

The origin of Spacewar

J. M. Graetz

This article is reprinted from the August, 1981 issue of *Creative Computing* magazine. *Creative's* policy at the time allowed reprinting of articles in newsletters and nonprofit publications provided this notice was reproduced:

Copyright © 1981 by Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950. Sample issue \$2.50, 12-issue subscription \$20.

(but they went out of business in 1985, so there's not much point in writing off for your sample issue).

I. BEFORE SPACEWAR

The Lensman, The Skylark, and the Hingham Institute

It's Kimball Kinnison's fault. And Dick Seaton's. Without the *Gray Lensman* and the *Skylark of Space* there would be nothing to write about. So most of the blame falls on E. E. Smith, but the Toho Film Studios and the American Research and Development Corp. have something to answer for as well. If Doc Smith had been content designing doughnuts, if American-International Pictures had stuck to beach blanket flicks, if (most of all) General Doriot hadn't waved money in front of Ken Olsen in 1957, the world might yet be free of **Spacewar!**

It all came together in 1961 at the Hingham Institute, a barely-habitable tenement on Hingham Street in Cambridge, MA. Three Institute Fellows were involved: Wayne Witanen, mathematician, early music buff, and mountain climber; J. Martin Graetz (which is me), man of no fixed talent who tended to act superior because he was already a Published Author; and Stephen R. (Slug) Russell, specialist in steam trains, trivia, and artificial intelligence. We were all about 25 (the more or less to be the same).

At the time, we were crashing and banging our way through the "Skylark" and "Lensman" novels of Edward E. Smith, PhD., a cereal chemist who wrote with the grace and refinement of a pneumatic drill. These stories are pretty much all of a piece: after some preliminary foofaraw to get everyone's name right, a bunch of overdeveloped Hardy Boys go trekking off through the universe to punch out the latest gang of galactic goons, blow up a few planets, kill all sorts of nasty life forms, and just have a heck of a good time.

In a pinch, which is where they usually were, our heroes could be counted on to come up with a complete scientific theory, invent the technology to implement it, build the tools to implement the technology, and produce the (usually) weapons to blow away the baddies, all while being chased in their spaceship hither and thither through the trackless wastes of the galaxy (he wrote like that) by assorted Fenachrone, Boskonians, and the World Steel Corporation.

Is that enough to turn the mind to margarine? It is not. In breaks between books, we would be off to one of Boston's seedier cinemas to view the latest trash from Toho. In the days before Mazdas and Minoltas, the Japanese (and occasionally the British and Californians) churned out a steady diet of cinematic junk food of which Rodan and Godzilla are only the best known examples. These movies depended for their effects on high quality modelwork, oceans of rays, beams, and explosions, and the determined avoidance of plot, character, or significance. They were the movie equivalent of *The Skylark of Space*.

If that's the case, we asked ourselves, why doesn't anyone make *Skylark* movies? Hearing no reply (our innocence of current film technology, economics, and copyright laws was enormous), we often passed the time in the Hingham

Street common room in deep wishful thought, inventing special effects and sequences for a grand series of space epics that would never see a sound stage. Nonetheless, these books, movies, and bull-sessions established the mind-set that eventually led to **Spacewar!**

When Computers Were Gods

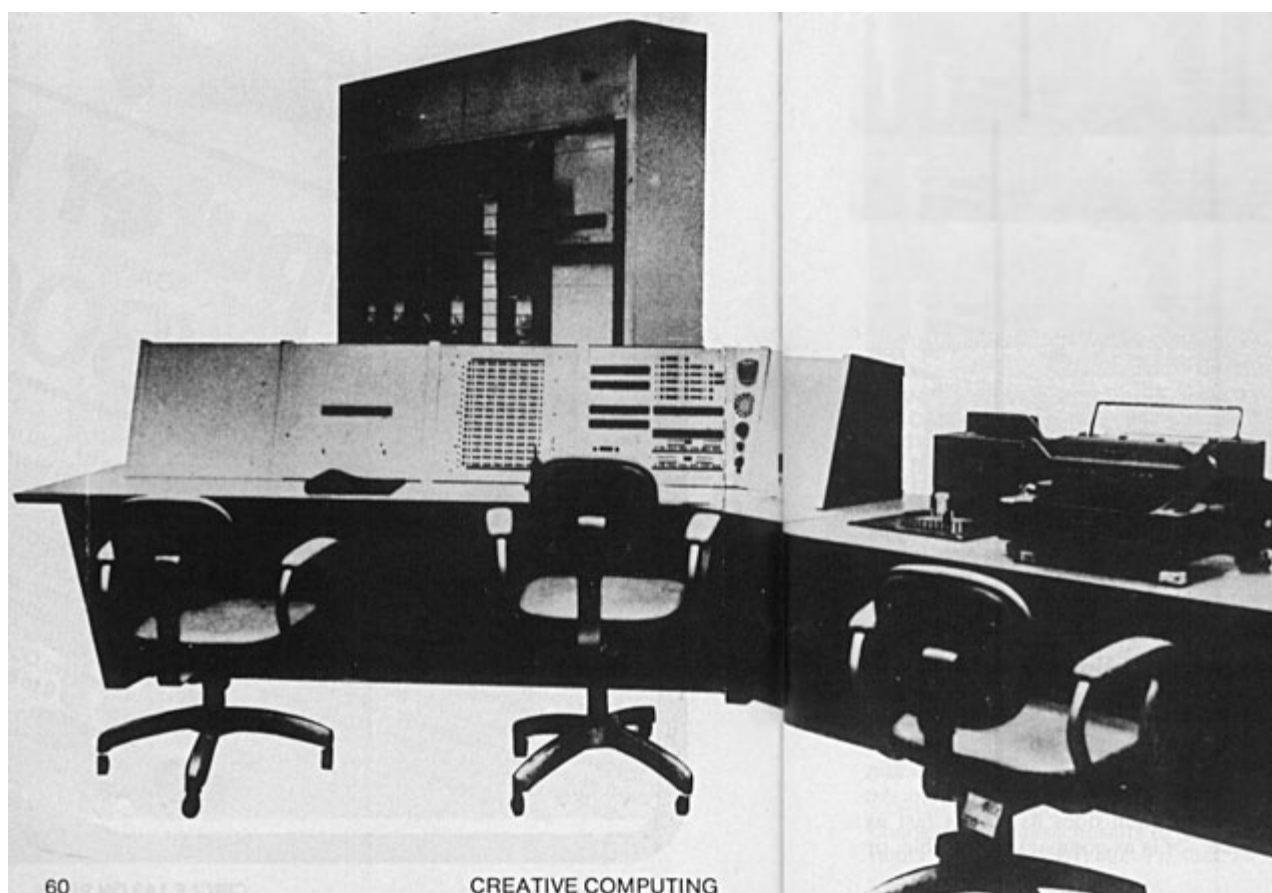
In early 1961 Wayne, Slug, and I, by no coincidence, were all working at Harvard University's Littauer Statistical laboratory. A large part of our jobs was to run statistics computations for various Harvard persons. The agent of choice for this work was an IBM 704.

