Alternative Techniques for Blind Low Vision Students Enrolled in Coding Courses

Robert Jaquiss,[1]

[1]Independence Science

Abstract

BLV (blind low vision) students face difficulties when taking coding classes. Individual Development Environments (IDE) can be inaccessible which make it difficult if not impossible for BLV students to complete coursework. Alternative techniques are described which will allow a BLV student to successfully complete coursework. The author has found the Command Line Interface (CLI) is often a recommended alternative to a Windows based IDE interface. Many blind computer programmers use the Windows and Linux command interfaces which are described as part of this discussion.

Editor's note: The author is a long-time computer programmer who is blind. He has been actively involved in the access technology industry along with the National Federation of the Blind for a number of decades. The author shares his extensive knowledge of the challenges that have presented themselves to persons who are blind or vision impaired who wish to pursue career paths in computer science. This manuscript is based on a presentation that he delivered at the 2021 ISLAND conference. A page providing more resources is at: https://independencescience.com/computer-science-resources/.

Brief Historical Background

The term coding was formerly known as computer programming. Historically, IBM created and established a hierarchy of job positions of which the coder was the lowest. The coder performed the task of coding, writing the actual code that was punched into cards to create a program. The coder's task was not a very creative task compared to that of the program analyst, senior programmer or programmer who designed the software.

BLV people have been involved in the computer industry since the 1960s. The ACM (Association of Computer Machinery) formed SIC-CAPH which was later renamed SIGCAPH Special Interest Group on Computers and the Physically Handicapped. It should be noted that SIGCAPH comprised persons with other disabilities than blindness.

BLV persons who read braille were provided braille output by means of modifying line printers such as the IBM 1403 and daisy wheel printers. The line printers were modified by placing a piece of stretched elastic between the paper and print hammers. Daisy wheel printers such as the Diablo 630 or 1630 could be modified by stretching a piece of laboratory tubing over the platen. In either case braille was produced by using the period to create braille dots on the back side of printer paper. The braille was legible, but not very long lasting. In the late 1960s, MIT developed the MIT BrailleEmboss.

According to an article; Duxbury Systems – Early History of Braille Translators and Embossers [1]; at a conference in 1969, four braille embossers were shown. The MIT BrailleEmboss was the only full-page embosser. The other three produced by Carbonneau, Morrison and Knox produced braille on paper tape.

In the 1970s, Triformation Systems (later known as Enabling Technologies) created the BD-3. LED-1 and LED-120 braille embossers. The BD-3 embossed braille on the same type of paper tape used in machines such as the ASR33 teletype. The LED-1 and LED-120 were able to emboss braille on continuous form braille paper. The latter two machines contained keyboards and were used in the same way teletypes were used for interacting with computers. By late 1976, four full-page embossers had been developed: the MIT BrailleEmboss, the Triformation LED-1 and LED-120, and the French Sagem Embosser.

The late Dr. Tim (T. V.) Cranmer developed the Cranmer Modified Perkins Brailler. This device was a substantial modification of a Perkins Brailler and could produce braille on a few pages at a time. Thiel later produced the Thiel Beta X3 embosser which could also be used like a teletype. Telesensory Systems Inc. also known as TSI produced the Optacon with an accessory which could be used to read a CRT display. It was possible to read a CRT screen with an Optacon, but this was very tiring.

In 1975, Handitech in Germany produced the first single line braille display. Other companies such as Tieman, and Papenmeier also produced braille displays. Most braille displays used piezoelectric crystals to raise the pins on the braille display. This technology has been the predominate technology until the present day.

Duxbury Systems has created a site: Documents on Automated Braille Production https://www.duxburysystems.com/braille_tech.asp. This site contains a wealth of information on the history of braille translators and automated production.

Contemporaneously with the development of braille displays and embossers was the develop-

ment of speech synthesizer hardware. In the past, the usual form of speech synthesizer hardware was a card which was inserted into a PC. The decTalk and some other systems used an RS232C connection to connect a PC with an external box. The latter configuration proved to not be as responsive as did internal cards. Both IBM type PCs and Apple iie™ systems could be made accessible. MSDOS and Apple systems could be configured to work with teletype type machines.

Until the advent of the Apple Macintosh™ and Microsoft Windows ™ GUI operating systems, BLV persons could easily work with IBM PCs, Apple iie™ and large mainframe computers. The character based CLI interface works well with braille displays. The BLV user can easily read the input that is typed and the output that is displayed.

In 1984, Apple introduced the Apple Macintosh. The Apple Macintosh used a GUI interface and was inaccessible by the BLV community. In 1990, Microsoft introduced Windows 3.0 which was also inaccessible (See Christoph Dernbach The history of the Apple Macintosh).[2]

In the early 90s, a software package from Slimware Bridge made it marginally possible to use Windows 3. In 1995, F.H. Papenmeier introduced a system using the Braillex braille display, hardware interface and software to make Windows 3 more accessible. In 1998 Windows 95 became marginally accessible. Windows 98 became accessible in 2000 and Windows 2000 became accessible in 2002. Windows XP became accessible a few months after its initial release. In the past few years, Microsoft has provided screen access software companies access to beta versions of their products. This has greatly decreased the time between the public release of a Windows system and that system being accessible.

