

Videospielgeschichten

Persönliche Geschichten über Videospiele

Wumpusjagd – die erste Game Map der Videospielgeschichte

Nele Abels · Samstag den 18. Februar 2023

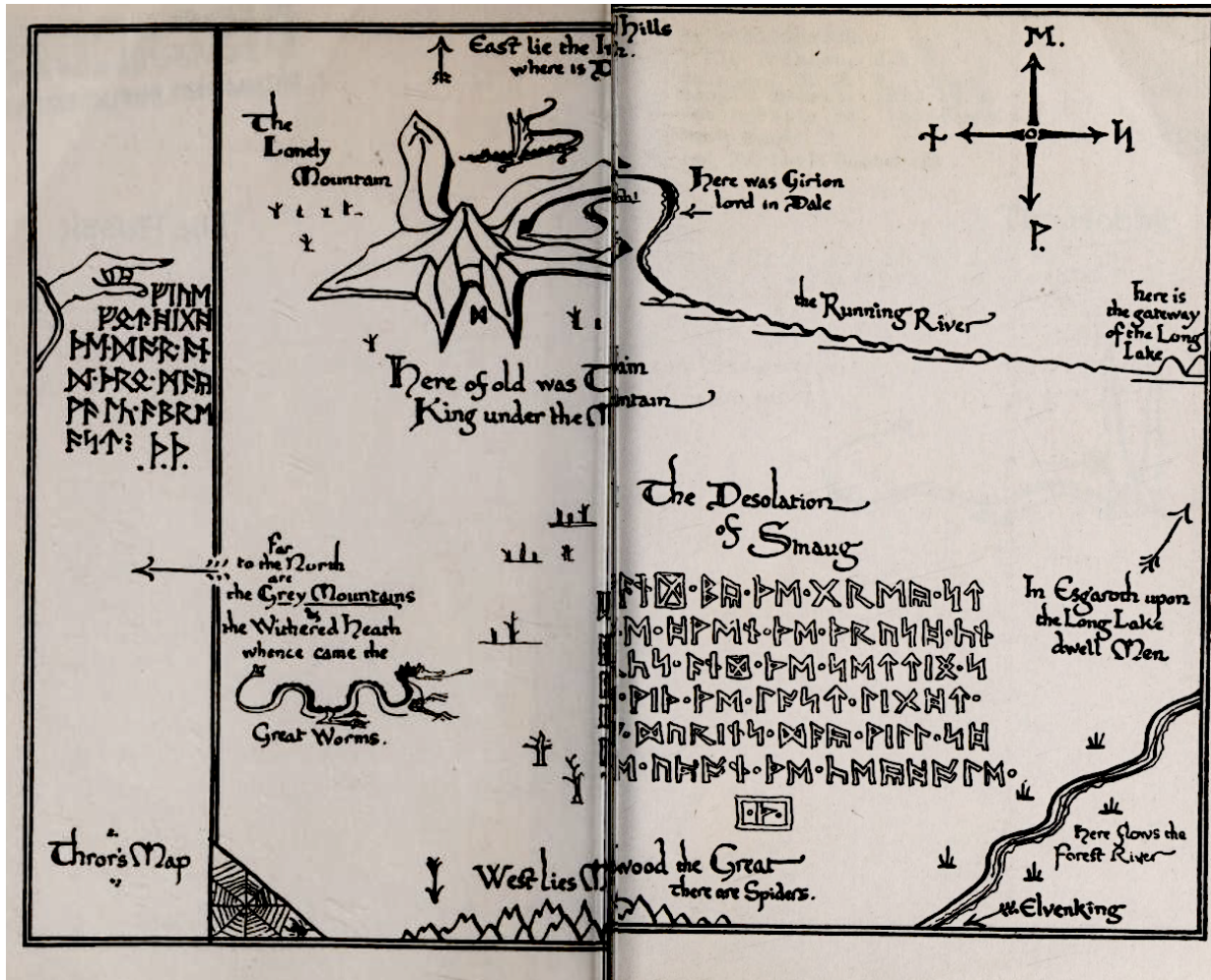
[Bilbo] was getting excited and interested again, so that he forgot to keep his mouth shut. He loved maps, and in his hall there hung a large one of the Country Round with all his favourite walks marked on it in red ink.

J.R.R. Tolkien, The Hobbit (1937), p.19.

Im ersten Abschnitt des Fantasy-Klassikers *The Hobbit. Or There and Back Again* nisten sich die Zwerge um Thorin Eichenschild beim Hobbit Bilbo Baggins ein, den Gandalf der Zauberer als “Meisterdieb” vermitteln will.