To a generation whose concept of a computer is founded on the Z80 chip, it may be hard to visualize a 704 or to comprehend the place it held in the public imagination (along with UNIVAC) as the type specimen of what a computer was: a collection of mysterious hulking gray cabinets approachable only through the intercession of The Operator. In the specially built computer room, The Operator set switches, pushed buttons, and examined panels of flashing lights, while his Assistants attended various whirring, clanking, and chattering devices, rushing to and fro with stacks of cryptically-printed paper, decks of weirdly-punched cards, and reels of recondite brown ribbon, all to the background hum of The Machine. Add a little incense and a few candles, and you could be forgiven for thinking these were the rites of some oracular shrine.

Everything about the 704, from the inscrutable main frame to the glowing tubes (yes, tubes!) in the glass-walled core memory case, proclaimed that this was a Very Complicated System operated only by Specially Trained Personnel, among whom programmers and other ordinary mortals were not numbered. In short, a computer was something that you simply did not sit down and fool around with.

A Stone's Throw from Olympus

In the summer of 1961 I went to work for Professor Jack B. Dennis, who was then the proprietor of the TX-0, a machine that to me was only slightly less legendary than its ancestor, Whirlwind. The TX-0 was transistorized, and while solid-state computers were beginning to appear on the market, the "Tixo" was the original. Even in 1961 it was acknowledged to be a historically important research facility; many of the programs developed on the TX-0, such as Jack Dennis's MACRO assembler and Thomas Stockham's FLIT debugging program, were the first of their kind. So the chance to work on this computer was in many ways a rite of passage; it meant that I had joined the ranks of the Real Programmers.



While hardly your average populist Apple, the TX-0 was definitely a step away from the Computer-as-Apollo. Instead of being sealed into its own special chapel, it sat at one end of a typical large, messy MIT research space. With its racks of exposed circuitry, power supplies and meters, and its long, low L-shaped console, the TX-0 looked for all the world like the control room of a suburban pumping station. And the thing of it was, you were expected to *run it yourself*.

So here was the former 704 Operator's Assistant pushing buttons, flipping switches, and pressing keys to make his own programs work. In some ways it was simpler than the 704; for one thing, there wasn't a battery of clanking mechanical monsters. The TX-0's input and output medium was something called a Flexowriter: an all-in-one keyboard, printer, paper-tape reader and punch, that worked like a mule and had a personality to match. There was also a "high-speed" paper tape reader, a Grand Prix whiz that could read programs into memory almost as fast as the cassette-tape reader on a TRS-80.

And the TX-0 had a Scope. Now console-mounted, programmable CRTs were not unheard of at that time but they were generally slow, inflexible, and awkward to program. The TX-0 scope, on the other hand, was easy to use; you could generate a useful display with fewer than a dozen instructions. And if that weren't enough, there was a magic wand: the light pen. (The importance of these two devices can't be overemphasized; Ivan Sutherland used the scope and the pen to develop his original "Sketchpad.")

That was the TX-0, the world's first on-line computer, and the training ground for the designers and programmers of later generations of hands-on machines. The first computer bums -- hackers -- were the products of this training; without it, and them, there would have been no **Spacewar!**

Tixo's people

The users of the TX-0 were a melange of students, staff researchers and professors with not much in common other than their need for large amounts of largely unstructured computer time. The feel of the place, however, was established by the hackers -- mostly students, but including a professor or two -- whose lives seemed to be organized in 18-bit strings. Many of them worked for Professors John McCarthy and Marvin Minsky in the Artificial Intelligence Group, an odd

bunch (even by MIT standards) who seemed convinced that given enough random-access memory and a really fast cycle time you could model the cognitive parts of the brain and hey presto! a real thinking machine. Others worked for Professor Dennis, who presided over the use and development of the TX-0 and more or less benignly kept a semblance of order. The man who kept it all running was a soft-spoken, white-haired gentleman, John McKenzie, the chief engineer.

Out of this cloud of computer bums emerged the group that brought **Spacewar!** to the silver (well, light gray) screen: Dan Edwards (AI Group), Lisp specialist; Alan Kotok (TX-0 staff), who wrote MIDAS, the successor to MACRO; Peter Sampson (AI Group), who made the Tixo and PDP-1 play Bach; and Steve Russell and I.

"You Mean That's All It Does?"

When computers were still marvels, people would flock to watch them still at work whenever the opportunity arose. They were usually disappointed. Whirring tapes and clattering card readers can hold one's interest for only so long. They just did the same dull thing over and over; besides, they were obviously mechanical -- at best, overgrown record changers -- and thus not mysterious. The main frame, which did all the marvelous work, just sat there. There was nothing to *see*.

On the other hand, something is always happening on a TV screen, which is why people stare at them for hours. On MIT's annual Open House day, for example, people came to stare for hours at Whirlwind's CRT screen. What did they stare at? Bouncing Ball.

Bouncing Ball may be the very first computer-CRT demonstration program. It didn't do much: a dot appeared at the top of the screen, fell to the bottom and bounced (with a "thok" from the console speaker). It bounced off the sides and floor of the displayed box, gradually losing momentum until it hit the floor and rolled off the screen through a hole in the bottom line. And that's all. Pong was not even an idea in 1960. (Note: Well, maybe not Pong, but something very much like it. Watch these pages. --DHA)

The TX-0's counterpart to Bouncing Ball was the Mouse in the Maze, written by Douglas T. Ross and John E. Ward. Essentially, it was a short cartoon: a stylized mouse searched through a rectangular maze until it found a piece of cheese which it then ate, leaving a few crumbs. You constructed the maze and placed the cheese (or cheeses -- you could have more than one) with the light pen. A variation replaced the cheese with a martini; after drinking the first one the mouse would stagger to the next.

