

1 Design Estruturado

Yourdon e Constantine (1979)

"introduzindo uma atividade específica de design formal para descrever completamente, e com antecedência, todas as partes de um sistema e suas inter-relações, não criamos uma nova atividade no ciclo de desenvolvimento de programas."

"o design estruturado é o processo de decidir quais componentes interconectados de qual maneira resolverão algum problema bem especificado."

2 Genesis II, Creation and recreation with computers

Peterson (1983)

Página 187

"Eu estava envolvido em um jogo de interpretação de papéis não-computadorizado chamado Dungeons and Dragons na época, e também vinha explorando ativamente cavernas — especialmente a Mammoth Cave, no Kentucky. De repente, eu tive uma ideia que combinasse o meu interesse por exploração de cavernas com algo que também fosse um jogo para as crianças, e talvez tivesse alguns elementos de Dungeons and Dragons, que eu vinha jogando. Minha ideia era que fosse um jogo de computador que não intimidasse pessoas que não usavam computadores, e esse foi um dos motivos pelos quais eu fiz com que o jogador dirigisse o jogo por meio de comandos em linguagem natural, em vez de comandos mais padronizados. Meus filhos acharam que foi muito divertido."

Página 193

"...o mundo de Zork exige um uso complexo da linguagem. Uma das características distintivas sobre a série de jogos Adventure era que os comandos estavam na forma de frases em inglês, ainda que fossem apenas frases de duas palavras. Mas em Zork, não basta digitar algo como DROP BOX; o jogador precisa ser mais específico. Por exemplo: PUT ONLY THE OPEN BOX UNDER THE TABLE ('COLOQUE APENAS A CAIXA ABERTA DEBAIXO DA MESA'). Para que uma sentença como essa tenha efeito significativo..."

3 Somewhere Nearby is Colossal Cave: Examining Will Crowther’s Original “Adventure” in Code and in Kentucky

Patricia Wilcox (antiga Patricia Crowther) não soube sobre o jogo de Will até depois de encontrar uma cópia da colaboração entre Crowther e Woods. Ela lembra de uma reunião do CRF em “1976 ou 77”, na qual muitos membros que supostamente tinham ido explorar cavernas acabaram passando horas jogando o jogo.

Ela descreve a geografia do jogo como “completamente diferente da caverna real. Usava nomes que nós inventamos.” Wilcox provavelmente está se referindo às seções adicionadas por Woods.

Outros espeleólogos que conhecem bem o jogo relatam que a geografia do jogo corresponde de forma bastante próxima à geografia da caverna real.

Em uma publicação de 1991 em um fórum de espeleologia, Mel Park escreveu que uma fã de Adventure e novata em exploração de cavernas, Bev Schwartz, conhecia o jogo tão bem que, em sua primeira visita à caverna real:

“Chegávamos a uma encruzilhada, ela perguntava as direções da bússola e começava a nos dizer o que havia por este ou aquele corredor — e sempre acertava!”

4 Debugging Game History: A Critical Lexicon

Lowood e Guins (2024) O Nome do Jogo

O nome desse gênero de jogos vem de um jogo específico e inicial que é altamente influente e, de fato, prototípico: Adventure, também conhecido como Colossal Cave ou Colossal Cave Adventure (Crowther, 1976), um jogo de texto em Fortran criado por Will Crowther e depois expandido em sua forma canônica por Don Woods. Esse jogo, disponível em sua forma mais antiga em 1976, foi um sucesso inicial na ARPANET (Advanced Research Projects Agency Network), a predecessora da Internet. Sem usar gráficos nem exigir uma tela (podia ser jogado em um terminal de impressão), ele delineou a maioria dos aspectos importantes dos jogos de aventura.

Um nome de gênero que deriva de uma obra específica é algo incomum nos videogames e em outras mídias. O gênero de comédia romântica no cinema, por exemplo, poderia ter sido chamado “Trouble in Paradise” (Problemas no Paraíso), em referência ao filme de 1932 que muitos consideram o marco definidor da categoria — mas, claro, não foi o caso. O mesmo vale para o

chamado “romance de formação” (ou “romance de amadurecimento”), que não leva o nome de “Tom Jones”, romance de Henry Fielding que é seu exemplar mais antigo; em vez disso, é conhecido pelo termo alemão *bildungsroman*. Embora existam “Sim games” (jogos com o prefixo “Sim”), esse nome designa apenas uma lista de jogos publicados pela mesma empresa, Maxis — mais um exemplo de estratégia de marca do que de formação de gênero.