Sie betrachten die Landkarte des Zwergen Thror, des Vaters von Thorin, die sie quer durch Mittelerde, durch den Finsterwald und über die Nebelberge hinweg zum Einsamen Berg, jenseits der Stadt am Langen See führen soll. Denn dort liegt Smaug der Drache auf seinem Hort, den er den Zwergen geraubt hat, inmitten des Königreiches, das Thorin als Erbe zurückerobern will.

Die Landkarte, ihre versteckten Hinweise und die Begeisterung Bilbos für die Entdeckung des Unbekannten sind ein Leitmotiv nicht nur des *Hobbits* sondern auch des *Herrn der Ringe*, des großen Erzählungskreises, in den diese Geschichte gehört.



Thror's Karte (*The Hobbit*, p.4)

Und es ist nicht von ungefähr, dass nicht nur den Ausgaben dieser Erzählungen ausklappbare Landkarten aus der Feder Tolkiens beigelegt waren sondern dass auch in den vielen, vielen Romanen oder "Choose your own Adventure"-Büchern der Fantasywelt der 70er und 80er Jahre Landkarten anscheinend unverzichtbar waren.

Terry Brooks' *The Sword of Shannara* (1977) und Joe Devers *Lone Wolf*-Spielbücher (ab 1984) fallen mir dabei spontan ein. Aber literaturhistorisch war J.R.R. Tolkien nicht der erste Autor, für den die Karte das bindende Element seiner erdachten Welt war.

Robert Louis Stevensons Piratenroman *Treasure Island, Die Schatzinsel*, (1882) war eine Karte beigelegt, auf der die Goldmedaillen Captain Flints eingezeichnet sind. Die Utopia des Thomas Morus enthielt schon 1516 die Darstellung der Insel als Holzschnitt. Ab dem Mittelalter waren in den unbekannten Bereichen der Welt fremdartige Fabelwesen wie Kopffüßler oder Einhörner eingezeichnet, die Seeungeheuer blieben auf den Seekarten bis ins 17. Jahrhundert.

Die Aufzeichnung des unentdeckten Digitallandes

Karten und die Erforschung des Unbekannten beflügeln den Menschen – in seinen Erzählungen will er die Grenzen des Bekannten immer weiter hinausschieben oder, wenn das nicht möglich ist, durch seine Fantasie ersetzen.

Was für ein wunderbares Vehikel sind dafür Landkarten; und wie viele Kinder haben spielerisch ihre Traumwelten gezeichnet, inklusive der finsternen Burgruinen in denen gefährliche Drachen oder hinterhältige Zaubermeister lauern?

Und nicht nur Kinder – das Kartenzeichnen gehört zum traditionellen Pen and Paper-Rollenspiel genau so wie zu Computerspielen. Natürlich ist heutzutage die Entdeckung der Welt Teil jeden Open World-Settings, egal, ob in *Assassin's Creed* die Stadt [Florenz](#) erkundet werden will oder ob Arthur Morgan die Berge und Ebenen des [Wilden Westens](#) in *Red Dead Redemption II* durchreitet. Und es ist immer noch die gleiche, alte Faszination: was gibt es noch zu entdecken, welche Wunder zu schauen, welche Geheimnisse zu enthüllen?

Heutzutage wird in Games allerdings nicht mehr selbst kartographiert, die mühsame Arbeit übernimmt der Computer, der im Spielverlauf Stück für Stück zuvor ausgegraute Kartenbereiche sichtbar macht und entdeckte Sehenswürdigkeiten einträgt.

Früher war das anders. Computerspielen war immer kartographieren; penible Zeichnungen der Dungeons in der *Ultima*-Reihe, in *Bard's Tale* und *Wizardry*. Komplizierte Netzwerke von Räumen und Linien mit vielfältigen Anmerkungen und Hinweisen auf Gefahren und Geheimnisse.

Die Gattung der Interactive Fiction, die man damals noch "Textadventures" nannte, kam ohne Kartenzeichnen nicht aus. Man hätte sich sofort in den komplizierten, oft nicht logisch aufgebauten Raumlabyrinthen von *Zork* et al. verloren. Spielerinnen und Spieler zeichneten ihre eigenen Karten, tauschten sie untereinander oder fanden sie in den Zeiten vor dem Internet abgedruckt in Computerspielmagazinen.

Wann fing das Kartenzeichnen in Computerspielen an?

Wie viele andere Ursprünge der Computerspielgeschichte auch lässt sich der Anfang des Kartenzeichnens in Games mit einer Person und einem kleinen Basic-Programm Anfang der 70er Jahre verorten, in diesem Fall *Hunt the Wumpus* von Gregory Yob (1973).

Dieses Spiel entstand an einem ganz bestimmten Moment der Geschichte, der eine Gruppe von Menschen mit einer ganz besonderen Mentalität und Haltung zur damals neuen Computertechnologie zusammenführte.