Besides the Mouse, the TX-0 also had HAX, which displayed changing patterns according to the settings of two console switch registers. Well-chosen settings could produce interesting shapes or arrangements of dots, sometimes accompanied by amusing sounds from the console speaker. The console speaker is a phenomenon whose day seems to have passed. More than just a plaything, for the experienced operator the speaker was a valuable guide to the condition of a running program.

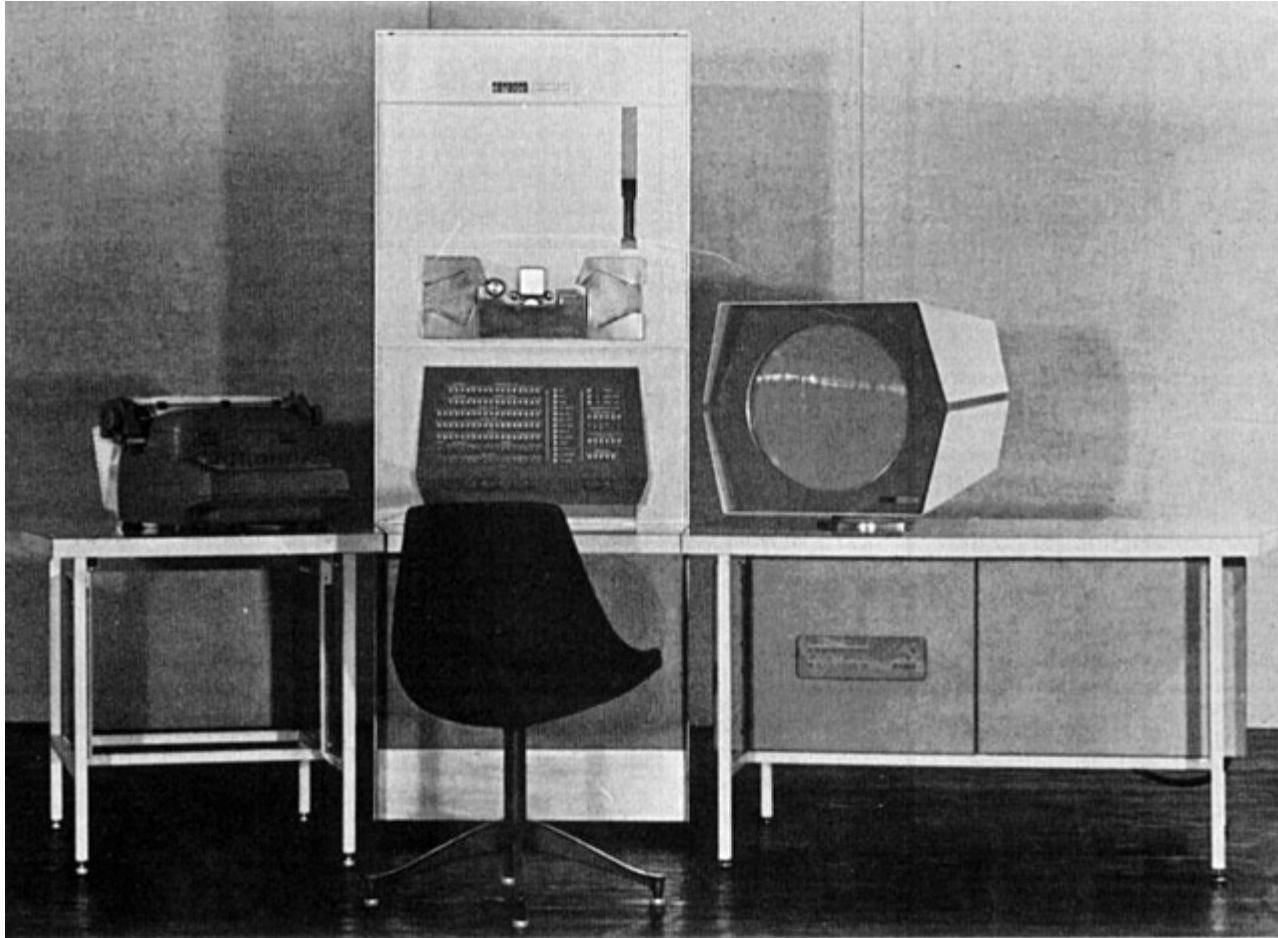
Finally, there was the inevitable Tic-Tac-Toe, with the user playing the computer. The TX-0 version used the Flexowriter rather than the scope. (The game is so simple to analyze that there was even a version for the *off-line* Flexo.)

These four programs pointed the way. Bouncing Ball was a pure demonstration: you pushed the button, and it did all the rest. The mouse was more fun, because you could make it different every time. HAX was a real toy; you could play with it while it was running and make it change on the fly. And Tic-Tac-Toe was an actual game, however simpleminded. The ingredients were there; we just needed an *idea*.

The World's First Toy Computer

For all its homeliness, the TX-0 was still very much a god. It took up lots of space, it had to be carefully tended, it took special procedures to start it up and shut it down, and it cost a lot of money to build.

All this changed in the fall of 1961, when the first production-model PDP-1 was installed in the "Kluge Room" next door to the TX-0. It had been anticipated for months; an early brochure announcing the machine (as well as a couple of no-shows called the PDP-2 and PDP-3, in case you had been wondering about that) had been circulating in the area for a while. It was clear that the PDP-1 had TX-0 genes; the hackers would be right at home.



The -1 would be faster than the Tixo, more compact, and *available*. It was the first computer that did not require one to have an E.E. degree and the patience of Buddha to start it up in the morning; you could turn it on any time by flipping one switch, and when you were finished you could turn it off. We had never seen anything like that before.