Pouquíssimos gêneros — do tragicômico ao steampunk — tiram seus nomes de uma única obra que exemplifica o gênero. Um dos poucos casos em que isso ocorre é o do gênero literário especializado “Robinsonada”, que indica o tipo de história colonial de sobrevivência e prosperidade exemplificada pelo romance de Daniel Defoe *Robinson Crusoe* (1719). Ainda assim, embora se espere ser compreendido ao pedir um jogo de aventura em uma loja, é provável que alguém precise se explicar para não ser direcionado à seção de livros de Robinsonadas.

Pode-se argumentar que a situação é semelhante ao dos “jogos de labirinto” (maze games), que no final dos anos 1970 e início dos 1980 eram jogos de ação bidimensionais derivados do jogo *Maze* (também lembrado como *Maze War*), criado por Steve Colley na NASA em 1972–1973 e expandido para múltiplos jogadores por Greg Thompson, Dave Lebling e outros no MIT em 1974. Entretanto, a cadeia de influência de *Maze* até, por exemplo, *K.C. Munchkin* (Averett, 1981) para o *Magnavox Odyssey 2* não é totalmente clara. É verdade que *Adventure*, como *Maze*, é um termo genericamente neutro (sem trocadilho), o que explica, em parte, por que ele foi usado para definir uma categoria quando não temos termos equivalentes para “*DOOM*”, “*Tetris*” ou “*Dance Dance Revolution*”. Ainda assim, compreender a relação do gênero de aventura com seu ancestral é importante, pois ajuda a explicar por que jogos de aventura parecem ser, hoje, algo do passado.

Aventuras Gráficas, para além de *Adventure*

Warren Robinett programou o protótipo do jogo de aventura gráfico — também chamado *Adventure* — para o Atari Video Computer System (VCS), mais tarde conhecido como Atari 2600, lançado em 1978. Embora possa surpreender quem conhece o jogo, Robinett concebeu seu projeto como uma adaptação do *Adventure* de Crowther e Woods, agora em versão gráfica e controlada por joystick. O principal jogo de aventura para consoles recebeu, portanto, o mesmo nome do jogo de texto *Adventure*, que era o grande expoente do gênero de aventuras textuais. O repertório de ações disponíveis para o jogador no Atari VCS *Adventure* não é extenso, mas o jogo mantém muitos dos elementos típicos de uma “missão”, encontrados em aventuras mais elaboradas. O jogo apresenta um início (em que o herói, munido apenas de sua astúcia, parte em sua jornada), um meio (envolvendo exploração e combate) e um fim (quando o cálice é levado de volta triunfalmente ao castelo

dourado) — embora essa estrutura narrativa segmentada não seja exclusiva do gênero de aventuras.

Embora *Adventure de Crowther e Woods* carecesse de gráficos, ele já apresentava todas as principais características do gênero de jogos de aventura.

Em um jogo de aventura, o jogador explora um mundo simulado e fictício para compreendê-lo e expressar essa compreensão por meio de ações dentro do jogo. Por “simulado e fictício”, entende-se que o mundo é interativo, operável e responsivo às escolhas e comentários do jogador, e que o jogador é convidado a abordá-lo com a mesma imaginação ficcional usada ao ler um romance.

Quase sempre há personagens em jogos de aventura e, mesmo quando estão ausentes do mundo simulado — como no best-seller *Myst* (Cyan Interactive, 1993) — o ambiente oferece importantes vestígios de acontecimentos passados. Embora alguns jogos de aventura sejam restritos a uma única linha de progressão ou de missões, outros são abertos, permitindo que diferentes jogadores percorram o enredo de maneiras distintas. Em qualquer caso, a narratividade costuma ser alta, com histórias de fundo, quebra-cabeças e ações em jogo que estão intimamente ligadas à ficção do mundo representado.