Screen access software developers face the difficult task of knowing what is on the screen and where specific icons and text are located. The access software then has the task of converting a visual presentation into text which can be either spoken, displayed as braille, or displayed and spoken together.

The Challenge for BLV Students

BLV students, like their sighted peers, need STEM education. The ADA and IDEA mandate that disabled students receive accommodations in order to take courses of study. An effort known as The Hour Of Code[3] has been created to expose all K-12 students to computer coding as a way to increase awareness of Computer Science. Significant difficulties are encountered when BLV students wish to take part in coding courses. Sighted students can make use of icon type low code or no code-based systems to create applications. The Hour Of Code site does provide information on how to provide accessible alternative activities for the Hour Of Code. For BLV students, the alternative activities require the use of the Quorum language.[4] See the articles: Astronomy Hour of Code Activity 1 | Quorum Programming Language[5] and Hour of Code: Part 1 | Quorum Programming Language.[6] And the American Printing House for the Blind (APH) has two helpful articles: Road to Code[7] and Coding With Mantis Q40.[8] Professional level systems allow for the creation of complex business applications. The problem with low and no code systems is that they are icon driven. The user uses a mouse or touch screen to drag and drop icons within a WYSIWIG screen. The system then generates the actual application. See the articles: What is Low-Code? A Quick Introduction to Low-Code Development[9], Ten Best Low-Code Development Platforms in 2021[10], Best No-Code Development Platforms Software in 2021[11], and The

Home of No-Code: A curated directory of resources and tools for non-techs.[12]

Make Preparations in Advance

The Quorum language and offerings from APH will certainly provide a BLV student an introduction to coding. BLV students and professionals need to learn other languages. Many mainstream applications such as those from Microsoft are accessible. Problems arise when specialized applications such as IDEs are utilized. BLV students can find themselves faced with trying to learn a new difficult to use application and keep up with schoolwork.

This author suggests the following strategies:

1. Plan ahead, the further the better. As much as possible plan out when each course required for a degree or certificate will be taken.

2. Find out what software will be used for each course.

3. Remember that if a course is to be taken two or three years in the future, the more likely the software used for the course will change.

4. Arrange for textbooks with as much lead time as possible. Coding courses like other STEM courses are difficult to read orally. A braille display and/or embossed braille will make the content very much easier to learn.

5. Be prepared to use human readers to help with coursework. Making good use of time is important. Struggling with an inaccessible system will lead to frustration and possible failure to complete a course.

6. Long before taking a course, try out the software and hardware that is to be used and determine if it is accessible. Talk with the instructor about possible alternatives.

Getting Started with Coding

Learning any language will require the use of reference materials and tutorials. Fortunately reference materials for computer languages, operating systems, networking hardware and Computer Science are available in accessible formats. The two leading sources for accessible materials are BookShare and Learning Ally. Some other sources are APH (American Printing House for the Blind), ATI (Access Technology Institute), NBP (National Braille Press) and Your Tech Vision. (See the Independence Science web site: http://www.independencescience.com page on coding for more information.)

Three IDEs are usable by BLV persons. These are: Visual Studio Code or Vscode[13], Eclipse and Sodbeans. This author has not seen recent references to the Eclipse IDE, Eclipse may have been superseded by Vscode.

Unix® and Linux® systems traditionally provide a Command Line Interface (CLI). This interface is sometimes referred to as the console. CLI interfaces work very well with the technology used by BLV persons. Braille displays can be connected to Unix® and Linux® systems using the Brltty[14] package. In earlier years, a PC running MSDOS with a braille display could be connected using Telnet or PCNFS. Today, the packages Putty and SSH can be used along with a terminal emulator.

The Gnome desktop can be used by BLV persons using the ORCA[15] screen reader. Emacs is made accessible using Emacspeak.[16] (See the Independence Science web site: http://www.independencescience.com page on coding for more information.)

A Common Approach

Since IDEs can be difficult to use, one common approach for BLV persons is to use the CLI. On Windows, start the Cmd system. On Windows 7, Cmd is located under the Accessories menu displaying the name Command Prompt. An alternative is to press the Windows-key+r, then press <enter>. When done correctly, the user will see, "C:>".

The user can then use Notepad or other editor to code a program. Programs can be compiled directly from the command line. Some examples are:

Python demo.py
The preceding command will compile and run the code within the file demo.py.

Cc –c demo.c
The preceding command will compile a program in the C language.

Javac demo.java
Java demo
The preceding two commands will compile a Java program. The second command runs the program presuming it compiled without errors.

When an error occurs either with compilation or when running a program, it can be difficult to read the error messages. One technique to solve this challenge is to use the following type of batch file:

@echo off
rem Run a Python program.
python demo.py> demo_output.txt 2>&1

The above batch file will run the Python compiler using demo.py for the Python code. The demo_output.txt directs the output to be saved in the file demo_output.txt. The 2>&1 at the end of the command line directs any error messages to also be saved in the demo_output.txt file. Notepad can be used to review the output and possible error messages.