II. SPACEWAR! BEGUN

The Hingham Institute Study Group On Space Warfare

Long before the PDP-1 [original says PDP-11, an obvious typo] was up and running Wayne, Slug, and I had formed a sort of ad-hoc committee on what to do with it -- it being the Type 30 Precision CRT display which was scheduled to be installed a couple of months after the computer itself. It was clear from the start that while the Ball and Mouse and HAX were clever and amusing, they really weren't very good as demonstration programs. Why not? Zooming across the galaxy with our Bergenholm Inertialess Drive, the Hingham Institute Study Group on Space Warfare devised its Theory of Computer Toys. A good demonstration program ought to satisfy three criteria:

1. It should *demonstrate* as many of the computer's resources as possible, and tax those resources to the limit;
2. Within a consistent framework, it should be interesting, which means every run should be different;
3. It should involve the onlooker in a pleasurable and active way -- in short, it should be a game.

With the Fenachrone hot on our ion track, Wayne said, "Look, you need action and you need some kind of skill level. It

should be a game where you have to control things moving around on the scope, like, oh, spaceships. Something like an explorer game, or a race or contest... a fight, maybe?"

"SPACEWAR!" shouted Slug and I, as the last force screen flared into the violet and went down.

The basic rules developed quickly. There would be at least two spaceships, each controlled by a set of console switches ("Gee, it would be nice to have a joystick or something like that..."). The ships would have a supply of rocket fuel and some sort of a weapon: a ray or beam, possibly a missile. For really hopeless situations, a panic button would be nice... hmmm... aha! Hyperspace! (What else, after all, is there?) And that, pretty much, was that.

The Hackers Meet SPACEWAR!

By the end of the summer of 1961, Steve Russell had returned to the Artificial Intelligence Group (he'd worked there before Littauer); consequently, whatever ideas the Study Group came up with were soon circulating among the hackers. **Spacewar!** was an appealing, simple concept, and the hackers were the appealingly simple people to bring it to life. First, however, there was the small matter of software.

The PDP-1 was a no-frills machine at the beginning; except for a few diagnostic and utility routines, there was no program library. In a way this suited the hackers just fine; here was a chance to improve on TX-0 software and to write new stuff that couldn't have been done before. MACRO and FLIT were translated from TXish to PDPese, FLIT becoming the first in a continuing line of DDT on-line debugging programs. Steve Piner wrote a text display and editing program called Expensive Typewriter (For a while, "expensive" was a favorite adjective for naming various PDP-1 routines that imitated the functions of more mundane devices. Among them was Peter Samson's E. Planetarium, as we shall see.), another original whose lineage you can trace, if you like, right down to the latest word processors.

With the software taken care of we could write real programs, which is to say toys. Bouncing Ball was successfully converted to PDP-1 use, but HAX, for some reason, was not. But no one really missed it, because we had a brand-new toy invented by Professor Marvin Minsky. The program displayed three dots which proceeded to "interact," weaving various patterns on the scope face. As with HAX, the initializing constants were set in the console switches. Among the patterns were geometric displays, Lissajous-like figures, and "fireworks." Minsky's program title was something like "Tri-Pos: Three-Position Display," but from the beginning we never called it anything but The Minskytron ("tron" was the In suffix of the early 1960s.)

The classic needle and wedge space ship outlines and the opposite-quadrant starting positions were established at this stage, as shown in Figure 1. Acceleration was realistic; it took time to get off the mark, and to slow down you had to reverse the ship and blast in the other direction; the rocket exhaust was a flickering "fiery tail." Rotation, on the other hand, was by something we called "gyros" -- a sort of flywheel effect invented to avoid consideration of messy things like moments of inertia. I guess they were really rotational Bergenholms.

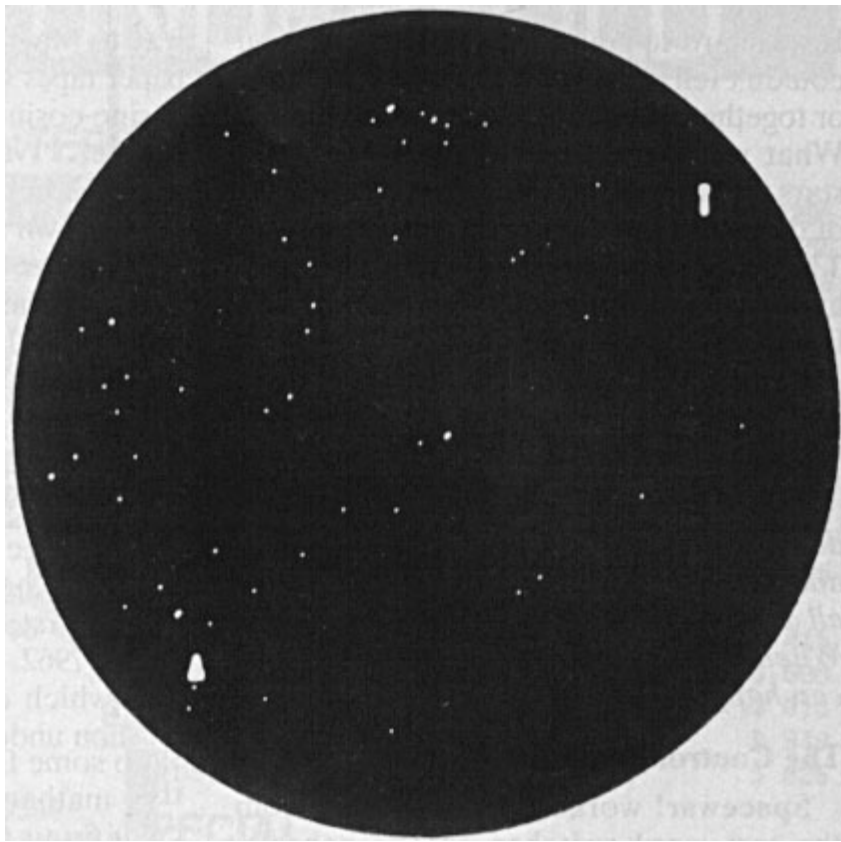


Figure 1. The Starting Position. The ships are in the centers of diagonally opposite quadrants. The vee of stars at top center is the horns of Taurus. You should be able to pick out the stars of Orion at the left (the bright star just above the wedge-ship is Rigel).

