Report 1

DECO3200 - Human Computer Experience Design Studio

Hanley Weng
Course Blog

Table of Contents

- Project: Misanthropic Bots
  - Project Description
  - Project Goals
  - Independent Goals
  - Background Research
    - Communication.
    - Audio.
  - Approach
    - Communication Findings.
    - Audio Findings.
    - Implementation.
  - Conclusion & Future Work
- Project: Lit-Ecology
  - Project Description
  - Project Origins
  - Project Goals
  - Background Research
  - Approach & Findings
  - Future Work & Conclusion
  - References

Project: Misanthropic Bots

Project Description

A disparate network of everyday computer systems (phones, desktops, tablets, a diy arduino robot) enjoy casual interactions with each other such as conversation (via text-to-speech google translate api) and computer gaming. As a viewer approaches, the computers hide their intelligence; interactions stop, and the system glares at the viewer through its many representations of sight: a robot hand; a swivelling mobile screen & laptop; a webcam-as-eyeball, placing the viewer on the screen. The longer a viewer stays, the more agitated the computers become. Once all viewers move a distance away or turn their backs, the interaction between the computers resumes — it is clear that humans are not welcome in this space.

Project Goals

Due to the expansive conceptual domain, this large group could cover, coupled by a lack of understanding in the technical potential of the installation, it was decided that a few short iterative implementation-conceptualisation cycles were required. Following the abundance of proposed concepts, the group was divided into areas of research: Tactility, Audio, Graphics, Human Detection and Robotic Server-Client Communication, under the umbrella concept of misanthropic bots. The hope is that when research under these different categories are brought back together again, and there has been time to conceptualise, a few simple gems will stand out from which a high impact work can be formed.

Independent Goals

Individual notes regarding each specialty, general concept, and sources of inspiration are added to a shared google doc from which all members contribute. For this week, we also independently specialise in different areas under the guideline:

Research Guideline:

- *Conceptual Research*
  - *past projects, academic papers (e.g. cognitive effects of different implementations)*
- *Technical Research*
  - *implementation of past projects, current technologies*
  - *reliability, flexibility of different potential systems*
  - *what are the environmental requirements, if any (e.g. no sunlight preffered if kinect is used)*
  - *logistics (e.g. what different hardware/software options are possible, what parts are needed, what is their torque/volume/range/etc., what will they cost)*
  - *implement / propose-the-creation-of different objects solely with this research (e.g. under audio; a standalone arduino with a speaker and microphone that mimics everything you say, logistic requirements …), i.e. unreliant on the resulting research of other categories.*

I have chosen to look into how to technically, and conceptually handle **communication** between devices (tablets, laptops, phones) and circuits (arduinos), with Mark Mitchell. I also looked into how **audio** could contribute to our project.

Background Research

Communication.

The technical requirements for implementing communication between robots was explored through research into different communication methods and the hardware constraints of different products (xbee, wifly shields, bluetooth, etc.). The communication protocals between our networked nodes have also been concemplated, but are still heavily reliant on the final concept. Beginning implementation of a basic prototype has helped establish some of these protocals.
Linked here is the Communication Research Notes.

Audio.

Conceptual exploration was conducted on bots that conveyed emotion or communication through audio. Sources of inspiration were commonly found in iPhone applications, and in the sound-design of animations. Audio communication could took the form of:

- Direct Speech
- Robotic Mumble mixed with select words
- Robotic Sounds - Beeps, Boops
- Robotic Hums - Melodic Strings

Upon discussing the concept with a few members and non-designers. It was found that though bots communicating in nonlinguistic buzzes and melodies were more conceptual and profound, language could have a much higher impact on audiences. This works well as text-to-speech appears easier to work with than customized sounds.

Text-to-speech implementations were explored on the operating system, and on the web (both in commercial and free products). Methods of determining what a device would say were also explored in the form of existing artificial intelligence chatterbots.
Linked here is the Audio Research Notes.

Approach

The general approach is listed above under Independent Goals, under Research Guideline.

Communication Findings.

We should aim to implement communication as asynchronous OSC commands that change program states, we should not send large packets of information quickly. The program should ideally be run off a webbrowser for access by more devices (laptop, tablet, phone). As we do require ranges beyond 30m, I recommend the XBee 1mW module for our system to allow for wireless communication. Due to their price ~$27au, I recommend clustering circuits around them – a mock diagram is attached here.

Audio Findings.

In exploring text-to-speech capabilities, it was found that OSX could emulate quite a wide range of expressive voices. Google Translate's Text-To-Speech capabilities also allowed the generation of mp3s in different languages, allowing for multilingual communication. Prewritten transcripts (and audio) seem the most effective form of speech for now. Automated chatterbots can not actually hold a topic, hence an interesting conversation, very well. To note, users tended to like the idea of bots speaking over robotic-sound-design, they also found the idea of influencing a robot's speech interesting (e.g. through their actions or speech).

Implementation.

A basic implementation simulating two laptops talking to one another has been implemented. This is built on Client-Server OSC code. Each client has a local copy of the same transcript (like the subtitles of a film) which they perform (like actors).