Gregory Yob (1945-2005)

Die “[People’s Computer Community](#)“, die für sich genommen eigentlich einen Artikel wert wäre, war eine US-amerikanische Grasroots-Bewegung, die sich zum Ziel gesetzt hatte, Computerwissen an die breite Masse zu vermitteln. Sie wollte, dass der Zugang zu Computern nicht mehr nur einer privilegierten Minderheit an Universitäten gewährt war und dass der Umgang mit Computern nicht mehr de facto geheimes Herrschaftswissen einiger weniger war.

Unter den Begründern und frühen Mitgliedern dieser Organisation waren Highschool-Lehrerinnen und -Lehrer, denen es gelang, frühe Multiuser-Systeme zu ergattern, mit denen sie in öffentliche Gemeinderäume gingen und einem interessierten, vornehmlich jungen Publikum Zugang zu den exotischen – und teuren! – Maschinen ermöglichte.

Die Community war einer der Gründe für [die Popularisierung der BASIC-Programmiersprache](#), die in den 80ern untrennbar mit dem Aufstieg der Heimcomputer verbunden war.

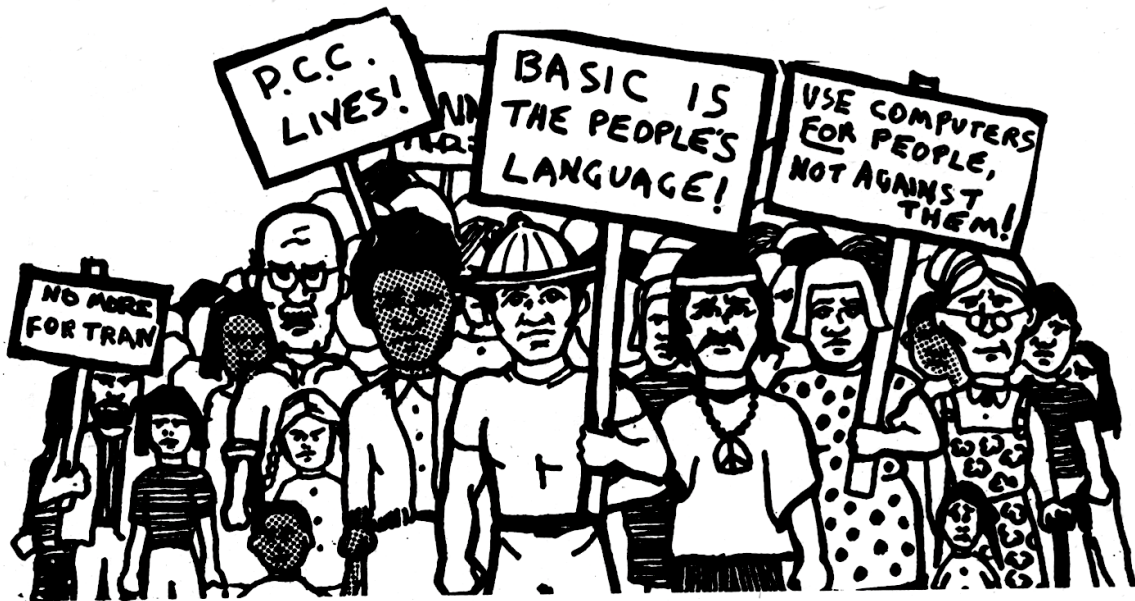


Illustration aus *People's Computer Company*, v.1 n.1 (1972).

Yob war Mitglied einer dieser Gruppen und beschrieb 1974 in einem Artikel in der Zeitschrift "Creative Computing", wie es zur Geburt von *Hunt the Wumpus* kam:

Vor zwei Jahren [1972] ging ich bei der People's Computer Company (PCC) vorbei und sah mir einige ihrer Computerspiele an – so wie *Hurkle*, *Snark* und *Mugwump*. Meine Reaktion war "EEECH!" Jedes dieser Spiele basierte auf einem 10 mal 10 Felder großen kartesischen Gitter und drei Stück davon waren mir zu viel. Ich fing an nachzudenken, so ungefähr im Sinn von 'Man muss doch irgendwie ein Versteck-dich-Spiel ohne dieses #*%*! Gitter machen können!' Warum eigentlich nicht ein topologisches Computerspiel – stell dir eine Menge Punkten vor, die irgendwie miteinander verbunden sind, und der Spieler bewegt sich über die Verbindungen durch diese Menge.

The Best of Creative Computing, vol. 1, p. 247. (1974)

Am gleichen Tag sei ihm nach einem Nachmittag der Meditation der Titel des Spieles gekommen – "Hunt the Wumpus" oder "Die Jagd nach dem Wumpus" – wenngleich er auch nach zwei Jahren noch keine genaue Vorstellung davon hätte, wie ein Wumpus aussehe.