It was apparent almost immediately that the featureless background was a liability. It was hard to gauge relative motion; you couldn't tell if the ships were drifting apart or together when they were moving slowly. What we needed, obviously, were some stars. Russell wrote in a random display of dots, and the quality of play improved. The only thing left, we thought, was hyperspace, and that was on the way. In fact, we'd just begun.

III. SPACEWAR! COMPLETE

Please keep in mind that what follows did not happen in a neat first-one-thing-and-then-the next progression, but rather all at once in a period of about six weeks. When hackers are aroused, anything that can happen will.

The Control Boxes

Spacewar! worked perfectly well from the test word switches on the console, except that the CRT was off to one side, so one player had a visual advantage. More to the point, with two excitable space warriors jammed into a space meant for one reasonably calm operator, damage to the equipment was a constant threat. At the very least, a jittery player could miss the torpedo switch and hit the start lever, obliterating the universe in one big anti-bang. A separate control device was obviously necessary, but joysticks (our original idea) were not readily available in 1962. So Alan Kotok and Robert A. Saunders, who just happened to be members of the Tech Model Railroad Club, trundled off to the TMRC room, scrabbled around the layout for a while to find odd bits of wood, wire, bakelite, and switch-board hardware, and when the hammering and sawing and soldering had ceased, there on the CRT table were the first **Spacewar!** control boxes (Figure 2. These boxes have long since disappeared, but the sketch is a reasonably accurate reconstruction).

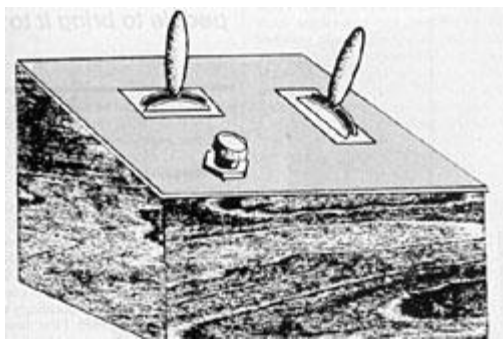


Figure 2. The original control boxes looked something like this. The controls are a) right-left rotation, b) acceleration (pulled back) and hyperspace (pulled forward), and c) torpedo button.

The box is wood with a Bakelite top. The two switches are double-throw; the button is a silent momentary switch. Their functions are as follows:

a. Rotation control.

It is pushed to the left to rotate the ship counterclockwise, to the right to rotate clockwise.

b. A two-function control.

Pulled back it is the rocket accelerator; the rocket continues to blast as long as the switch is thrown. Pushed forward, the switch is the hyperspace control, as described below.

c. The torpedo button.

It had to be silent so that your opponent could not tell when you were trying to fire. (There was a fixed delay between shots "to allow the torp tubes to cool" and fire was not automatic; you had to keep pushing the button to get off a missile.)

With the control boxes, players could sit comfortably apart, each with a clear view of the screen. That, plus the carefully designed layout of the controls, improved ones playing skills considerably, making the game even more fun.

First Steps

By the end of 1961, all the elements were in place: a brand new, available computer, a cloud of hackers, tolerant when not actively implicated employers, and an exciting idea. Slug Russell was getting the head from everyone to "do something" about **Spacewar!** (I was in a different department at MIT by this time, and Wayne, alas, was one of those unlucky Army Reservists called to active duty during the Berlin Wall panic in October. He never got to participate in developing his own idea.)

Russell, never one to "do something" when there was an alternative, begged off for one reason or another. One of the excuses for not doing it, Slug remembers, was "Oh, we don't have a sine-cosine routine and gee, I don't know how to write a sine-cosine routine..." Then Alan Kotok came back from a trip all the way to Maynard (DEC headquarters) with paper tapes saying "All right, Russell, here's a sine-cosine routine; now what's your excuse?" "Well," says Slug, "I looked around and I didn't find an excuse, so I had to settle down and do some figuring."

With the heavy mathematics in hand, Slug produced the first object-in-motion program in January 1962. This was nothing more than a dot which could accelerate and change direction under switch control. Even without a hardware multiply-divide capability (on the early PDP-1s, anything stiffer than integer addition and subtraction had to be done by subroutine) the computer was clearly not being pushed.

From dot to rocket ship was a surprisingly easy step: "I realized," Slug says, "that I didn't have to worry about the speed of the sine-cosine routine, because there were only two angles involved in each frame -- one for each ship. Then the idea of rotating the grid came out." The ship outlines were represented as a series of direction codes starting from the nose of the ship; when the ship was vertical and tail-down, each code pointed to one of the five possible adjacent dots could be displayed next. To display the ship at an angle, Russell calculated the appropriate sine and cosine and added them to the original direction code constants, in effect rotating the entire grid. With this method, the ship's angle had to

be calculated only once in each display frame. The outline codes were kept in a table so that different shapes could be tried out at will, but this meant the table had to be searched every frame to generate the outline. As the game developed, this arrangement proved to be a sticking point which we shall see was neatly solved by Dan Edwards.

By February, the first game was operating. It was a barebones model: just the two ships, a supply of fuel, and a store of "torpedoes" -- points of light fired from the nose of the ship. Once launched, a torpedo was a ballistic missile, zooming along until it either hit something (more precisely, until it got within a minimum distance of a ship or another torpedo) or its "time fuse" caused it to self-destruct.

The Stars of the Heavens

One of the forces driving the dedicated hacker is the quest for elegance. It is not sufficient to write programs that work. They must also be "elegant," either in code or in function -- both, if possible. An elegant program does its job as fast as possible, or is as compact as possible, or is as clever as possible in taking advantage of the particular features of the machine in which it runs, and (finally) produces its results in an aesthetically pleasing form without compromising either the results or operation of other programs associated with it.

"Peter Samson," recalls Russell, "was offended by my random stars." In other words, while a background of miscellaneous points of light might be all very well for some run-down jerkwater space fleet, it just wouldn't do for the Galactic Patrol. So Peter Samson sat down and wrote "Expensive Planetarium."