Existem muitas versões do jogo de texto *Adventure*, assim como muitos outros jogos fortemente inspirados por ele, como *Zork* (1977) — um minicomputador desenvolvido por Timothy A. Anderson, Mark Blanc, Bruce Daniels e David Lebling no MIT, depois publicado pela Infocom como uma trilogia de aventuras. O *Zork* é analisado em detalhes no livro *Twisty Little Passages: An Approach to Interactive Fiction* (Montfort, 2003). *Adventure* também foi o tema da primeira dissertação de mestrado em estudos de jogos (Buckles, 1985). A pesquisa mais notável sobre o jogo é de Dennis Jerz (2007), que visitou a caverna de Kentucky que inspirou *Crowther*, recuperou o código-fonte original antes de *Woods* modificá-lo e demonstrou que os estudos de jogos podem incluir tanto o contexto exploratório e experiencial do designer quanto a análise do jogo no nível do código.

5 Jeremy Norman’s HistoryofInformation.com

url: <https://www.historyofinformation.com/detail.php?id=2020>

Adventure was originally called ADVENT because a filename could only be six characters long in its operating system. The game was renamed Colossal Cave AdventureOffsite Link, as it was based on part of the Mammoth CaveOffsite Link system in Kentucky.

6 A História de Zork

url: <https://samizdat.co/shelf/documents/2004/05.27-historyOfZork/historyOfZork.pdf>
por Tim Anderson e Stu Galley Primeira Parte da Série

No começo, lá nos anos 1960, a DEC (Digital Equipment Corporation) criou o PDP-10, um computador de médio porte. O “10”, como era chamado, tornou-se popular em muitos centros de pesquisa, e muito software foi desenvolvido para ele — alguns desses programas ainda são tecnologicamente mais avançados do que sistemas modernos.

No Laboratório de Inteligência Artificial do MIT, foi desenvolvido um sistema operacional chamado ITS (Incompatible Time-Sharing System) para o PDP-10. O ITS foi projetado para tornar o desenvolvimento de software fácil. Os criadores do sistema presumiram que ele seria usado por um pequeno grupo de pessoas experientes e amigáveis, então não incluíram nenhum recurso de segurança.

Por volta de 1970, foi inventada a ARPAnet, que permitia que pesquisadores em todo o país (e até no mundo) se comunicassem e usassem os computadores uns dos outros. Naqueles tempos idílicos, o acesso era irrestrito: bastava estar conectado à rede ou conhecer o número de telefone certo. Jovens hackers logo descobriram que isso era um playground maravilhoso. Descobriram também que havia alguns computadores no MIT com coisas bem interessantes e sem segurança — qualquer um que conseguisse se conectar podia entrar.

Também por volta de 1970, surgiu uma linguagem chamada MUDDLE (posteriormente renomeada para MDL), desenvolvida como sucessora do LISP. Ela nunca chegou a substituí-lo, mas criou uma comunidade fiel, especialmente no Project MAC (atual Laboratório de Ciência da Computação do MIT), e particularmente no Dynamic Modelling Group (DM) — mais tarde chamado Programming Technology Division.

O Dynamic Modelling Group foi responsável por alguns jogos notáveis. O primeiro foi um jogo gráfico multiplayer chamado Maze, no qual os jogadores vagavam por um labirinto tentando atirar uns nos outros. A tela de cada usuário mostrava a visão do labirinto de seu “alter ego” digital, atualizada em tempo real. Dave Lebling foi um dos principais responsáveis (ou culpados?) pela criação do jogo.

O próximo jogo de destaque foi Trivia (quem disse que laboratórios de pesquisa não estão à frente do seu tempo?), uma competição contínua de perguntas e respostas para mentes verdadeiramente obsessivas. Diferente de Maze, Trivia podia ser jogado por usuários de toda a ARPAnet e alcançou grande popularidade. Marc Blank escreveu a segunda versão, e eu (Tim Anderson) mantive e hackeei o código — na verdade, era um teste legítimo

de um sistema de banco de dados usado em um projeto de pesquisa.

No início de 1977, Adventure varreu a ARPAnet. Will Crowther foi o autor original, mas Don Woods expandiu o jogo e o lançou na rede, surpreendendo a todos. Quando Adventure chegou ao MIT, a reação foi típica: depois que todos gastaram muito tempo resolvendo o jogo (estima-se que Adventure atrasou a indústria de computadores em duas semanas), os verdadeiros fanáticos começaram a pensar em como poderiam fazer algo melhor.

