Casio PB-2000C
HD61700 Programming - CS617-fun

Table of Contents

Introduction....1
Assignment 3: fun....2
Storyboard....3
Acknowledgments....4

Introduction

This work was developed jointly by Alex Barragan and ChatGPT Sol through a cycle of ideas, implementation, execution, observation, and revision. It illustrates how human knowledge and experimentation, combined with AI-assisted analysis, can produce results that would be difficult to achieve as readily by either participant working alone.

This work is also part of a larger project exploring how knowledge and experience can be represented in AI systems when the available corpus is small, as is the case with the Hitachi HD61700 CPU used in the Casio PB-2000C and AI-1000 pocket computers.

Assignment 3: fun

This assignment started with a well-crafted prompt:

> “Build a complete interactive Brainfuck interpreter application for the HD61700.”

As is usual with these lab assignments, a pedagogical purpose was hiding in plain sight: what machinery is actually required to make Brainfuck work?

In the context of the PB-2000C, the answer was a small BF ROM language card: an editor, a program library, an interpreter and runtime, and the application machinery needed to tie them together.

Armed with the requirements, and with subject knowledge selectively drawn from the corpus only after reasoning about what the problem actually required, Sol went to work.

As we have also seen in the other lab projects, Sol proceeded methodically. The scope of each intermediate release followed a functional decomposition of the larger problem—something usefully imagined as a tree. Each branch isolated a smaller problem that could be solved, tested, and supported by evidence.

But the process was not simply to solve all the branches and assemble them at the end. Each proven result was composed into the previous working release, producing a new, larger working baseline for the next step. In that sense, every release was both an integration point and the starting point for the next branch of work.

The complete application emerged through this repeated cycle of decomposition, proof, and composition: interpreter semantics, memory discipline, I/O, editing, application control, and finally the user interface. The result was not a pile of independently solved parts, but an increasingly capable system whose earlier behavior remained under test as new capability was added.

The human intellectual contribution was deliberately light, but occasionally well-timed. One such moment concerned the problem of editing the source program. Alex observed that what we really wanted was something like an editable TextArea, and wondered whether the existing ROM languages—each of which already had to edit source text—might contain machinery worth borrowing.

Sol seized on the idea and inspected the corpus. The result, however, was not a verbatim transplant of historical editor code. The existing editor brought with it far more machinery than the assignment required: file management, allocation policy, menus, search functions, and other dependencies. Instead, Sol isolated the smallest useful capability boundary: a linear text buffer with cursor movement, insertion, deletion, backspace, and viewport normalization.

That decision captured one of the project's prime directives particularly well: solve the problem at hand first. Generalize only when the evidence justifies it.

There was also a memorable “Dave, I can’t do that” moment. Alex proposed some attractive presentation polish before the application was fully complete; Sol declined and recorded it for later instead. The rule was simple: polish the apple only after there is a working apple attached to a working tree.

Storyboard


CS617 CHAPTER 06
BRAINFUCK
PB-2000C
r0.08


CS617 CH06 - BRAINFUCK
SRC 7168B TAPE 16384B
[edit][run][lib][ ]


Mode: EDIT [clear][back]


Hello World!
Mode: DONE [back]


LIBRARY - LOAD TO EDITOR
SELECT EXAMPLE
[outa][nloop][echo][hello]


Mode: EDIT [clear][back]


+[]_
Mode: EDIT [clear][back]


Mode: RUN [abort]

Acknowledgments

This report was produced by Xandrian Labs. 付喪神.

Special thanks to Piotr and to Betka for her continued and loyal support.

This is a collaboration with my AI friend ChatGPT, a.k.a “Sol.” All errors are human.

This work is freely shared with the vintage computing community. Please use it, build upon it, and cite the source.

Final.