Course: CS 61: Systems Programming and Machine Organization

Course Level: Upper-level undergraduate

Course Description: "CS 61 is a first course in computer systems programming, meaning the creation of high-performance programs that use computer hardware effectively. Although many programs today are written in high-level programming languages—and many programs simply glue together existing components—the best programmers are craftspeople who understand their tools. For software builders, this requires a working knowledge of computer internal organization. It means understanding how machines interpret instructions, how compilers turn programming languages into instructions, and how operating systems combine programs and libraries to create running code. And it requires understanding the factors that affect code performance.
CS 61 introduces you the tools you need to build robust, efficient software and the mental tools you need to understand software systems written by others. We hope you'll discover that systems software development is fun and worth the effort. We intend the course to be broadly accessible, though it will be easier for those who have some experience with systems programming in C++ or other C-like languages. (Course description)"

Module Topic: ASCII, Unicode and the Ethics of Natural Language Representation

Module Author: Cat Wade [later contributions by Marion Boulicault]

Semesters Taught: Fall 2018, Fall 2019

Tags:

systems [CS]
ASCII [CS]
Unicode [CS]
natural language encoding [CS]
harm [phil]
representational harm [phil]
allocative harm [phil]
stereotypes [phil]

Module Overview:

In this module, we consider the ethics of natural language representation in modern software systems. Software systems play a central role in how we communicate with one another, and the computer scientists who design these systems are sometimes faced with difficult choices about what representational resources they should make available to their users. To what extent should social media platforms support the vast array of languages used throughout the world? To what extent should the developers of smartphone operating systems provide their users with emoji reflecting the diverse identities and communicative needs of members of minority groups?

This module is co-taught by the professor for the course and the Embedded EthiCS Fellow. After an introduction to the ethical issues considered in the module from the TA, the professor gives a brief presentation on the technical and historical dimensions of the module's core case study: the shift from ASCII to Unicode, and the associated choices developers made about which languages to support. The Fellow then leads a discussion of how these choices members of different linguistic communities, and why those effects matter morally. Finally, students consider various strategies that the developers of Unicode could adopt to better address the needs of minority communities.

Connection to Course Material:

This module occurs during the course's first unit on data representation and storage. The professor's presentation during the module expands on the technical material already covered in this unit, with a focus on how it applies to the module's central case study: the shift from ASCII to Unicode. This sets the TA up to lead a discussion of how the technical issues discussed by the professor interact with broader social and ethical concerns.

Marginal Note: We have found that it is important to build modules around real-world case studies that both connect to technical material discussed in the course and raise ethical issues that students can readily appreciate. The shift from ASCII to Unicode has both features. Further, Unicode is the standard for encoding emojis, which provide a particularly intuitive and relatable way to illustrate the module's core philosophical concepts (see the sample class activity below).

Module Goals:

- Familiarize students with the technical aspects of ASCII and Unicode, and with the social and technical considerations that drove the shift from ASCII to Unicode.
- Introduce students to two philosophical concepts that are useful for evaluating formal systems for representing natural language: allocative harm and representational harm.
- Give students practice applying these concepts to evaluate choices made by software developers about what representational resources to provide to their users.
- Give students practice identifying and evaluating possible strategies for alleviating representational harms in the design of formal systems for representing natural languages.

Key Philosophical Questions:

1. How should software developers decide what representational resources to make available to their users for use in communication?
2. In what ways can the choices developers make about what representational resources to make available negatively affect the members of different communities, including minority communities?
3. What is the difference between 'representational' and 'allocative' harm?
4. What are stereotypes, and in what ways can relying on stereotypes harm others?
5. Were the choices made by the developers of ASCII and Unicode the right choices, given the constrains they were operating under, or were there other choices that would have been better from a moral perspective?

Key Philosophical Concepts:

- Harm and intent
- Representational harms and allocative harms

- Stereotypes

Assigned Readings:

Note: Due to the fact that this was an intensive first year course, the module had no assigned readings.

Class Agenda:

1. An introduction to the ethics of character encoding: should emoji be more inclusive?
2. Representational harm vs. allocative harm.
3. Active learning exercise: how could developers make the current set of emoji more inclusive?
4. Technical material – ASCII, Unicode, UTF-8 (presented by CS professor)
5. Representational and allocative harms in the development of ASCII and Unicode.
6. Remaining ethical issues with Unicode, and how best to address them.

Sample Class Activity:

After being introduced to the concept of representational harm, students are presented with a slide containing the current set of 'yellow' emoji representing families of different kinds. In small groups, students discuss what kinds of families are left out from the current set and whether those omissions constitute representational harms. The Embedded EthiCS Fellow then asks the students to split into small groups again. Half the groups are asked to formulate an argument that the current set does represent a representational harm (e.g. those groups who are already marginalized are usually the ones not represented, furthering their marginalization). The other half is asked to formulate an argument that set does not constitute a representational harm (e.g. it is pragmatically impossible to represent every different in an emoji set). Groups are then called upon alternately to generate a debate-like discussion.

Module Assignment:

Note: Due to the fact that this was an intensive first year course, the module had no assignment.

Lessons Learned

Student response to this module has been positive. Students are highly engaged and seem to grasp the key concepts. Class discussion is energetic and fruitful.

- This module is distinctive in the way it integrates philosophical, historical and technical content. The class is structured so that the philosophical content (e.g. the introduction and use of the concepts of representational and allocativeharm) taught by the Embedded EthiCS Fellow fits in directly with the historical and technical content taught by the professor. This demonstrates to the students that the philosophical, historical and technical dimensions of natural language processing are strongly intertwined and must be understood in tandem
- Co-teaching the class also demonstrates to the students that the professor is invested in the ethical dimensions of the course. Given that the professor is the main authority figure and

that she already has a relationship with the students, this demonstration of investment gives extra weight and authority to the ethical material.

- This module uses the example of emojis to introduce the students to the philosophical concepts. Starting with a familiar and simple example before getting into complex material makes the module more accessible and engages students right from the start (even those who might be struggling with the technical aspects).