Using data from the American Ephemeris and Nautical Almanac, Samson encoded the entire night sky (down to just above fifth magnitude) between $22\frac{1}{2}^{\circ}$ N and $22\frac{1}{2}^{\circ}$ S, thus including most of the familiar constellations. The display can remain fixed or move gradually from right to left, ultimately displaying the entire culinder [cylinder?] of stars. The elegance does not stop there. By firing each display point the appropriate number of times, Samson was able to produce a display that showed the stars at something close to their actual relative brightness. An attractive demonstration program in its own right, E.P. was "duly admired and inhaled into **Spacewar!**"

The Heavy Star

Up to this point, **Spacewar!** was heavily biased towards motor skills and fast reflexes, with strategy counting for very little. Games tended to become nothing more than wild shootouts, which was exciting but ultimately unrewarding. Some sort of equalizer was called for.

Russell: "Dan Edwards was offended by the plain spaceships, and felt that gravity should be introduced. I pleaded innocence of numerical analysis and other things" -- in other words, here's the whitewash brush and there's a section of fence -- "so Dan put in the gravity calculations."

The star blazed forth from the center of the screen, its flashing rays a clear warning that it was not to be trifled with. Its gravity well encompassed all space; no matter where you were, if you did not move you would be drawn into the sun and destroyed. (As a gesture of good will towards less skillful or beginning players, a switch option turned annihilation into a sort of hyperstatal translation to the "anti-point," i.e., the four corners of the screen.)

The star did two things. It introduced a player-independent element that the game needed; when speeds were high and space was filled with missiles, it was often sheer luck that kept one from crashing into the star. It also brought the other elements of the game into focus by demanding strategy. In the presence of gravity both ships were affected by something beyond their control, but which a skillful player could use to advantage.

The first result of this new attention to strategy was the opening move in Figure 3, which was quickly dubbed the "CBS opening" because of its eye-like shape. It took a while to learn this maneuver but it soon became the standard opening among experienced players, as it generally produced the most exciting games.

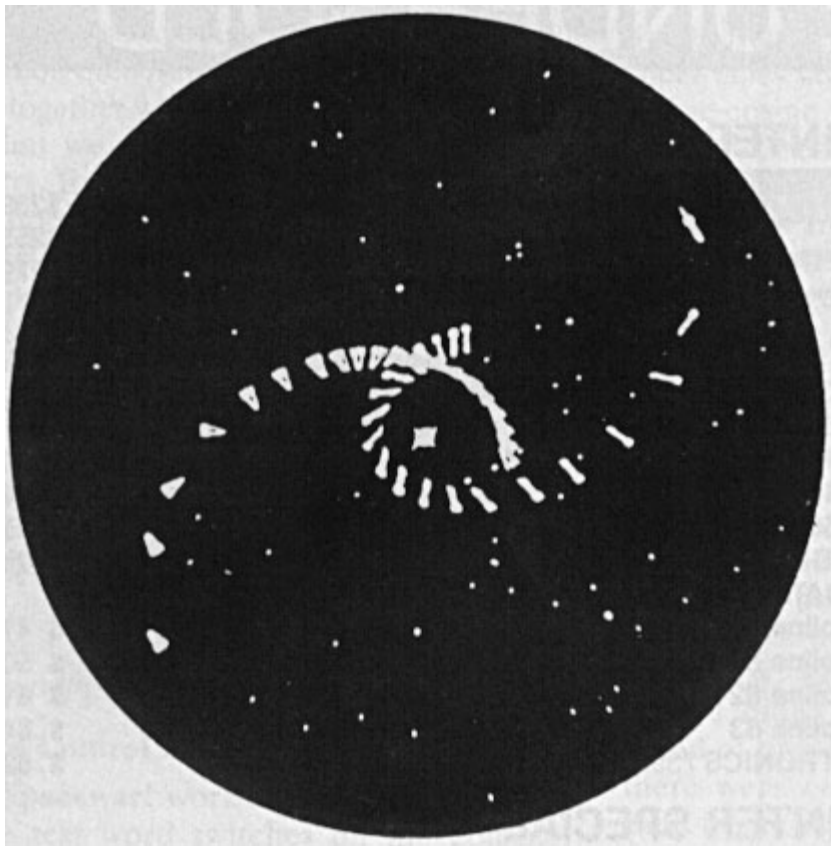


Figure 3. The "CBS Opening." The ships turn slightly away from the star and fire a short rocket blast (note the needle-ship's exhaust) to get into a comet-type orbit, then rotate the other way to try shooting torpedoes at the opponent.

The addition of gravity pushed **Spacewar!** over the edge of flicker-free display. To get back under the limit, Dan Edwards devised an elegant fiddle to speed up the outline display routine.

In Russell's original program, the outline tables were examined and interpreted in every display frame, an essentially redundant operation. Edwards replaced this procedure with an outline "compiler," which examined the tables at the start of a game and compiled a short program to generate the outline for each ship. This dramatically reduced calculation time, restoring the steady display and making room for the last of the original bells and whistles.

Hyperspace

While all this was going on, I was in my secret hideaway (then known as the Electronic Systems Lab) working on the ultimate panic button: hyperspace. The idea was that when everything else failed you could jump into the fourth dimension and disappear. As this would introduce an element of something very like magic into an otherwise rational universe, the use of hyperspace had to be hedged in some way. Our ultimate goal was a feature that, while useful, was not entirely reliable. The machinery, we said, would be the "Mark One Hyperfield Generators... hadn't done a thorough job of testing... rushed them to the fleet" and so on. They'd be good for one or two shots, but would deteriorate rapidly after that. They might not work at all ("It's not *my* fault, Chewie!") or if they did, your chances of coming back out intact were rather less than even. Slug: "It was something you could use, but not something you wanted to use."

The original hyperspace was not that elegant. "MK I unreliability" boiled down to this: you had exactly three jumps. In each jump your ship's co-ordinates were scrambled so that you never knew where you would reappear -- it could be in the middle of the sun. You were gone for a discernible period of time, which gave your opponent a bit of a breather, but you came back with your original velocity and direction intact. To jump you pushed the blast lever forward.

Hyperspace had one cute feature (well, I thought it was cute). Do you remember the Minskytron? One of its displays looked very much like a classical Bohr atom, which in those days was an overworked metaphor for anything to do with

space and science-fiction. Reasoning that a ship entering hyperspace would cause a local distortion of space-time resulting in a warp-induced photonic stress emission (see how easy this is?), I made the disappearing ship leave behind a short Minskytron signature (Figure 4).