Der Spieler ist auf der Jagd, er durchsucht auf der Suche nach dem unvorstellbaren Ungeheuer ein weitläufiges Höhlensystem.

Doch er muss sich in Acht nehmen! Ein Wumpus ist nicht ungefährlich – er schläft zwar gerne in einer kuscheligen Ecke seiner Höhle, aber wenn man unvermutet hereinpoltert und ihn stört, kann es zu lebensbedrohlichen Folgen kommen. Und auch andere Gefahren lauern im Höhlensystem. Es gibt tiefe Löcher, in die der Jäger im Dunkeln stürzen kann.

Und eine Spezies von gigantischen Fledermäusen, die einen zwar nicht verschlingen, aber sich einen Spaß daraus zu machen, den Spieler zu greifen und unter lautem Fiepsen an einer zufälligen anderen Stelle des Höhlensystems wieder abzuwerfen. Doch zum Glück kann man diese Laute bei der vorsichtigen Erforschung der Gänge aus benachbarten Höhlenräumen hören und den Luftzug der abgrundtiefen Löcher spüren.

Hunt the WUMPUS



What do you do after you hit return?, p.63. (1975)

Weil der Wumpus ein gefährliches Untier ist, gibt es nur eine Art ihn zu erlegen, man muss ihn mit Pfeil und Bogen aus der Ferne niederstrecken! Glücklicherweise sind die Pfeile im Spiel nicht die primitiven angespitzten Stöcke, die wir kennen und die nur geradeaus fliegen können.

Es handelt sich dabei um eine Art programmierte Lenkwaffen, denen man vorher mitteilt, in welchen Kurven und Windungen sie durch die Höhlengänge sausen sollen. Das erhöht die Chance, den Wumpus zu treffen, aber wenn man nicht aufpasst, endet der Pfeil eventuell im eigenen

Allerwertesten, was die Jagdfreude endgültig vermindert...

Eine mathematische Spielwelt

Was meinte Yob nun mit einem “topologischen Computerspiel” und was störte ihn an den “kartesischen Gittern”? Letztere sind nichts anderes eine zweidimensionale Tabellen, in die Daten eingetragen werden können.

Ein klassisches Spiel mit einem kartesischen Gitter, das wir wahrscheinlich alle kennen, ist “Schiffe versenken”, wo jedes einzelne Feld entweder von einem Teil eines Schiffsrumpfes besetzt ist oder Wasser repräsentiert. Ein anderes das Schachspiel, bei dem die Felder von unterschiedlichen Spielfiguren besetzt sind.

In Programmiersprachen wie BASIC, in denen solche Felder durch den Datentyp Array abgebildet werden, ist es sehr einfach, kartesische Gitter zu verwalten – die Variable `a(4,5)` repräsentiert den Inhalt des Feldes an den Koordinaten (4,5), mit Schleifen und Zählervariablen lassen sich die Daten gut verwalten. Das ist sicherlich bequem für die Programmierer, aber nicht so sonderlich interessant für Spieler. Die Landschaft eines Schachbrettes ist nicht wirklich spannend.

Die Topologie ist in der Mathematik die Lehre von der Lage und Anordnung geometrischer Körper im Raum. Ein Knoten ist mit seinen Schleifen und Windungen ein typisches topologisches Objekt.

Und worum es Yob ging, war, dass das Spielfeld dadurch interessanter wurde, dass es nicht mehr eine platte Fläche war, auf der man sich nur in vier Richtungen bewegen konnte, sondern ein Netzwerk, das nicht leicht überschaut werden konnte und das Möglichkeiten zu taktisch überlegten Bewegungen bot. Yobs geometrischer Lieblingskörper war zu dieser Zeit der Dodekaeder, den die Pen-and-Paper-Rollenspieler unter uns als zwölfseitigen Würfel kennen.



Ein zwölfseitiger Würfel, wie er z.B. in D&D verwendet wird. ([Wikimedia](#))

Diesen dreidimensionalen Körper nahm Yob zur Abbildung seines Höhlensystems, in dem jede der 20 Ecken einen Raum darstellt, der jeweils drei Ausgänge zu drei anderen Räumen hat. So entsteht eine komplexere Spielwelt als ein Schachbrett und über das Userinterface des Spiels wird den Spielern das Gefühl vermittelt, dass sie sich innerhalb eines Systems von Räumen befinden, in dem sie sehr vorsichtig ihren Weg ertasten müssen.

Das ist es ein ganz anderes Spielgefühl als der Blick “von oben” auf das Schachbrett oder auf Meer bei “Schiffe versenken”, auf dem man die Flotte des Gegners aufspüren muss. Was hier vorweggenommen wurde ist die Perspektive, die in den späteren Textadventures normal wurde, die Kombination aus einem Ort, an dem man sich befindet, und den Wegmöglichkeiten zu anderen Orten, die er bietet.

