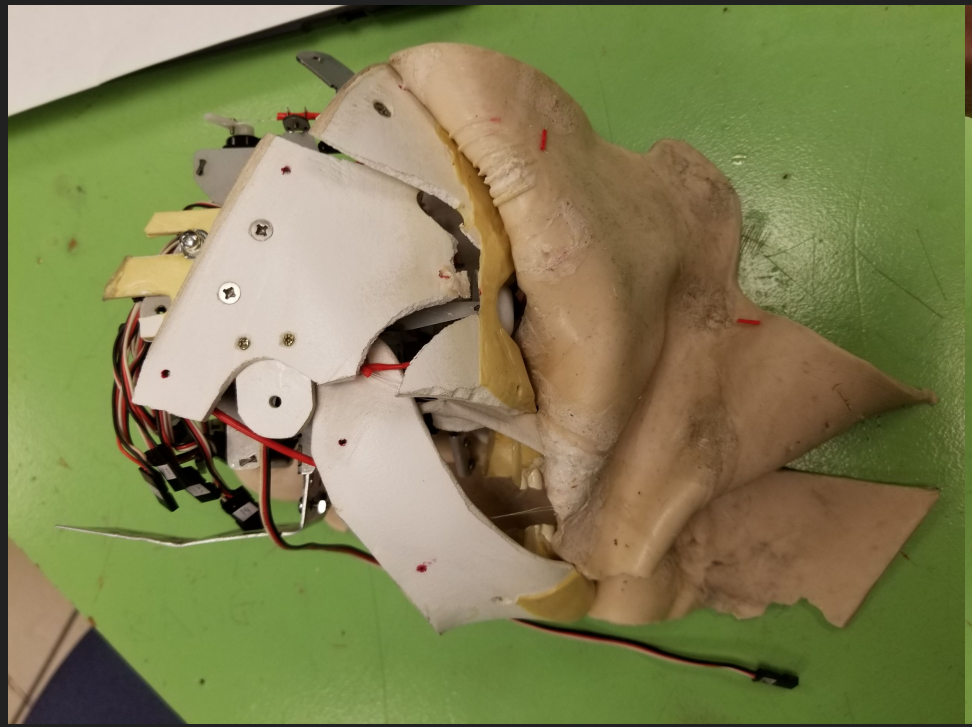
Team 3: Curie Head

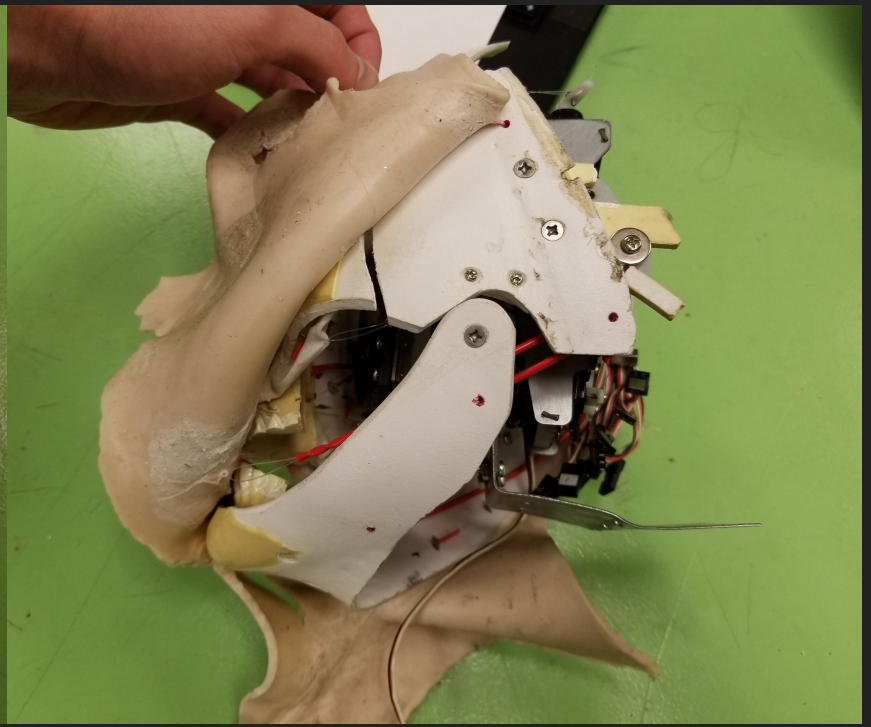
Rob Holt
David Yakovlev

Project Summary

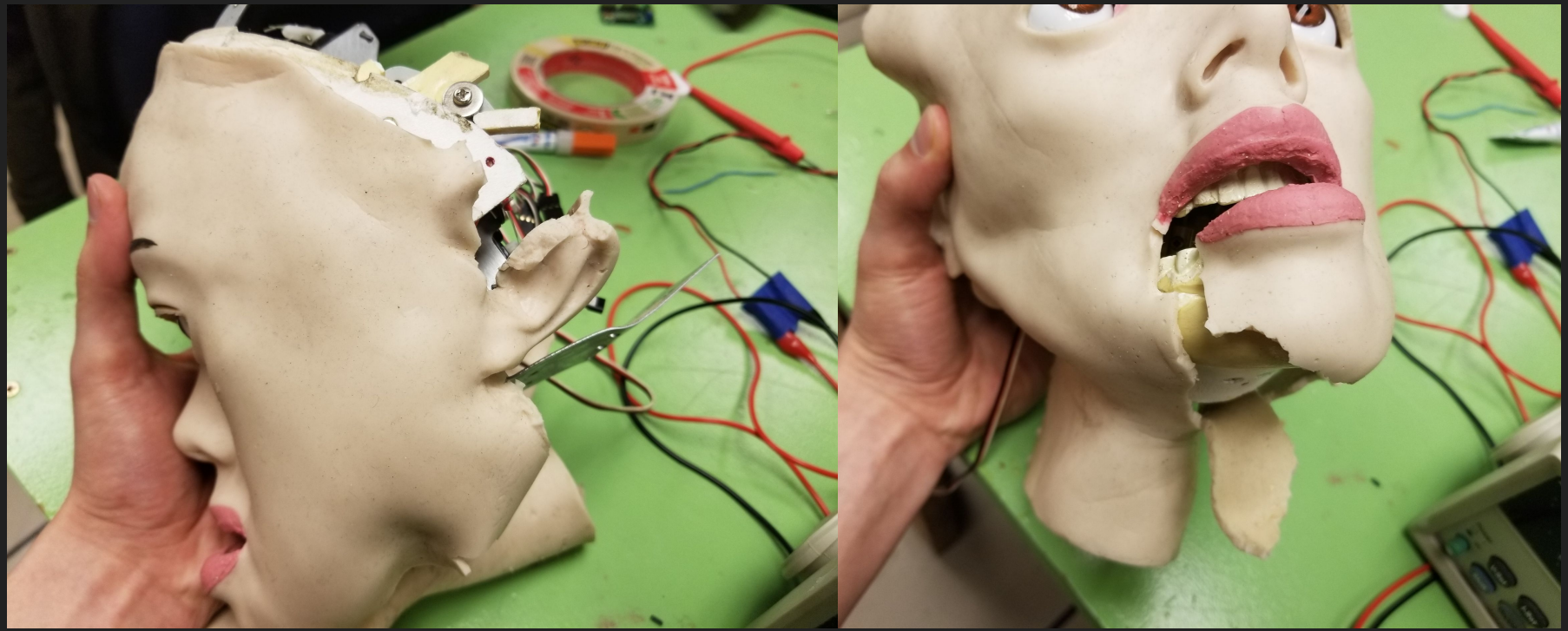
1. Repair Curie Head
    a. Skull broken
    b. Latex ripped
    c. Several other repairs detailed later
2. Verify physical components are in working order and understand computational platform
3. Develop code base to allow easy inclusion of new gestures

Skull Repair





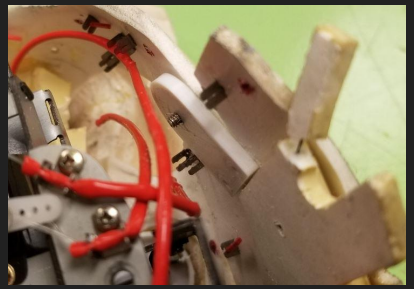
Latex Repair (1 of 2)