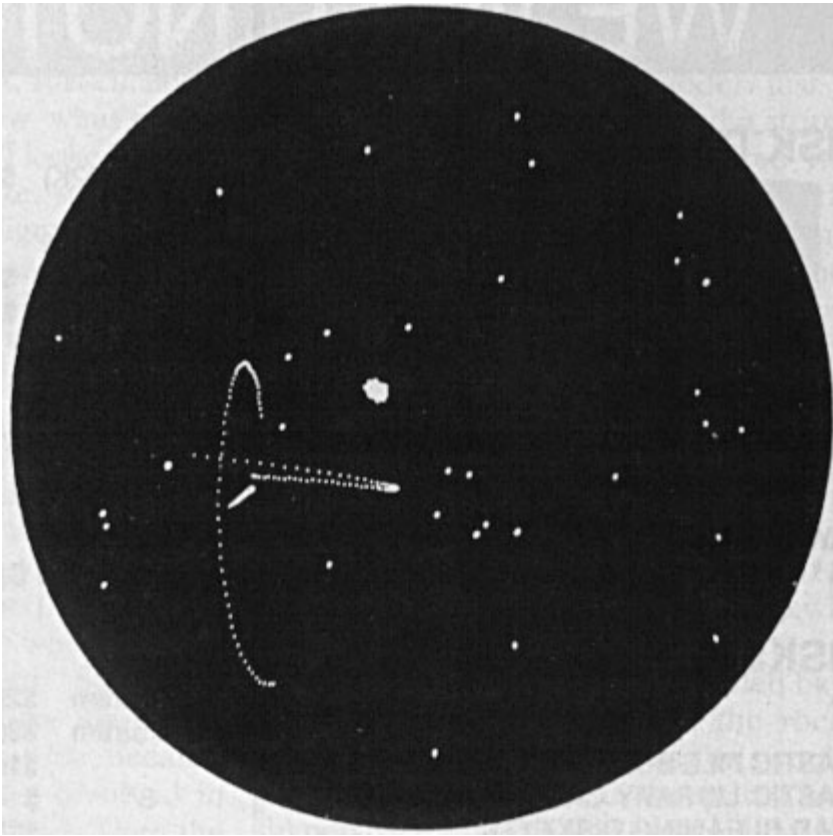


Figure 4. "Warp-induced photonic stress emission." The Hyperspace Minskytron signature.

Crocks and Loose Ends

In retrospect, it is remarkable that the original **Spacewar!** managed to include so many features, given the limitations of our PDP-1: 4K words (about 9K bytes) of memory, an instruction cycle time of five microseconds, and a subroutine multiply-divide. It's hardly surprising, then, that we had to let a few unsatisfactory (all right, inelegant) bits go by.

The most irritating of these (and the first to be improved in later versions) was the appropriately-named Crock Explosion. Something dramatic obviously had to happen when a ship was destroyed, but we were dealing with a plain dot-matrix screen. The original control program produced a random-dot burst confined within a small square whose outlines were all too discernible (Figure 5).

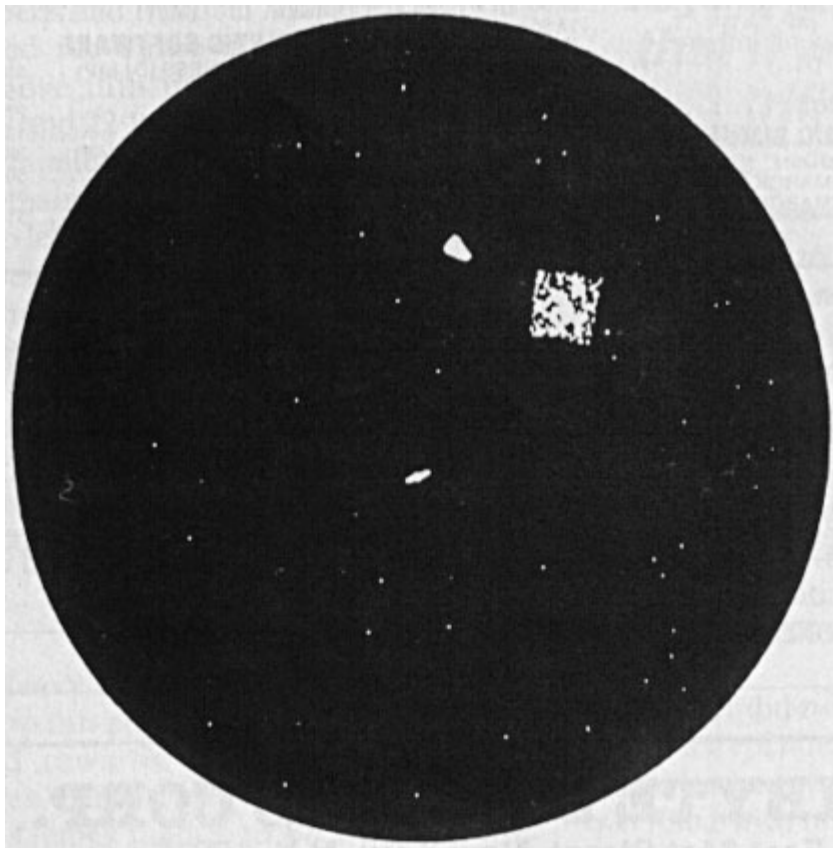


Figure 5. The Crock Explosion. Nobody's perfect.

This explosion was intended merely as a place-holder until something more plausible could be worked out, but after all the other features had been "inhaled," there wasn't room or time for a fancier calculation.

Similarly, the torpedoes were not quite consistent with the **Spacewar!** universe after the heavy star was in place. The gravity calculations for two ships was as much as the program could handle; there was no time to include half a dozen missiles as well. So the torpedoes were unaffected by the star, with the odd result that you could shoot right through it and hit something on the other side (if you weren't careful getting round the Star, it could be you). We made the usual excuses... mumblemumble photon bombs mumblemumble... but no one really cared.

The heavy star itself was not entirely Newtonian. The common tactic of plunging down the gravity well to gain momentum by whipping around the sun (Figure 6) gave you somewhat more energy than you were really entitled to. As this just made the game more interesting, nothing was immediately done to correct it.