Hier ist die Spielwelt noch eine sehr einfache geometrische Figur, die sich in den Textadventures dann in beliebige Wegenetze wandelt, die völlig frei nach der Vorstellung der Programmierer entworfen werden sein können. Die Codierung eines solchen Wegenetzes ist in beiden Fällen die gleiche:

```
0068  REM- SET UP CAVE (DODECAHEDRAL NODE LIST)
0070  DIM S(20,3)
0080  FOR J=1 TO 20
0090    FOR K=1 TO 3
0100      READ S(J,K)
0110    NEXT K
0120  NEXT J
0130  DATA 2,5,8,1,3,10,2,4,12,3,5,14,1,4,6
0140  DATA 5,7,15,6,8,17,1,7,9,8,10,18,2,9,11
0150  DATA 10,12,19,3,11,13,12,14,20,4,13,15,6,14,16
0160  DATA 15,17,20,7,16,18,9,17,19,11,18,20,13,16,19
```

Auszug aus dem Originalprogramm. (*The Best of Creative Computing*, vol. 1, p. 250.)

In diesem Programmausschnitt von *Wumpus* erkennt man, dass jedem der zwanzig Räume im Array S drei Ausgänge zugeordnet sind, die mit Schleifen aus den Data-Zeilen geladen werden.

Raum 1 hat also Ausgänge in die Räume 2, 5, 8. Das ist haargenau die Art und Weise, wie in allen frühen BASIC-Textadventures die Spielkarte generiert wurde, auf der die traditionellen Ausgänge nach Norden, Süden, Osten und Westen in einem zweidimensionalen Array für jeden Raum abgebildet wurden.

Netterweise sind die Wege in “Wumpus” fest verdrahtet und nicht zufallsgeneriert, was Yob damit begründet, dass engagierte Spieler die Zusammenhänge erkennen und zur Grundlage ihrer

Strategie machen könnten. In der Tat funktioniert das auch und macht das Spiel gewinnbar.

Gedacht war eine sorgfältige Strategie...

Yob beschreibt in dem Begleittext zum Listing von “Wumpus”, das in Best of Creative Computing 1 (1976) abgedruckt wurde, welche möglichen Lösungsstrategien er für das Spiel sah. Für ihn stand von Anfang an die systematische Erforschung des Höhlensystems im Vordergrund:

Meine grundsätzliche Idee zu dieser Zeit [als Yob das Spiel programmierte, N.A.] war, dass der Spieler sich dem Wumpus näherte, sich dann wieder zurückzog und dann um den Dodekahedron herum wieder auf ihn los ging. Meines Wissens ist das niemals geschehen... die meisten Spieler verfolgen eher andere Strategien als diesen kaltblütigen Ansatz.

The Best of Creative Computing, vol. 1, p. 247.

Leider demonstriert Yob seine Strategie nicht am praktischen Beispiel, ebenso wenig wie die Vorgehensweise der anderen Spieler. Als ich das Spiel als Kind kennenlernte, hatte ich überhaupt keine Strategie. Ich torkelte mehr oder weniger orientierungslos durch die Höhlen, fiel nichtsahnend in die Löcher oder wurde von Fledermäusen davongetragen. Fand ich jemals einen Wumpus, ohne von ihm aufgefressen zu werden? Ich weiß es nicht mehr; Fool of a Took...

Der lange Weg zum Spiel oder “Der Fluch der Basicprogramme”...

Spielen wir doch einfach mal eine Runde “Wumpus”, so wie Yob es sich vor einem halben Jahrhundert vorgestellt hat, nämlich ohne Rumprobieren und mit einer klaren Strategie.

Für mich hat sich dabei zuerst die Frage nach der geeigneten Plattform gestellt. Dank Emulatoren und Internet ist es ja heute weder ein Problem, die passenden Programme zum Download zu finden, noch passende Softwareemulationen alter Systeme.

Wumpus-Adaptionen gibt es wie Sand am Meer, die meisten sind jedoch mehr oder weniger verändert worden, teilweise sogar zu dem von Yob verachteten kartesischen Spielfeld in der [Version auf dem TI-99/4a](#) oder in [grafischer 3D-Umsetzung](#), was ja nun überhaupt nicht passt.

Mein Ziel war ein möglichst authentisches Spielerlebnis, deshalb tippte ich das Originallisting aus dem Artikel in der *Creative Computing* in GW-Basic auf der DOSBox ab. In anderen Worten, ich verwendete auf einem MacBook des 21. Jahrhunderts einen Emulator eines dreißig Jahre alten Betriebssystems, um in einem vierzig Jahre alten Basicdialekt ein fünfzig Jahre altes Computerspiel zum Leben zu erwecken. Computerarchäologie ist schon ein seltsames Hobby.