Adventure era escrito em FORTRAN, o que o tornava limitado. Ele aceitava apenas comandos de duas palavras, era difícil de modificar e tinha problemas que poderiam ser melhorados. (Eu estava presente quando Bruce Daniels, do DM, conseguiu descobrir como obter o último ponto em Adventure analisando o jogo com um depurador em linguagem de máquina — não havia outro jeito.)

Por volta de maio de 1977, Adventure já tinha sido solucionado, e vários membros do DM começaram a procurar novas diversões. Marc Blank aproveitava uma pausa na faculdade de medicina; eu havia acabado de concluir meu mestrado; Bruce Daniels estava entediado com seu tema de doutorado; e Dave Lebling estava farto de código Morse. Dave escreveu, em MDL, um analisador de comandos quase tão inteligente quanto o de Adventure; Marc e eu, que costumávamos passar noites programando, usamos isso para criar um protótipo de jogo com quatro salas. Esse jogo se perdeu há muito tempo. Havia uma banda, uma “sala do amendoim” (onde a banda tocava “Hail to the Chief”) e uma “câmara cheia de prazos”. Dave testou o jogo, achou horrível e foi tirar férias de duas semanas.

Marc, Bruce e eu então resolvemos fazer um jogo de verdade. Começamos desenhando mapas, inventando problemas e discutindo muito sobre como as coisas deveriam funcionar. Bruce ainda sonhava em se formar e preferiu se concentrar no design, então Marc e eu passamos o resto das férias de Dave no terminal, implementando a primeira versão de Zork.

O nome “Zork” nunca foi oficial. Era apenas uma palavra sem sentido que usávamos — geralmente como verbo, tipo “zork the fweep” — e pode ter vindo de outra gíria, “zorch”, que significava destruição total. Costumávamos dar o nome “zork” a um programa enquanto ele ainda estava em desenvolvimento.

Quando Dave voltou, já havia um jogo funcional (mais ou menos). Não era tão grande quanto Adventure — provavelmente menos da metade do tamanho da versão final — mas já tinha o ladrão, o ciclope, o troll, o reservatório e a represa, a casa, parte da floresta, a geleira, o labirinto e várias outras áreas. Os enigmas ainda não eram muito interessantes; demorou um tempo até aprendermos a criar bons desafios, e os primeiros analisadores não suportavam soluções complexas.

O que fizemos certo estava na infraestrutura: havia uma teoria bem de-

finida (e facilmente modificável) para as interações entre objetos, verbos e salas. Era fácil adicionar novos analisadores — algo que acontecia com frequência, já que todo mundo queria tentar escrever o seu. (Marc acabou ficando obcecado e escreveu os últimos 40 ou 50 sozinho.) Também era fácil adicionar novas salas, objetos e criaturas — mas ainda não novas ideias abstratas.

Zork, assim como Adventure, sobreviveu porque foi jogado por pessoas fora da comunidade que o criou. Adventure era portátil por ser escrito em FORTRAN, enquanto Zork era em MDL, que só rodava em alguns PDP-10. Mesmo assim, ele encontrou um público: os “net randoms” — usuários curiosos que vagavam pelos sistemas do MIT, já que não havia segurança nenhuma.

O grupo DM tinha se tornado popular por causa de Trivia, e quando Trivia morreu, os usuários ficaram à espera do próximo jogo. Os primeiros jogadores de Zork variavam de John McCarthy (criador do LISP) até crianças de 12 anos da Virgínia.

Ninguém anunciou oficialmente Zork. As pessoas simplesmente entravam no sistema do DM, viam alguém executando um programa chamado “Zork” e ficavam curiosas. Elas espionavam o terminal da pessoa, percebiam que era um jogo estilo Adventure e logo descobriam como iniciá-lo. Por muito tempo, o comando mágico era :MARC;ZORK. Mesmo quem nunca tinha ouvido falar de ITS, DM ou PDP-10 acabava ouvindo que, se conseguisse acessar o “host 70” na ARPAnet, entrar e digitar a palavra mágica, poderia jogar um jogo de aventura.

Em junho de 1977, Zork já era muito mais primitivo que o Zork I, mas tinha o mesmo espírito. A família Flathead já aparecia, com Lord Dimwit Flathead, o Exagerado, governante do Grande Império Subterrâneo, e a moeda oficial era o zorkmid. Bruce foi responsável pelo texto pomposo onde esses elementos foram apresentados.

