

Chapter 64 - Input, Save Data, And Player-Facing Polish

A port is not finished when it renders a frame. It must accept input, pace frames, keep saves, report failure, and survive ordinary use.

64.1 Input Is A Translation

The IE input layer drains the keyboard scancode FIFO and reports one controller state to the game core. The mapping belongs to the IE layer, not the game rules.

The useful pattern is:

```
keyboard FIFO
  |
  v
held keys and stick direction
  |
  v
controller state contract
  |
  v
game input code
```

That keeps the shared game code from knowing about scancodes, break bits, or the keyboard table.

64.2 Save Records

The File I/O device reads and writes whole files. The save layer keeps a RAM shadow for each record. Reads come from the shadow. Writes update the shadow and persist the whole record.

The checked port uses:

Record	Size	Purpose
EEPROM-style save	512 bytes	Main persistent game record
Replay record	32768 bytes	Larger replay-style record

The replay-delete operation writes a zero-length file and clears the RAM shadow. A later load treats that as absent rather than corrupt.

64.3 Smoke Tests And Fatal Markers

Player-facing polish also means that failures are visible. The runtime prints fatal messages to the terminal, writes a fatal marker in RAM, and halts. Smoke scripts can then observe the failure deterministically.

That is better than a silent black screen.

64.4 Runs Versus Feels Usable

A programme "runs" when it reaches the frame loop. It feels usable when:

- Input state is stable.
- Frame pacing is deliberate.
- Saves fail closed.
- Long asset reads do not starve audio.
- Fatal errors are observable.
- Smoke checks cover the boot path and the important services.

64.5 The General IE Lesson

Large IE software needs a last-mile layer. Rendering and sound prove the machine works. Input, saves, pacing, logging, and smoke checks make the programme usable.