Wie jeder weiß, der in den 80ern vor der hakeligen Tastatur seines Heimcomputers gesessen und die schwer leserlich abgedruckten Listings aus Computerzeitschriften “in den Rechner gehackt” hat, ist die Eingabe von solchen Programmen eine mühselige und fehlerträchtige Angelegenheit.

Ein Buchstabenverdrehen in den Variablen? Eine falsche Zahl oder ein fehlendes Komma? Wir brauchen uns nicht einbilden, dass da irgendein Debugger einer IDE mit hilfreichen Farbmarkierungen oder klaren Fehlerangaben hilfreich eingesprungen wäre.

Und ein bequemer Bildschirmeditor mit dem man den Sourcecode auch über den Bildschirminhalt hinaus scrollen kann? Wir hatten ja nichts, unsere Computer liefen mit Kohle und mussten im Winter vorgeheizt werden! Was es dagegen reichlich gab, war Spaghetti-Code mit unzähligen GOTOs und schauerlichen Verbrechen, wie dem Sprung aus einer Schleife, ohne sie vorher ordnungsgemäß abzuschließen, was Yob gleich mehrfach beging. Anscheinend wusste er nicht, was ein Stack ist...

```

0110 NEXT K
0120 NEXT J
0130 DATA 2,5,8,1,3,10,2,4,12,3,5,14,1,4,6
0140 DATA 5,7,15,6,8,17,1,7,9,8,10,18,2,9,11
0150 DATA 10,12,19,3,11,13,12,14,20,4,13,15,6,14,16
0160 DATA 15,17,20,7,16,18,9,17,19,11,18,20,13,16,19
0170 DEF FNA(X)=INT(20*RND(0))+1
0180 DEF FNB(X)=INT(3*RND(0))+1
0190 DEF FNC(X)=INT(4*RND(0))+1
0200 REM-LOCATE L ARRAY ITEMS
0210 REM-1-YOU,2-WUMPUS,3&4-PITS,5&6-EATS
0220 DIM L(6)
0230 DIM M(6)
0240 FOR J=1 TO 6
0250 L(J)=FNA(0)
0260 M(J)=L(J)
0270 NEXT J
0280 REM-CHECK FOR CROSSOVERS (IE L(1)=L(2),ETC)
0290 FOR J=1 TO 6
0300 FOR K=J TO 6
0310 IF J=K THEN 330
0320 IF L(J)=L(K) THEN 240
0330 NEXT K
0340 NEXT J
0350 REM-SET# ARROWS
0360 A=5
0365 L=L(1)
0370 REM-RUN THE GAME
0375 PRINT "HUNT THE WUMPUS"
0380 REM-HAZARD WARNINGS & LOCATION
0390 GOSUB 2000
0400 REM-MOVE OR SHOOT
0410 GOSUB 2500
0420 GOTO 0 OF 440,480
0430 REM-SHOOT
0440 GOSUB 3000
0450 IF F=0 THEN 390
0460 GOTO 500
0470 REM-MOVE
0480 GOSUB 4000
0490 IF F=0 THEN 390
0500 IF F>0 THEN 550
0510 REM-LOSE
0520 PRINT "HA HA HA - YOU LOSE!"
0530 GOTO 560
0540 REM-WIN
0550 PRINT "HEE HEE HEE - THE WUMPUS'LL GETCHA NEXT TIME!"
0560 FOR J=1 TO 6
0570 L(J)=M(J)
0580 NEXT J
0590 PRINT "SAME SET-UP (Y-N)";
0600 INPUT I$
0610 IF I$="Y" THEN 240
0620 GOTO 360
1000 REM-INSTRUCTIONS
1010 PRINT "WELCOME TO 'HUNT THE WUMPUS'"
1020 PRINT "THE WUMPUS LIVES IN A CAVE OF 20 ROOMS. EACH ROOM"
1030 PRINT "HAS 3 TUNNELS LEADING TO OTHER ROOMS. <LOOK AT A">"
1040 PRINT "DODECAHEDRON TO SEE HOW THIS WORKS-IF YOU DON'T KNOW"
1050 PRINT "WHAT A DODECAHEDRON IS, ASK SOMEONE)"
1060 PRINT
1070 PRINT "HAZARDS:"
1080 PRINT "BOTTOMLESS PITS - TWO ROOMS HAVE BOTTOMLESS PITS IN THEM"
2500 REM-CHOOSE OPTION
2510 PRINT "SHOOT OR MOVE (S-M)";
2520 INPUT I$
2530 IF I$="S" THEN 2560
2540 O=1
2550 RETURN
2560 IF I$="M" THEN 2510
2570 O=2
2580 RETURN
3000 REM-ARROW ROUTINE
3010 F=0
3020 REM-PATH OF ARROW
3030 DIM P(5)
3040 PRINT "NO. OF ROOMS(1-5)";
3050 INPUT J9
3060 IF J9<1 OR J9>5 THEN 3040
3070 FOR K=1 TO J9
3080 PRINT "ROOM #";
3090 INPUT P(K)
3095 IF K <= 2 THEN 3115
3100 IF P(K) <> P(K-2) THEN 3115
3105 PRINT "ARROWS AREN'T THAT CROOKED - TRY ANOTHER ROOM"
3110 GOTO 3080
3115 NEXT K
3120 REM-SHOOT ARROW
3130 L=L(1)
3140 FOR K=1 TO J9
3150 FOR KI=1 TO 3
3160 IF S(L,KI)=P(K) THEN 3295
3170 NEXT KI
3180 REM-NO TUNNEL FOR ARROW
3190 L=S(L,FNB(1))
3200 GOTO 3300
3210 NEXT K
3220 PRINT "MISSED"
3225 L=L(1)
3230 REM-MOVE WUMPUS
3240 GOSUB 3370
3250 REM-AMMO CHECK
3255 A=A-1
3260 IF A>0 THEN 3280
3270 F=-1
3280 RETURN
3290 REM-SEE IF ARROW IS AT L(1) OR L(2)
3295 L=P(K)
3300 IF L#L(2) THEN 3340
3310 PRINT "AHA! YOU GOT THE WUMPUS!"
3320 F=1
3330 RETURN
3340 IF L#L(1) THEN 3210
3350 PRINT "OUCH! ARROW GOT YOU!"
3360 GOTO 3270
3370 REM-MOVE WUMPUS ROUTINE
3380 K=FNC(0)
3390 IF K=4 THEN 3410
3400 L(2)=S(L(2),K)
3410 IF L(2)#L THEN 3440
3420 PRINT "TSK TSK TSK- WUMPUS GOT YOU!"
3430 F=-1
3440 RETURN
4000 REM- MOVE ROUTINE
4010 F=0
4020 PRINT "WHERE TO?";

