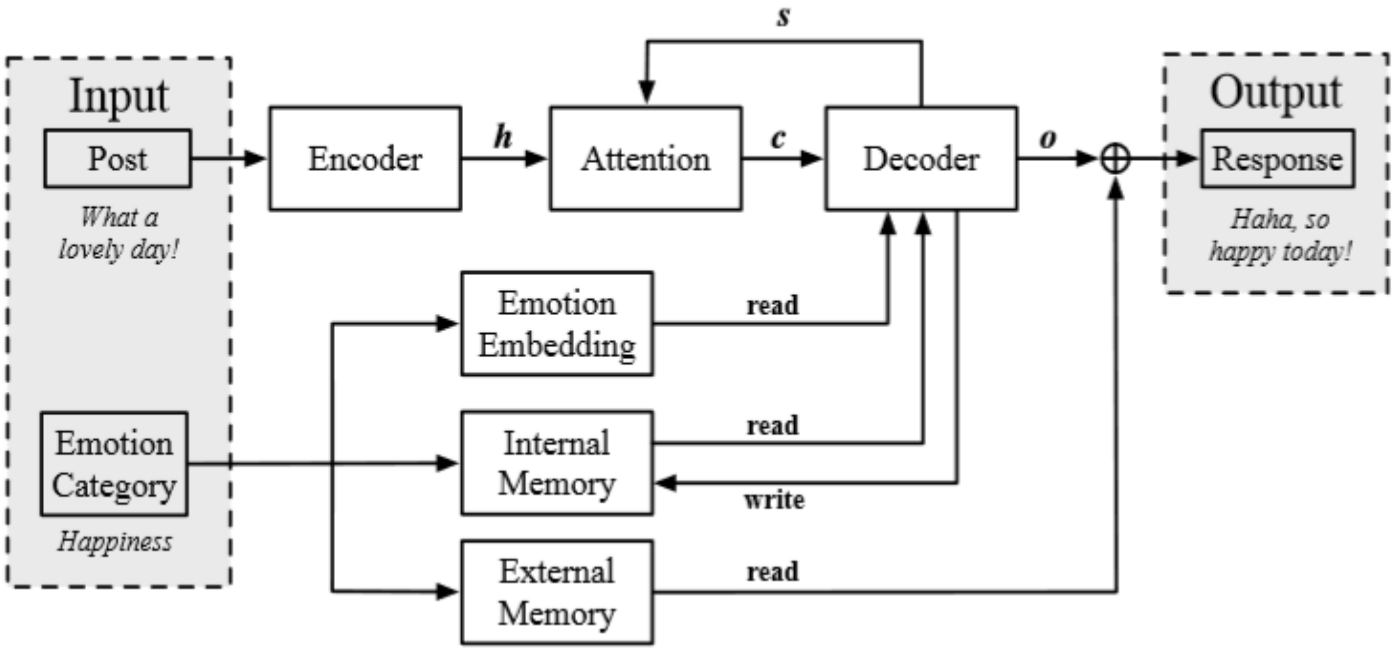
@speaker_id → I'm sorry, who are you?

Response Altering

- Sentiment analysis in reverse



Response Altering (rNN)



- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Procedural Lip Sync

- Phonemes (Audio)
  - Voice cloning
  - Voice recognition
- Visemes
  - Bone transforms/blendshapes
  - Phoneme => Viseme (Naive)
  - Half keys
  - Phoneme => Viseme (Dynamic)

Phonemes

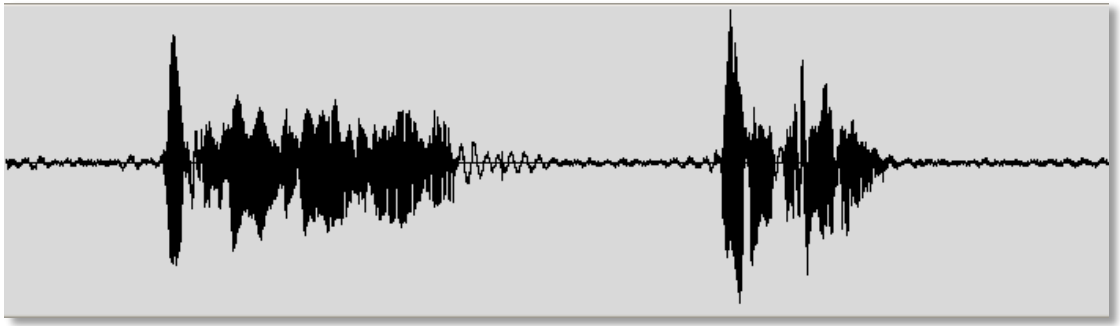
- Distinct auditory sounds in speech

Timit Phoneset	Examples
Pau	-
ay, ah	bite, but
ey, eh, ae	bait, bet, bat
Er	bird
ix, iy, ih, ax, axr,y	debit, beet, bit, about, butter, yacht
uw, uh, w	boot, book, way
ao, aa, oy, ow	bought, bott, boy, boat
Aw	bout
g, hh, k, ng	gay, hay, key, sing
R	ray
l, d, n, en, el, t	lay, day, noon, button, bottle, tea
s, z	sea, zone
ch, sh, jh, zh	choke, she, joke, azure
th, dh	thin, then
f, v	fin, van
m, em, b, p	mom, bottom, bee, pea

