Computer Programming has shifted and evolved many times throughout the history of computing. These shifts have been the result of new advancements in hardware or software within the computing realm. The first largest impactful shift was when the keyboard was invented and introduced as a means of input rather than punch cards. Computing then moved on to include mouse input as well as keyboard input. Now computing moves forward and is looking toward vocal input in the near future.

In the early years of computing technology, electronic computers could only be programmed a certain number of ways. They could be programmed by numbers, tape, punch cards, or the manual manipulation of the thermionic valves or vacuum tubes to certain settings. Punch cards, also known as Hollerith cards and IBM cards, are paper cards containing several punched or perforated holes that were punched by hand or machine to represent data (Hope, 2017). The cards then allow for the storage and access of information through entering the card into a computer. Punch cards were known to be used as early as 1725 for controlling textile looms (Hope, 2017). Joseph Marie Jacquard used punch cards to create patterns in woven silk (Hope, 2017). Later cards were used by Semen Korsakov to store and search for information in 1832 (Hope, 2017). In 1890, Herman Hollerith developed the method for machines to record and store information on punch cards for the US Census and later he formed the company known as IBM (Hope, 2017). Punch cards were essential for the storage of information because many early electronic computers did not have much if any storage. Most cards featured 80 columns (the 1890 Census had 80 questions), multiple punches in each column could be used to represent either a letter or number. These were also typically printed at the top of the cards in ink so that the cards could be read by eye. For programming, each card usually contained one statement, so that re-organizing code could often be done through re-ordering the cards (Cardinal, 2011).

Punch cards themselves are not input devices but a punch card reader that sends data to the computer it an input device. Researchers at MIT in the 1950s began experiencing with direct keyboard input to computers. Up to this point computer users fed their programs in computers using the punched cards or paper tape and their readers. Doug Ross from MIT connected an electrically-controlled typewriter to an MIT computer to experiment with keyboard input (Computer Science – Evolution of Computers in Society). Ross’s experiment was successful and confirmed how useful and convenient a keyboard input device could be. Keyboards were a large leap forward from punch cards. They made editing code easier and time effective through the matter of re-keying input, instead of ripping up and retyping entire cards (Cardinal, 2011). The “QWERTY” keyboards layout was adopted form typewriter design (Cardinal, 2011). Keyboards were also limited in comparison with the post-mouse era. The keyboards as a pre-mouse input mechanism affected its development. An example is that it was not until recently that the Scroll Lick key disappeared from PC keyboards, even though the need for a dedicated scroll mode dissipated with the advent of the mouse and scrollbar (Cardinal, 2011). Another example is the System Request key, dating back to the need for mainframe terminals to issue interrupts. Now the key has an additional purpose of printing the screen image although even that function is often rudimentary and typically been superseded by window clipping utilities in most use cases (Cardinal, 2011).

Keyboards made many impacts on the realm of programming languages. They took programming from stored information on punch cards to being typed on-screen characters. Keyboards made programming more effective, less time consuming, easier to correct mistakes, increased cost-effectiveness, and opened the door for the expansion and invention of new programming languages to be invented. As keyboards improved computer also improved and began to increase their storage capacity, meaning that longer programs could be made and stored than that that could fit on punch cards. The invention of new languages could also be attributed to the invention of the keyboard due to making computer language capable to be put into a form where people could read, learn, and understand computer languages (BASIC). Even the movement of computers becoming personal and entering the everyday home setting was enhanced through the keyboard. Without it, people and businesses would not have been able to adopt computers in their lives and would have stunted the expansion of computing. We would not have the computing technology today without the invention of the keyboard.

New methods of input devices for computing are being made today. For example, gesture control is a new computing input concept that is presenting increasingly possible. Part of the new technology is already in use today in the form of motion control and gesture recognition systems in game consoles like that of Microsoft Kinect (McDonald, 2015). Another example is that of speech recognition with accuracy. There have been astounding advances in artificial intelligence and “deep learning” algorithms that have the potential to soon bring a hands-free computing environment (McDonald, 2015). This would bring about many changes to programming languages and the way people program. There are many projects even today that are heavily involved in making a programming language that will convert human speech into code or even control all aspects of computing in real-time. When this becomes a reality and new programming languages are created for the purpose of turning speech into code, at first the new languages will be very simple at first, then it is believable that the new languages would be increasing time effective if you can simply speak. However, there may be some issues with how to debug and correct code using speech to code. The new languages could also be quite extensive due to the extensiveness of human speech and therefore improved compilers may be needed to process the code back into machine language. This next shift could bring about some very new and intense changes to computing and coding.

As technology advances, the realm of computing follows. Whether it be software or hardware coding and programming languages are affected through the advancement. From the invention of the keyboard to today where speech to code is becoming an increasingly feasible reality, programming languages have been and are evolving to meet the demands within computing. As new discovers are made the programming languages used today will continue to evolve and transition with technology.

Works Cited

*Computer Science – Evolution of Computers in Society*, www.curriculumonline.ie/getmedia/52d35f98-074d-44b5-bcc8-e6c9225240dc/NCCA-The-Evolution-of-Computers-in-Society-LC-SC-programming.pdf.

Brush, Ryan. “Code Is Design.” *97 Things Every Programmer Should Know: Collective Wisdom from the Experts*, by Kevlin Henney, O'Reilly, 2010, pp. 24–25.

Cardinal, David. “From Punchcards to Siri: The History (and Future) of Input Devices.” *ExtremeTech*, 11 Oct. 2011, www.extremetech.com/computing/98287-from-punchcards-to-ipads-the-history-of-input-devices/8?print.

Garson, Edward. “Apply Functional Programming Principles.” *97 Things Every Programmer Should Know: Collective Wisdom from the Experts*, by Kevlin Henney, O'Reilly, 2010, pp. 4–5.

Hope, Computer. “What Is a Punch Card?” *Computer Hope*, 30 Oct. 2017, www.computerhope.com/jargon/p/punccard.htm.

Lane, Malcolm, and Shannon R. “Computer Peripheral Devices.” *AccessScience*, 2014, doi:10.1036/1097-8542.757260.

McDonald, Glenn. “Good-Bye, Keyboard: The Future of Input Devices Is (Almost) Here.” *InfoWorld*, InfoWorld, 22 Jan. 2015, www.infoworld.com/article/2872529/computer-hardware/goodbye-keyboard-the-future-of-input-devices-is-almost-here.html.

Molin, Jesper, et al. “A Comparative Study of Input Devices for Digital Slide Navigation.” *Journal of Pathology Informatics*, vol. 6, no. 1, Feb. 2015, p. 7., doi:10.4103/2153-3539.151894.

Smith, Steve. “Don't Repeat Yourself.” *97 Things Every Programmer Should Know: Collective Wisdom from the Experts*, by Kevlin Henney, O'Reilly, 2010, pp. 60–61.

Taveira, Alvaro D., and Sang D. Choi. “Review Study of Computer Input Devices and Older Users.” *International Journal of Human-Computer Interaction*, vol. 25, no. 5, 2009, pp. 455–474., doi:10.1080/10447310902865040.

VoiceCode. “Advanced Voice-Control Platform.” *Advanced Voice-Control, Speech to Code, Program by Voice, Stop RSI*, 2017, voicecode.io/.