```

Programmsprünge allein schon im Code für den Pfeilschuss...

Aber sei es drum – in der guten alten Zeit war das mühselige Abtippen ja ein wesentlicher Teil des Spielspaßes und bei der Fehlersuche das Programm zu analysieren und zu verstehen brachte fast mehr Vergnügen, als es dann tatsächlich zu spielen.

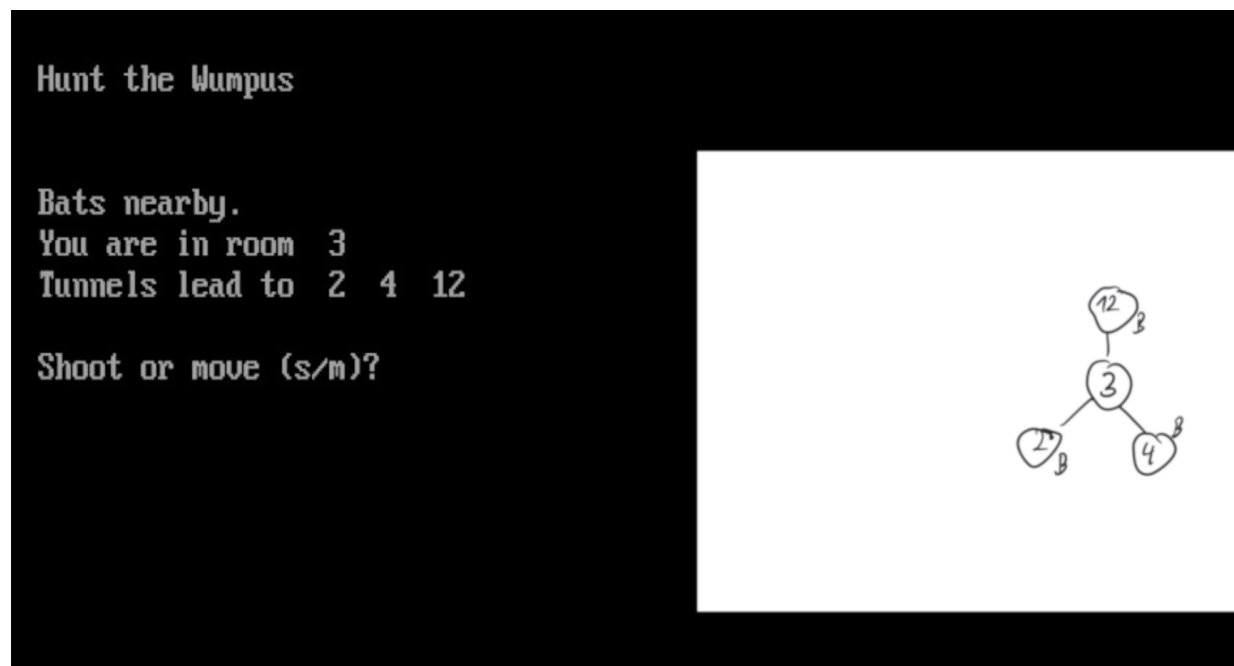
Nach einer Weile habe ich *Hunt the Wumpus* zum Laufen gebracht und konnte es endlich selber durchspielen, wie es gedacht war.