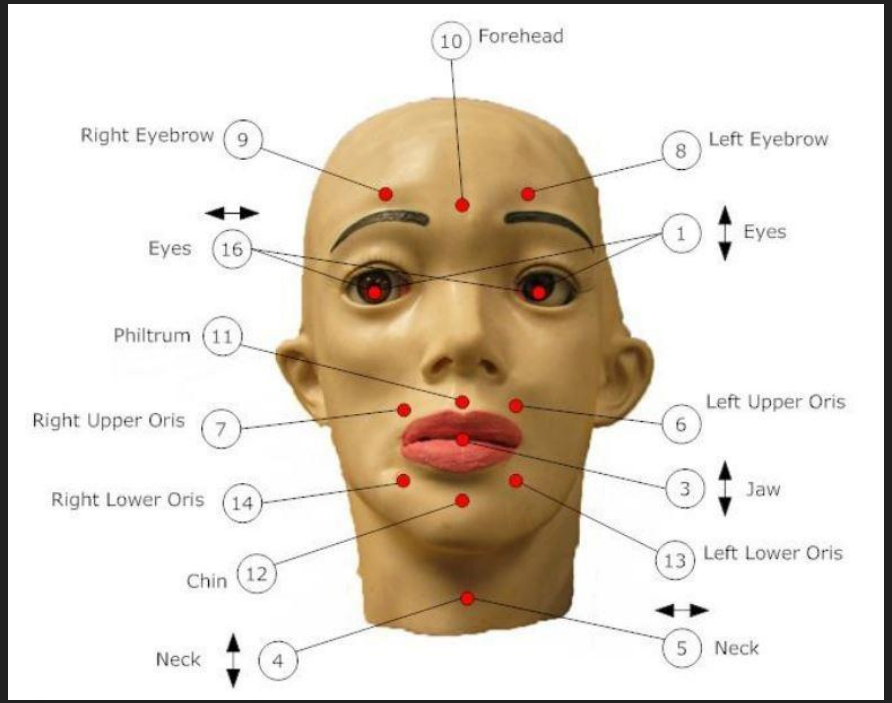


Latex Repair (2 of 2)

Other Repairs

- 2 Servos replaced
- Jaw joint linked to servo
- Jaw to Skull attachment
- Tensioning servo strings to latex
- Re-attach fabric to skin for full oris control
- Acquired new servo controller board





Expression and Gesture Framework

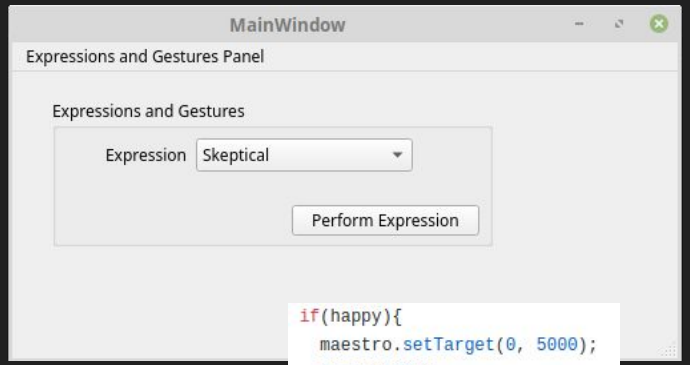
- Desired easy framework for addition of new expressions and gestures
- Needed to reduce complexity from past projects which included Powerpoint presentation, Python, Arduino, obscure protocols.
- **Desired semantic and logical abstractions for servo control and positions**

```yaml
eye_l_r:
  channel: 0
  default_position: center
  positions: {center: 5604, left: 8000, right: 1984}
eye_u_d:
  channel: 1
  default_position: center
  positions: {center: 5792, down: 8000, up: 3584}
jaw_u_d:
  channel: 2
  default_position: closed
  positions: {closed: 7056, halfway: 6360, open: 5212}
eyebrow_l:
  channel: 3
  default_position: resting
  positions: {resting: 5832, up: 8000}
```

New semantic values can be added simply via YAML to an existing text file.

Expression and Gesture Framework (Cont.)

- GUI largely unnecessary but allows for simple testing of created expressions.
- GUI can be extended (if there is interest) to allow individual servos to be set to semantic positions.




```
if(happy){
  maestro.setTarget(0, 5000);
  delay(100);
  maestro.setTarget(1, 6000);
  delay(100);
  maestro.setTarget(6, 4000);
  maestro.setTarget(7, 8000);
  maestro.setTarget(11, 4000);
  delay(100);
  maestro.setTarget(6, 8000);
  maestro.setTarget(7, 4000);
  delay(100);
  maestro.setTarget(11, 8000);
  maestro.setTarget(13, 4000);
  delay(100);
}
```

```
expressions:
  skeptical:
    eyebrow_l: up
    eye_l_r: right
  resting:
    upper_lip: resting
  angry:
    middle_forehead: clinched
    upper_lip: clinched
  meh:
    eye_l_r: left
    eye_u_d: down
  surprised:
    eyebrow_l: up
    eyebrow_r: up
    jaw_u_d: open
gestures:
  Something:
  - !!python/tuple
    - meh
    - 0.0
  - !!python/tuple
    - surprised
    - 1.0
  - !!python/tuple
    - meh
    - 2.0
```

Next Steps ...

- Continue with implementing elegant gesture (expressions coordinated in time). Concept in place, implementation next.
- Looking for guidance on requests in initial project definition
  - Gender recognition using webcam + OpenCV per request in project description?
  - Mouth synchronizations with audio using Amazon Polly and speech marks?

Demo