Muitos detalhes do mundo subterrâneo eram brincadeiras internas (ou francamente bobos), mas nem tudo era absurdo. No início, se o jogador andasse em uma área escura, caía em um poço sem fundo — até que os usuários começaram a apontar que um poço sem fundo num sótão deveria ser visível do térreo. Foi então que Dave criou as “grues”, criaturas que devoravam quem andasse no escuro.

Desde quase o começo, havia no jogo um jornal fictício chamado “US News e Dungeon Report”, que anunciava mudanças no jogo. Todas as mudanças eram creditadas a algum grupo de “desenvolvedores”, mesmo que não fossem os verdadeiros responsáveis. Um número famoso dizia que Bruce passara semanas enchendo todos os poços sem fundo, o que levou hordas de grues a vagar pelo mapa.

A primeira grande expansão do jogo, feita em junho de 1977, foi a seção do rio, criada por Marc. Ela sobrevive praticamente inalterada em Zork I e mostra bem as dificuldades de criar uma simulação coerente. Havia problemas de consistência — partes do rio eram iluminadas, outras escuras —, mas o maior desafio foi o conceito de veículos. Antes, o jogo tinha apenas salas, objetos e o jogador. Marc introduziu objetos que funcionavam como salas móveis (como um barco). Isso exigiu mudanças delicadas nas interações entre verbos, objetos e salas (por exemplo, o que “andar” significava quando o jogador estava em um barco?).

Os jogadores logo começaram a tentar usar o barco em todo lugar. O código original não permitia isso, mas nada impedia o jogador de levar o barco desinflado até o reservatório e tentar navegar. Eventualmente, o barco foi permitido ali, mas esse tipo de problema persistia: qualquer mudança no mundo simulado afetava todo o resto do mundo simulado.

Embora Zork tivesse apenas um mês de idade, já conseguia surpreender seus criadores. Por causa de um detalhe na implementação, o barco funcionava como uma “bolsa de contenção infinita” — os jogadores podiam colocar qualquer coisa dentro dele, mesmo ultrapassando o limite de peso permitido. O barco era, na verdade, dois objetos diferentes: o “barco inflado” (que continha os itens) e o “barco desinflado” (que o jogador carregava). Nós nem sabíamos disso — alguém acabou relatando como um bug. Até onde sei, o bug ainda está lá.

7 The Untold Story of the Women Who Made the Internet

Evans (2018)

The cavers believed that Flint Ridge met Mammoth past a choke of sandstone boulders at survey point Q-87, a remote spur miles from the surface, but moving the boulders with lengths of metal pipe was backbreaking work. One expedition tried an alternate route, through a vertical crevice called “the Tight Spot.” Caving humor has a nihilistic streak: the Tight Spot is a dark slit so small that only one person in the party dared enter. She was a reedy computer programmer, all of 115 pounds, named Patricia Crowther. Pat wedged herself into the Tight Spot and came out the other end onto a mud bank. In the cool carbide light, she spotted the calling card of a previous visitor: the initials “P. H.,” engraved on the wall. Back at the surface, her party kept the discovery secret. Anyone familiar with the area would know the legend of old Pete Hanson, who had explored Mammoth before the Civil

War. Those had to be his initials down there, which could mean only one thing: Flint Ridge and Mammoth were connected, in a single contiguous cave spanning 340 miles. The monumental discovery would come to be known as the Everest of speleology. Pat returned to broach the juncture ten days later. “By the way, Pat— you’re leading this one,” the other cavers told her. Just beyond the Tight Spot, they waded into muddy water up to their chests, until only a foot of air separated the subterranean river and the dripping cave ceiling. Soaked through and caked with mud “like chocolate frosting,” they struggled to keep their headlamps dry. Eyeless crayfish skittered around their waists. When the passage opened, it revealed a wide hall, where they glimpsed the edge of a hand railing: a tourist trail in the heart of Mammoth Cave. The link was complete. Only moments before, they’d been farther afield than any cavers in history; now, weeping and falling over each other in the water, they were only a few steps away from a public restroom. Riding to base camp in the back of a park ranger’s pickup, they looked up at the stars, bright in the summer sky. Lying “in the open truck bed, with the treetops filing past overhead and falling away behind into the darkness,” they contemplated their feat in silence. The long drive reinforced its magnitude: had they really traveled these seven miles underground? Their final passage, through the Tight Spot and beyond what would come to be known as Hanson’s Lost River, joined an unmarked line on Stephen Bishop’s hand-drawn 1839 map. After hamburgers and champagne at dawn, they slept. “It’s an incredible feeling,” Patricia wrote in a journal account of the trip, “being part of the first party to enter Mammoth Cave from Flint Ridge. Something like having a baby. You have to keep reminding yourself that it’s really real, this new creature you’ve brought into the world that wasn’t here yesterday. Everything else seems new, too. After we wake up on Thursday I listen to a Gordon Lightfoot record. The music is so beautiful, it makes me cry.” The new creature Patricia felt she had brought into the world had always been there, slumbering in the darkness of geologic time. What she’d given birth to that day was not the cave but the map—not the thing, but its description. By wedging herself into the Tight Spot and bringing her lamplight to the darkness, she moved an earthly place into the symbolic Cartesian plane. Or at least that’s how she might have seen it, being the party’s mapmaker.

