The Current State Of Technology for the Blind and Visually Impaired

Jeff Wilson, Andreas Stefik
Southern Illinois University Edwardsville

SIGCSE 2012

Accessibility technology is highly fractured and inconsistent

- Most major operating systems now have some kind of disabilities support built in (with varying success)

- Unfortunately, these technologies are woefully outdated, having changed little since the days of console applications

- Accessibility technologies are, almost always, incompatible with similar technologies on different platforms
  - Windows – IAccessible, COM (JAWS, Windows EYES, NVDA)
  - Mac – NSAccessible (Voice Over)
  - Linux – Varies
  - Mobile Phones – Varies
  - Java – Java Accessibility (Major technical problems)

Tools like modern screen readers work well … sometimes

- Screen readers are remarkably effective for the following kinds of applications (with notable exceptions)
  - Web pages
  - Forms
  - Console apps/Trivial apps
- Screen readers either do not work at all, or are woefully ill equipped, to handle applications like the following:
  - Video games
  - Highly visual environments (e.g., Novice Programming Environments)
  - Cross platform applications (e.g., even Microsoft PowerPoint is not accessible on Mac – it makes no sound as you type)

Blind individuals have a vested interest in solving technological problems

- Many of the problems that currently exist can potentially be solved by training blind and visually impaired individuals in technology creation

- Training blind and visually impaired individuals may provide a lucrative employment path, which is important because
  - Approximately 8% of families have a member with a sensory disability (e.g., blind, deaf)
  - Families with a member that has a sensory disability:
    - Make less median income (54, 515 vs. 38,755)
    - Are more likely to live in poverty (11% vs. 9.2%)
    - Are more likely to be either unemployed or not in the labor force (14.9% vs. 41.3%) (2000 Census data)
  - 56% of working adults with vision loss are out of the workforce (September 2010 Bureau of Labor Statistics (BLS) data).

We are training students with visual impairments

- As part of the NSF Broadening Participation in Computing program, we have created custom technologies and social programs for working with the blind and visually impaired population

- Multiple partners, including schools for the blind and visually impaired in five states. Each partner teaches the equivalent of CS 1 or 2 to students. These partners are in:
  - Washington
  - Texas
  - Tennessee
  - Massachusetts
  - Indiana

Custom technologies assist in training these students

- **Phonemic** – A cross-platform, unified, architecture for outputting text-to-speech
- **Sodbeans** – A cross-platform integrated development environment with features designed for the blind (e.g., talking debuggers, magnifiers, auditory code completion, auditory annotations)
- **Quorum** – A general purpose programming language. Studies have shown this language is easier to understand for novices in general compared to C-style syntax. It is also easier to “say” through a screen reader.

Even highly visual applications can be made accessible

- Using phonemic and other tools enables us to cooperate with traditional screen readers much more easily. This has allowed our lab to build:
  - An accessible, cross-platform integrated development environment
  - A fully accessible 3D computer game in the Unity 3D game engine
  - A first person shooter game for the blind and visually impaired
  - A number of smaller games (e.g., maze playing games)
  - Soon: Botball robotics platform.

The CS community needs a stronger focus on *technology* for the blind and visually impaired

- Notice that most major endeavors for novice computer scientists are not accessible:
  - M.I.T.’s Scratch is almost completely non-accessible (using Mac Voice Over or PC NVDA)
  - C.M.U’s Alice is “mostly” non-accessible (using Mac Voice Over or PC NVDA)
  - Other platforms, screen readers, and novice tools may vary
- Even general purpose programming languages have challenges for the blind community
  - How do Python white space rules affect the blind?
  - Typical languages use esoteric syntax.
    - Note that the common for(int i = 0; i < 10; i++) {} translates into
    - “for left paren int I equals 0 semicolon I less than ten semicolon I plus plus right paren left brace right brace,” a phrase which is aurally tedious for a screen reader user.

Improving technology for the blind is highly tractable

- Even full-animated 3D games can be made accessible
- Novice programming environments *could* consider the blind and visually impaired community with software changes
- Researchers could develop a new generation of technologies for the blind and visually impaired, fixing some of the core problems:
  - Significant cross-platform inconsistencies
  - Screen reader APIs are highly inflexible and screen reader specific (e.g., JAWS scripts, NVDA scripts, Windows EYES scripts, Voice over – Not extensible)