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Voice Cloning/Voice Recognition

- While generating audio via voice cloning, you get the phonemes for free
- For voice recognition, you use machine learning to grab the phonemes from audio



- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Static Visemes

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Static Visemes

```
[System.Serializable]
public class
TransformAnimationCurve {
      private AnimationCurve _posX;
      private AnimationCurve _posY;
      private AnimationCurve _posZ;
      private AnimationCurve _rotX;
      private AnimationCurve _rotY;
      private AnimationCurve _rotZ;
      private AnimationCurve _rotW;
//… Scale, etc.
}
```

```
public int AddKey (float time, Vector3
position, Quaternion rotation, Vector3 scale,
float inTangent, float outTangent) {
      int index = _posX.AddKey(new
      Keyframe(time, position.x, inTangent,
      outTangent));
      //… y, z, etc.

      Quaternion fixedRotation =
      Quaternion.Euler(
            CentreAngles(rotation.eulerAngles
      ));

      _rotX.AddKey(new Keyframe(time,
      fixedRotation.x, inTangent,
      outTangent));
      //… y, z, w, etc.

      return index;
}
```

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Static Visemes

```
public Vector3 EvaluateScale (float time)
{
	float x = _scaleX.Evaluate(time);
	float y = _scaleY.Evaluate(time);
	float z = _scaleZ.Evaluate(time);

	return new Vector3(x, y, z);
}

public Vector3 EvaluatePosition (float
time) {
	float x = _posX.Evaluate(time);
	float y = _posY.Evaluate(time);
	float z = _posZ.Evaluate(time);

	return new Vector3(x, y, z);
}
```

```
public Quaternion EvaluateRotation
(float time) {
	float x = _rotX.Evaluate(time);
	float y = _rotY.Evaluate(time);
	float z = _rotZ.Evaluate(time);
	float w = _rotW.Evaluate(time);

	return new Quaternion(x, y, z, w);
}
```

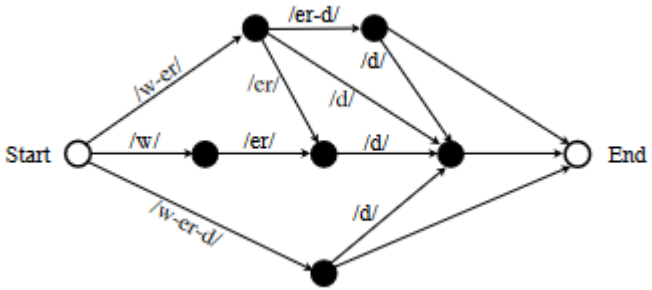
- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

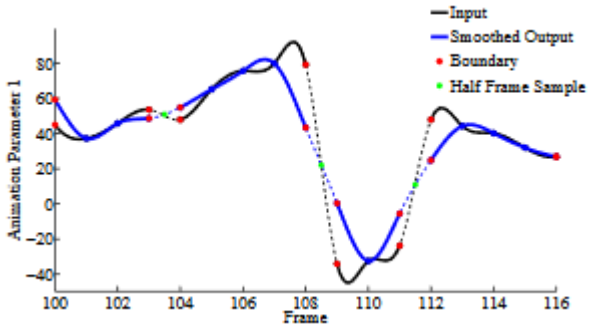
Static Visemes (Demo)

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Dynamic Visemes

- Dynamic visemes are generated physical representations of phoneme clusters





- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Dynamic Visemes

```
public void SmoothValues(float tolerance)
{
      for (int i = 1; i < keys.Length - 2; i++)
      {
          if (_posX[i + 1].time - _posX[i].time < tolerance)
          {
                Vector4 position1 = new Vector4(_posX[i].value, _posY[i].value,
                _posZ[i].value, _posX[i].time);
                Vector4 position2 = new Vector4(_posX[i + 1].value, _posY[i +
                1].value, _posZ[i + 1].value, _posX[i+1].time);

                Vector4 newPosition = Vector4.Lerp(position1, position2, 0.5f);

                _posX.RemoveKey(i);
                _posY.RemoveKey(i);
                _posZ.RemoveKey(i);

                AddKey(newPosition.w, newPosition);
          }
```

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Dynamic Visemes (Demo)

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

CENTER FOR INNOVATION
IN PHARMACY SIMULATION

Hyper Parameters for Tuning

```
public void SmoothValues(float tolerance)

{
      for (int i = 1; i < keys.Length - 2; i++)
      {
            if (_posX[i + 1].time - _posX[i].time < tolerance){

                Vector4 position1 = new Vector4(_posX[i].value, _posY[i].value,
                _posZ[i].value, _posX[i].time);
                Vector4 position2 = new Vector4(_posX[i + 1].value, _posY[i +
                1].value, _posZ[i + 1].value, _posX[i+1].time);

                Vector4 newPosition = Vector4.Lerp(position1, position2, 0.5f);

                _posX.RemoveKey(i);
                _posY.RemoveKey(i);
                _posZ.RemoveKey(i);

                AddKey(newPosition.w, newPosition);
            }
```

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work

Future Work

CENTER FOR INNOVATION
IN PHARMACY SIMULATION

NLU Resources

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work
- Resources

Other Resources

- Presenter
- Project
- Overview
- Dialog
- Lip sync
- Future work
- Resources

CENTER FOR INNOVATION
IN PHARMACY SIMULATION