Back home in Massachusetts, Pat and her husband Will ran a “map factory,” tracking the cartographic data each Cave Research Foundation expedition surfaced. Both being programmers, they brought considerable technical sophistication to mapmaking. As Pat described it, the couple typed raw survey data from “muddy little books” into a Teletype terminal in their living room, which was connected to a PDP-1 mainframe computer at Will’s workplace. From this data, they generated “plotting commands on huge rolls

of paper tape,” using a program Will wrote—Pat contributed a subroutine to add numbers and letters to the final map—which they “carried over and plotted using a salvaged Calcomp drum plotter attached to a Honeywell 316 that was destined to become an ARPANET IMP.” The Crowthers’ maps were simplified line plots, but they represent some of the earliest efforts to computerize caves, a leap in technical sophistication made possible by the hardware to which they had access: a PDP-1 mainframe and a Honeywell 316, a sixteen-bit minicomputer, both far beyond consumer-grade. Will Crowther’s employer was Bolt, Beranek and Newman (BBN), a Massachusetts company that specialized in advanced research. In 1969, BBN was contracted by the U.S. government to help build the ARPANET, the military and academic packet-switching network that spawned our present-day Internet. A few years after they used it to plot their cave maps, the Honeywell 316 minicomputer was repurposed and ruggedized to become an Interface Message Processor, or IMP—what we now call a router. These routers formed a sub-network of smaller computers within the ARPANET, shuffling data around and translating between primary nodes, a vital component of the Internet then and now. Will was one of the strongest programmers at BBN, and his tight, frugal code expressed his fastidious manner. A lifelong mountaineer, he taught Patricia to climb the vertical faces of New York’s Shawangunk Mountains, and was known to hang from his office door frame by his fingertips while deep in thought. Will was a caver, too, and the couple spent all their vacations deep underground. “I get cold when he’s not keeping me company,” she wrote in one caving diary. “There’s quite a draft here; the cave’s breathing.” Will didn’t come along on the final connection trip. He’d been at Patricia’s side on earlier surveys, pushed to his limit in the underground wilderness. But the final survey fell in early September, right as their daughters, Sandy and Laura—aged eight and six, respectively—were headed back to school. One of the Crowthers had to stay home, buy the girls their books and school clothes, take them to the dentist, and register them for classes. Will knew how much the expedition meant to Patricia. She had, after all, found the lead, what cavers call “going cave,” and she was dying to see it through. He told her to go ahead. He’d take care of the girls. When Pat came home, deeply moved by the experience, Will was waiting for her. They stayed up late, holding each other and talking about the connection. When Will fell asleep, Pat crept to the Teletype terminal in the living room and entered, as quietly as she could, the bearings of the survey they’d made in Kentucky. She ran a coordinate program, and the data spooled into her hands in the form of a long paper tape. In the morning, Pat and Will brought the tape to his office, and she watched the BBN computer plot the link she’d made, beneath the earth, between two vast and lonely places. “Now I can

