Space Travel: Exploring the solar system and the PDP-7

As recounted in 'The Evolution of the Unix Time-sharing System,' [links are on my home page] the game of Space Travel was written by Ken Thompson, first for GECOS in interactive batch mode. I've written a couple of accounts of it on Usenet or in mail, but haven't fixed it up for the WWW. Here's an edited version of one version I saved from 1998. (Since this rendition isn't particularly historical, I just fixed poor wording instead of marking changes).

Ken first did Space Travel on the GE 635, before it had a time-sharing facility, but it did have an "interactive batch" mechanism. A job would be submitted, you could type to it, and there were several locally-built displays attached as peripherals. It was expensive to run; a game would cost about $50. Of course it was internal "funny money."

More or less at the same time, Ken discovered the PDP-7, which had been fitted out with a nice vector display. The display was a joint Bell Labs and Digital Equipment design, built by a nearby department as an output facility for the (then) main IBM 7094 computer. This PDP-7 Graphics-II system was much neater than the 635's display, even if older. The PDP-7 was a real computer, not a peripheral, albeit small even by 1968 standards. By then, it was not used much and it was free even by funny-money standards. So Ken moved ST to it, and ran the game standalone. The program was written from scratch, all in assembly language, and even including a complete floating-point arithmetic simulator.

Also, about this time, Ken again got the urge to write his own operating system. He had started on such a project before, but on a much bigger machine -- the GE 645 Multics machine. It didn't take long to realize that he couldn't keep the machine.

Because Ken was now familiar with the '7 and knew he could use it as much as he wanted, the first version of Unix was written on this PDP-7. So ST came before Unix, but doing ST led him to a place in which he could write the first version of Unix.

The game itself: it was a simulation of the solar system, to scale, with the true (scaled) orbit sizes and masses of the planets and most of the moons. However, it wasn't a general numerical simulation; instead the bodies ran on circular tracks at scaled speed in a plane.

All the bodies had gravity, but it acted only on the rocket ship, and only the locally strongest-attracting body acted on the ship. In the simulation, the internal origin for the coordinate system of the computation was at the center of this strongest attractor. This was important for keeping some vague numerical stability in the calculations (with at best single-precision, software floating-point), but did lead to some anomalies.

For example, it turned out that the gravity of Mars (with the usual inverse-square formula) is dominates that of Phobos until you get below the surface of Phobos. So landing on Phobos was strange-- you sort of floated through its surface, then, when it became strongest, you snapped back to the surface.

For controls, there were several buttons: jet forward, jet back, turn left, turn right; go up scale on the display, go down scale. The acceleration of the ship was fixed, but it scaled with the display scale--it was thus fixed in terms of pixels/sec/sec. Normally the ship was in the center of the screen and always pointed up; the display showed the plan view of the solar system, so that the "rotate" controls rotated the solar system around you on the display.

No relativity; scale up enough and you travel to Pluto in a few seconds. But don't scale up too much, or you might not find the Solar System again without restarting.

The object of the game was simply to fly around, get into orbits, land. "Land" meant to cross the surface with a small enough speed.

Playing with orbits was fun; how you accelerate to change the size of the orbit and circularize it isn't very intuitive. One of the variations was to change the display coordinate system so that it rotated with the nearest body. I.e., if you were orbiting Mercury, the ship and Mercury would always lie on a horizontal line. It was also amusing to crank up the gravity (really, change the timescale) so that landing was just barely possible.

Among the features of the program was a built-in debugger that would continuously display the contents of a selected location. This was especially helpful in getting the software floating-point arithmetic to work.

The program also displayed the name of the body that was the current dominating attractor. As it happens, Ravi Sethi (of Aho, Sethi, Ullman and other fame, for example as my former boss and currently as head of Avaya research) was a summer student: for fun the name of the Sun was Surya, and Earth's moon Chandra, after their Hindi/Sanskrit names.

Later we fixed Space Travel so it would run under (PDP-7) Unix instead of standalone, and did also a very faithful copy of the Spacewar game originally done (I think) at MIT on the PDP-1, but that was a different game.

The first PDP-11 didn't have a good display, so the only games were text-based, and were not very interesting, e.g blackjack. They were described in the early Unix manuals. Of course, Ken later did a series of chess programs and hardware that three times won the world computer chess championship, and was the first program to attain USCF master level.