The Notepad++ program provides a more advanced editing system. Plugins are available for several languages. Notepad++ can be set up to allow the user to compile and run a program.

The Advantages of Braille

Coding is by nature a precise activity. The proper syntax of a language must be entered correctly. Omitting a comma, semicolon or brace will cause an error. Variables are case sensitive. In languages such as Python, Java and Go, there are conventions for when and where to capitalize the letters used in class, object and method names. C and Java use braces ( ) to delimit blocks of code. Within a pair of braces, the code is expected to be indented from the left margin. Proper indentation is considered so important that code beautifier software is used to read programs and properly format them. In C and Java, indentation will not hinder compilation of a program. Python is the exception. The indentation from the left margin is part of the program syntax. A block of code must be properly indented in order for it to compile correctly.

The standard eight dot cell used in most braille displays allows uppercase letters to be displayed using one cell. This is especially important when dealing with columnar material. This author has for some years used an eighty-cell braille display for coding. The advantage of an eighty-cell display is that normal lines of code don't wrap to the next line. A braille embosser can be a useful tool for producing reference printouts of programs and manuals.

Conclusion

In conclusion, coding can be accessible to BLV persons. A BLV person can by using alternative

techniques successfully take a coding course and pursue a career in computer related technologies. Like other STEM subjects, a knowledge of braille is absolutely essential. For reading manuals, a forty-cell braille display will make it possible to read brailled materials. For coding, an eighty-cell braille display will allow lines of code and other information to be displayed without breaking up lines of text. An eighty-cell braille display allows for displaying in braille the same material as a sighted person sees in print.

For more information, See the Independence Science web site: https://independencescience.com/computer-science-resources/ page on coding for more information.

References

1. Duxbury Systems – Early History of Braille Translators and Embossers
https://www.duxburysystems.com/bthist.asp
Retrieved Sept. 2, 2021.

2. The History of the Apple Macintosh
February 10, 2021 by Christoph Dernbach
https://www.mac-history.net/featured-2/2021-02-10/the-history-of-the-apple-macintosh
Retrieved August 31, 2021.

3. Hour of Code
https://hourofcode.com/us
Reprieved Sept. 4, 2021.

4. The Quorum Programming Language.
https://quorumlanguage.com/
Retrieved Aug. 2, 2021.

5. Astronomy Hour of Code Activity 1 | Quorum Programming Language
https://quorumlanguage.com/hourofcode/astro1.html
Retrieved Sept. 4, 2021.

6. Hour of Code: Part 1 | Quorum Programming Language
https://quorumlanguage.com/hourofcode/part1.html
Retrieved Sept. 4, 2021.

7. Road to Code
https://www.aph.org/aphs-road-to-code/
Retrieved Sept. 3, 2021.

8. Coding With Mantis Q40
https://www.aph.org/coding-with-the-mantis-q40/
Retrieved Sept. 4, 2021.

9. What is Low-Code? A Quick Introduction to Low-Code Development
https://www.mendix.com/low-code-guide/
Retrieved Sept. 4, 2021.

10. 10 Best Low-Code Development Platforms in 2021
https://www.softwaretestinghelp.com/low-code-development-platforms/
Retrieved Sept. 4, 2021.

11. Best No-Code Development Platforms Software in 2021
https://www.g2.com/categories/no-code-development-platforms
Retrieved Sept. 4, 2021.

12. The Home of No-Code: A curated directory of resources and tools for non-techs
https://www.nocode.tech/
Retrieved Sept. 4, 2021.

13. Visual Studio Code or Vscode
https://code.visualstudio.com/
Retrieved Sept. 6, 2021.

14. Brltty
https://brltty.app/
Retrieved Aug. 23, 2021.

15. Orca Screen Reader
https://help.gnome.org/users/orca/stable/
https://en.wikipedia.org/wiki/Orca_%28assistive_technology%29
Retrieved Sept. 6, 2021.

16. Emacspeak Mail List
https://www.emacspeak.org/
Emacs - Wikipedia
https://en.wikipedia.org/wiki/Emacs
Retrieved Sept. 6, 2021

Appendix: Definition of Terms

ACM Association of Computer Machinery

ADA Americans with Disability Act

APH American Printing House for the Blind

BLV Blind or Low Vision braille display A braille display consists of a line of refreshable braille cells. The user can feel what is displayed. Typical sizes of displays contain 12, 20, 32, 40 or 80 braille cells.

CLI Command Line Interface; MSDOS, Linux and UNIX are examples.

CRT Cathode Ray Tube, the same technology which was used in older television sets

Daisy wheel A daisy wheel printer uses a rotating wheel containing a type font. The wheel rotates to the desired position and a solenoid strikes the character causing it to be printed on the page.

Duxbury Duxbury software available from Duxbury Systems is a package used to translate documents into braille. Duxbury can read a variety of document formats.

GUI Graphical User Interface

IDE Individual Development Environment, Eclipse, NetBeans and Sodbeans