sleep,” she wrote. Caving is unforgiving. Until the late 1960s, anyone entering Mammoth would have passed the glass-topped coffin of Floyd Collins, a country caver who died pinned by a boulder. Cavers become enveloped by the earth, their every move constrained by walls and ceilings of rock. They eat very little—candy bars and canned meat—and carry their waste with them back to the surface. They have no sense of time. Emerging, they may be surprised to see the moon. As the Crowthers’ friends Roger Brucker and Richard Watson wrote in *The Longest Cave*, their account of the connection trip, “The route is never in view except as you can imagine it in your mind. Nothing unrolls. There is no progress; there is only a progression of places that change as you go along.” Making the route visible is the central pursuit of serious caving. The Cave Research Foundation had a group doctrine: no exploration without survey. A map is the only way to see a cave in its entirety, and making maps is caving’s equivalent of summiting mountains. It’s also a survival mechanism. To stay safe, cavers map as they go, “working rationally and systematically to locate known passages.” It’s no wonder the hobby attracts computer programmers. Code is a country populated by the fastidious. Like programmers, cavers may work in groups, but they always face their challenges alone.

Not long after the connection trip, Patricia and Will’s marriage deteriorated. They divorced in 1976, after a separation that left Will “pulled apart in various ways.” Caving without Patricia in the company of their mutual friends in the small Cave Research Foundation community “had become awkward.” Alone and surrounded by their maps, including an extensive survey of the Bedquilt section of Mammoth they’d made together in the summer of 1974, he consoled himself with long Dungeons & Dragons campaigns and late nights coding at home. When Sandy and Laura visited their father, they usually found him hard at work on a long and elegantly structured string of FORTRAN code. He told them it was a computer game, and that when he was done, it would be theirs to play. The novelist Richard Powers once wrote that “software is the final victory of description over thing.” The painstaking specificity with which software describes reality approaches, and sometimes even touches, a deeper order. This is perhaps why Will Crowther felt compelled to make one last map. This one wasn’t plotted from his wife’s muddy notebooks but rather from his own memories. Translated into seven hundred lines of FORTRAN, they became *Colossal Cave Adventure*, one of the first computer games, modeled faithfully on the sections of Mammoth Cave he had explored with Patricia and mapped alongside her, on a computer that would form the backbone of the Internet. *Colossal Cave Adventure*—now more commonly known as *Adventure*—doesn’t look like a game in the modern sense. There are no images or animations, no joysticks or controllers.

Instead, blocks of text describe sections of a cave in the second person, like so: You are in a splendid chamber thirty feet high. The walls are frozen rivers of orange stone. An awkward canyon and a good passage exit from the east and west sides of the chamber. A cheerful little bird is sitting here singing.

In order to interact with this cave, players type terse imperative commands, like GO WEST or GET BIRD, which trigger fresh onslaughts of description. Adventure’s puzzles are an endless shuffle of magical inventory: to pass the snake coiled in the Hall of the Mountain King, you must unleash the bird from its cage, but you can’t GET BIRD if you’re in possession of the black rod, because the bird is afraid of the rod, and in turn, the crystal bridge will not appear without a wave of the rod, and all the while you are in a maze of twisty little passages, all different—or worse, all alike.

This would have been familiar to Will’s colleagues from the ongoing Dungeons Dragons campaign they sometimes played after work. In DD, a game with no winnable objective, a godlike “Dungeon Master” describes scenes in detail, prompting players at actionable decision points. But Will wrote the game for his young daughters. After the divorce, Sandy and Laura came to expect that they’d play computer games whenever they visited their father. According to a researcher who interviewed members of the Cave Research Foundation, “Another caver who was with the Crowthers on an expedition in the summer of 1975 reports that one glance at ‘Adventure’ was enough to identify it immediately as a cathartic exercise, an attempt by Will to memorialize a lost experience.”

Referências

YOURDON, Edward; CONSTANTINE, Larry L. **Structured design: fundamentals of a discipline of computer program and systems design**. [S. l.]: Prentice-Hall, Inc., 1979.

PETERSON, Dale. **Genesis II, Creation and recreation with computers**. [S. l.]: Reston Publishing Company, 1983.

LOWOOD, H.; GUINS, R. **Debugging Game History: A Critical Lexicon**. [S. l.]: MIT Press, 2024. (Game Histories). ISBN 9780262551106. Disponível em: <https://books.google.com.br/books?id=RWbqEAAAQBAJ>.

EVANS, Claire L. **Broad Band: The Untold Story of the Women Who Made the Internet**. New York: Portfolio, 2018. ISBN 9780735211759.