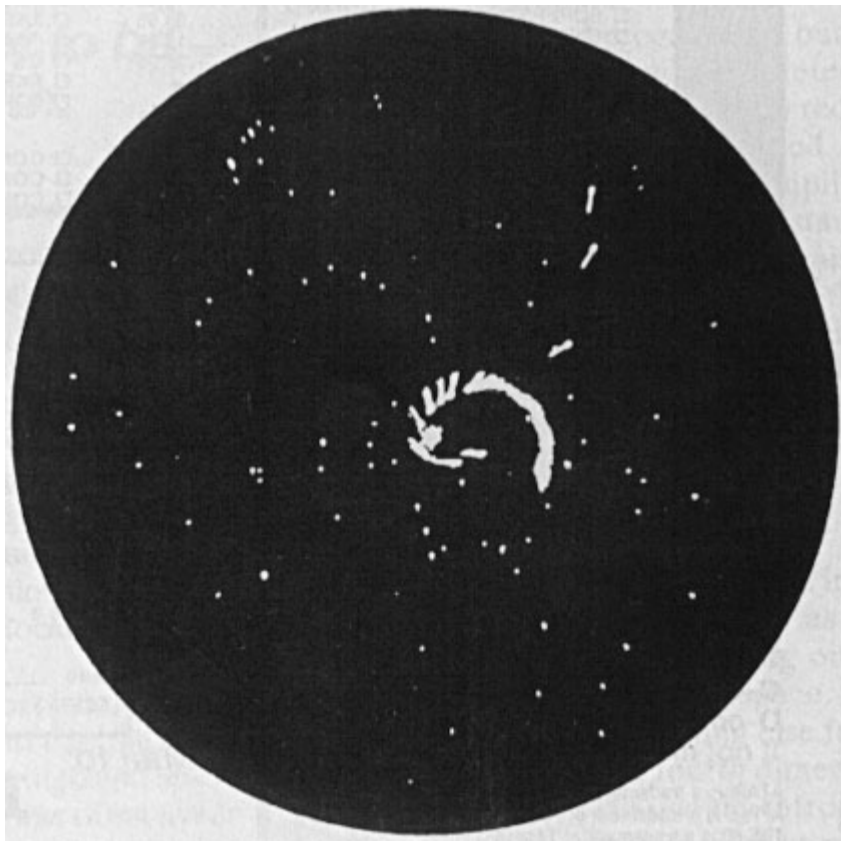


Figure 6. A common mid-game flourish -- and don't talk about G-forces!

IV. AFTER SPACEWAR!

The game was essentially complete by the end of April, 1962. The only further immediate work was to make **Spacewar!** presentable for MIT's annual Science Open House in May. A scoring facility was added so that finite matches could be played, making it easier to limit the time any one person spent at the controls. To provide for the crowds that we (accurately) anticipated, a large screen laboratory CRT was attached to the computer to function as a slave display. Perched on top of a high cabinet, it allowed a roomful of people to watch in relative comfort.

Also in May, the first meeting of DECUS (Digital Equipment Computer Users' Society) was held in Bedford, MA. At that meeting, I delivered the first paper on the subject, pretentiously titled "SPACEWAR! Real-Time Capability of the PDP-1."

Over the summer of 1962, the original Spacewar hackers began to drift away. Alan Kotok and I went to work for Digital. Steve Russell followed John McCarthy to Stanford University. Peter Samson and Bob Saunders stayed in Cambridge for a while, but eventually they too went west. Dan Edwards remained with the AI group for a few years, then moved to Project MAC. Jack Dennis and the PDP-1 also wound up at Project MAC, which evolved into MIT's Laboratory for Computer Science. Others took up the maintenance and development of **Spacewar!** Program tapes were already showing up all over the country, not only on PDP-1s but on just about any research computer that had a programmable CRT.

A Mystery, Just For Good Measure

Slug tells me that there is a Lost Version of **Spacewar!** There would be, of course. He says the game is pretty much like the original, but the scoring is much more impressive. After each game of a match, cumulative scores are displayed as rows of ships, like a World War II fighter pilot's tally. Slug says he saw this version for a short time on the PDP-1, but never found out who produced it or what became of it.

Twenty Years Later

The original Spacewar PDP-1 was retired in 1975 and put in storage at DEC's Northboro warehouse, where it serves as a parts source for the similar machine now on display at Digital's Computer Museum in Marlboro, MA. At this writing, DEC engineer Stan Schultz and I are trying to put the original **Spacewar!** back into operating condition. So far, all attempts at finding the original control boxes have been futile; we will probably build replicas (the plastic Atari joysticks we have no got no class).

Dan Edwards still works for the U.S. Government, developing computer security systems. Alan Kotok is still a consulting engineer with DEC. Peter Samson is now director of marketing for Systems Concepts, Inc., in San Francisco. Bob Saunders had gone to Silicon Valley, where [he] is an engineer-programmer for Hewlett-Packard.

Jack Dennis is a Professor of Science in the Electrical Engineering Department at MIT.

John McKenzie, the chief engineer, is retired, but over the past year or so has been helping to restore the TX-0 and PDP-1 to life at the Computer Museum.

And what of the Hingham Institute? Wayne Wiitanen has recently become a Senior Research Scientist at the General Motors Research Laboratory, where he is happily designing eyes for robots. Slug, after various adventures, is now a programmer-analyst for Interactive Data Corp. in Waltham, MA. I am reduced to writing for a living, but tend to act somewhat less superior therefore.

Spacewar! itself has bred a race of noisy, garishly-colored monsters that lurk in dark caverns and infest pizza parlors, eating quarters and offering degenerate pleasures. I think I know a few former hackers who aren't the slightest bit surprised.

Acknowledgements

I was able to reach all of the original **Spacewar!** perpetrators, hackers and Hingham Institute Fellows alike. Not to mention Professors Dennis and Minsky, and John McKenzie. In addition, I am grateful to Marcia Baker, Professor F.J. Corbato, and Professor R.M. Fano, all of MIT, for help with dates and places, and other facts. The help was theirs; any mistakes are mine.

[Back to SpaceWar](#)

[Back Home](#)