Thomas E. Kurtz & John G. Kemeny
The BASIC Programming Language

Author: David Kilroy <kilroyda@tcd.ie> Date: 2018-10-31

Introduction

Thomas E. Kurtz created the original BASIC language in 1964 alongside his collegue John G. Kemeny. BASIC was developed by Kurtz and Kemeny alongside the creation of the Dartmouth Time Sharing System which was the first large-scale time sharing system to be implemented. Their aim in creating this system was to allow computers to be freely used by ALL students of Dartmouth university, not just scientists and mathematicians.
They focused heavily on giving immediate feedback to the user to make it seems as if each terminal was its own computer. This concept of time-sharing was a great step towards maximising the efficiency of the computers of the 60's, especially in universities.

BASIC, which was created to run on the Dartmouth Time Sharing System, was their most successful culmination of their desire to create a programming language that was usable by people who are not professional programmers. Kurtz argued that the popular languages of the time such as FORTRAN (which BASIC is syntactically similar to) were made to be implemented by professional computer programmers and were not viable to be used by people outside this field.

Kurtz and Kemeny didn't anticipate the growth and popularity of BASIC originally. While they created it for an academic context, companies such as Microsoft and Apple would go on to create their own versions of BASIC to be used in their own computers.

The Spread of BASIC

With the spread of "microcomputers" (or home computers) in the late 70's, the programming language saw huge growth in popularity. By the 1980's, most models of home computer would include a BASIC interpreter. This resulted as a combination of two reasons:

1. It was small enough to fit onto the computers of the day
2. BASIC was [*true to its name*] easy to learn. This lead to widespread use among computer hobbyists and people interested in making computer games. Even my

> own father who is by no means an IT expert nowadays remembers creating BASIC programs on the ZX Spectrum he owned in his youth.

By the end of the 80's, with the creation of faster computers with more processing power, the use of BASIC declined as languages like C and Pascal became more accessible.

Was BASIC a good idea?

Much more interesting than the technology of BASIC however is the idea behind it. Kurtz and Kemeny had the aim of creating a programming language that was more accessible for students of subjects other than Computer Science and Mathematics. BASIC was a clear success in this regard as it became widely mainstream not just in academia but in the personal computer and video game industries which exposed many people to a basic, fundamental understanding of computer programming.

Despite this, there have been skeptics of BASIC and the aim of its creators.
It would be impossible to write an essay on BASIC without addressing the most famous criticism of it by Edsger Dijkstra who said:

> "*It is practically impossible to teach good programming to students that have had prior exposure to BASIC; as potential programmers they are mentally mutated beyond hope of regeneration*" [2.]

In the very same paper, Dijkstra also states:

> "*Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians*"

Should everyone learn coding or should only the best?

In the present day context where I see hyperbolic articles [3.] in the media that describe coding as a necessary skill on the same level as literacy, I am inclined to agree with Dijkstra to some degree. In his paper, he emphasises the importance of programming *well*.
The world we now live in has been entirely dominated by computers. They have become the general machines used by people in nearly every profession that Kurtz and Kemeny envisioned. In this day and age, the skill of knowing how to use a computer is required to some degree or another, but I'm not yet convinced on the fundamental importance of being able to program one if the trade off is that the result is poor code.

BASIC may not have been the greatest attempt at balancing the quality of code with the accessibility of programming but it was a significant contribution into the introduction of people to personal computers and computing in general.

For this reason, I find the work of John G. Kemeny and Thomas E. Kurtz in their creation of BASIC to be no small part of the history of software engineering.

References:

[1.] John G. Kemeny, Thomas E. Kurtz, *Basic: A Manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System* (1964)

[2.] Edsger W. Dijkstra, *How Do We Tell Truths That Might Hurt?* (published 1982)

[3.] John Holden(The Irish Times), *Coding will be the most important skill in the future,* (2017)
https://www.irishtimes.com/business/innovation/coding-will-be-the-most-important-skill-in-the-future-1.3233019

[4.] Laurence Bradford(Forbes), *Why Every Millennial Should Learn Some Code,* (2016)
https://www.forbes.com/sites/laurencebradford/2016/06/20/why-every-millennial-should-learn-some-code/#44adafd070f2