The ability to print text was also implemented. It remains to be seen if disparate mediums of simulated communication (printed text and speech) can convey conversation.

Conclusion & Future Work

As our concept solidifies through the joining of our resulting areas of research, more focused implementations can be executed, products bought, and user reactions tested.

References

- XBee & XBee-PRO OEM RF Module Antenna Considerations
- Two ‘Hot Issues’ in Cooperative Robots: Network Robot Systems and Formal Models and Methods of Cooperation” , HL Akin, 2008
- Coordination of Robotic Systems, LectureSlides, Francesco Bullo, NICTA, 2007
- audience-openframeworks
- Text-To-Speech in Processing
- Speech JS
- ESpeak - implementation of Speech JS
- ReadSpeaker
- Chrome TTS Engine
- shakespearebot
- alicebot
- Interview with alicebot creator dr richard wallace expounds
- Jeeney - Chat with AI
- eliza_script
- how-to-create-your-own-online-chat-bot
- 7 Groundrules For An Intelligent Conversation, Jim Berman, 2008

Project: Lit-Ecology

Project Description

> *An ecology of flocking fireflies, represented in an eternally reflected room of hanging vines of leds, move and flutter around their environment, jumping from light to light. As humans enter and traverse their domain, they interact with them through responsive movement and sound.*

Project Origins

This (currently unrproposed) project arose recently from a desire to create and exhibit something aesthetically pleasing. An installation that acknowledges human presence and promotes coexistence across entities. It is formed as a response against more humanly disparate projects. This was derived from initial technical graphical research from the Misanthropic Bots project.

Project Goals

1. Investigate methods of implementing an LED Matrix.
2. Refine the concept, with software that allows control of said matrix, and refine the physical structures hosting this concept.
3. Add human detection to the system.

I am currently in the process of task 1. My current aims are to create an explorable LED Matrix before addressing what this can be conceptually refined into. Holding off on refining the concept faciliates the subconscious processing of varying outcomes, an idea in the current understanding of design thinking and the origins of creativity.

> *“The design thinking process allows us to follow our intuition, valuing the sensibilities and insights that are buried in our subconscious - like the ice below the water. “*
> *– Bill Moggridge.*

Background Research

In undertaking this task, research was done into the primary problem of controlling multiple leds with a single arduino. LED cubes were a great source of documentation that helped guide what hardware and software implementation were required to solve this problem. The components that similiar projects were created from generated a fairly concise list of parts that warranted further research.

Approach & Findings

In exploring for similar projects, LED cubes came up as the most frequent, similar, and well documented examples. These projects utilised multiple techniques to maintain control (or the illusion of control) over a vast amount of leds from a few pins. Most notable was their specific wiring, whereby all cathodes in one dimensional plane were connected, and all anodes were connected on another. Of practical note, a blown led on such a led cube, will not affect other leds. Many used forms of **multiplexing**, and at times, **bitshifting**. '**Persistence of vision'** was also used, which allowed asynchronous control of individual layers of leds, as opposed to an entire cube.

Shift Registers, and other Integrated Circuits (ICs) were used quite a bit as well, including LED Driver chips, which are often simply large shift registers. For example, 5 STP16CP LED sink drivers were used in Nick Schulze's 8^3 led cube. The TLC5940 is another alternative LED driver which has the same concept of a shift register, only requiring 3 digital arduino pins to control a series of LEDs.

Alternatives to individually wiring up and controlling multiple leds were also explored in existing products. The most useful one being the DMXController and Individually Addressable LED strips. However, the spacing of the leds, and the current need to better understand these constructs before adding on additional hardware, prompts the use of individual leds for the time-being. LED Dot Matrixes were another interesting product that I believed could be a quick, reverse-engineer job, however that is not the case as they contain ICs.

Working with shift registers is definitely a good starting point. Tutorials can be found in the Sparkfun Inventer's Kit Manual and the Earthshine Design Arduino Starters Kit Manual.

Future Work & Conclusion

In conclusion, multiplexing, shift registers, various ICs, and persistence of vision are all techniques for creating a collection of individually accessible leds without requiring a pin per led.

In future work, I intend to get myself acquainted with shift registers and similar ICs. Following this, finalise and purchase the relevant products (currently I am considering an Arduino Mega for its 54 pins, a bunch of leds, and a few ICs).

I intend to use some form of 3d modelling, processing, to visualise the ideal setup of the scene, dictating the number of leds, their spacing, and what components will be required in the installation.

As the technical implementation improves throughout the progress of this project, conceptual issues will be addressed. For example, ping pong balls or foil can be used to optimize the brightness of leds.

References

- Earthshine Design Arduino Starters Kit Manual
- How Not To Engineer - LED Cube
- TLC5940
- DIY LED Cube 4x4x4
- LED Cube 4x4x4
- Alibaba - addressable led strips
- Instructables - ED Dot Matrix Display
- LED Dot Matrix Display Box (8X8M) - Assembly