Papier und Bleistift in elektronischer Variante gezückt

Nun geht es aber wirklich los mit dem Spielerlebnis von 1973. Nun ja, nicht ganz, denn zu Yobs Zeit hätte man keinen Bildschirm gehabt sondern die Interaktion mit dem Computer wäre über einen Fernschreiber der Marke **Teletype Model 33** (inklusive Lochstreifen!) gelaufen und man hätte sich seine Notizen mit Papier und Stift gemacht.

Ich mache es mir da etwas bequemer und lasse das Programm auf dem Computerbildschirm laufen und benutze für die Karte iPad und Apple-Pen. Look und Feel des Basicprogramms sind allerdings so wie von Yob vorgesehen. (Einen – etwas verschnupften – Videomitschnitt des fünf Minuten

dauernden Spiels [habe ich auf Youtube gestellt.](#))



Das geht ja gut los! Wo sind die Fledermäuse?

Von atmosphärischer Beschreibung oder gar multimedialer Ambience-Ausstattung, wie wir es als Spieler in der heutigen verwöhnten Zeit erwarten, ist da natürlich nichts zu sehen. Das muss man sich alles selber vorstellen.

Aber wir haben Informationen. Ich sehe die Ausgänge des Raums 3 und ich erfahre, dass in der Nähe Fledermäuse lauern, die mich ja ergreifen und zufällig irgendwo im Labyrinth wieder absetzen können.

Vielleicht sogar im Nest des Wumpus oder über einer bodenlosen Fallgrube. Also zeichne ich den Anfang des Wegenetzes und schreibe an alle Räume ein kleines “B”, in dem Fledermäuse lauern können.

In die erste Falle getappt...

Ohne Informationen kann ich nur blindlings in einen der Gänge stolpern und mit einer Chance von 1:3 auf Fledermäuse stoßen. Immerhin habe ich mit einiger Wahrscheinlichkeit nicht das Risiko schon im ersten Spielzug zu sterben, denn nur ein zufälliger Abwurf über einem Loch (Chance 2:18) oder über einem Wumpus (Chance 1:18, wenn er bei 1:4 im Raum bleibt) bedeuten mein Ende. Auf gut Glück entscheide ich mich für den Ausgang nach Raum 2...

Hunt the Wumpus

Bats nearby.

You are in room 3

Tunnels lead to 2 4 12

Shoot or move (s/m)? m

Where to? 2

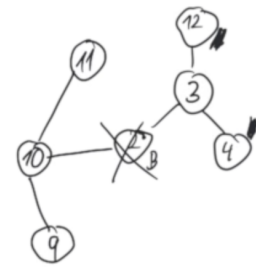
Zap -- Super Bat Snatch! Elsewhereville

Bats nearby.

You are in room 10

Tunnels lead to 2 9 11

Shoot or move (s/m)?



Verdammte Blutsauger!

...und erwische prompt den Raum mit den Fledermäusen, die mich in den Raum 10 tragen. Aber weiter passiert nichts, Glück gehabt. Und ich habe jetzt Informationen – ich weiß mit Sicherheit, dass die Fledermäuse in Raum 2 sind, und kann ihn Zukunft vermeiden, und weiß darüber hinaus, dass die Räume 12 und 4 sicher sind. Ich markiere meine Karte entsprechend.

Ich muss allerdings weiter aufpassen, denn die Warnung vor Fledermäusen steht da weiterhin. Das könnte sich natürlich auf Raum 2 beziehen, da in dem Labyrinth aber zwei Räume mit Fledermäusen angelegt sind, besteht für die Räume 11 und 9 jeweils eine Chance von 1:13, dass ich erneut davon getragen werde. (1:13 deshalb, weil ich den Inhalt von sieben Räumen schon kenne.)

Das ist kein großes Risiko, aber für ein strategisch kluges Spiel sollte man die Wahrscheinlichkeit berücksichtigen und die Möglichkeit in der Karte vermerken. Auf gut Glück geht es weiter nach Raum 11.

```

Bats nearby.
You are in room 3
Tunnels lead to 2 4 12

Shoot or move (s/m)? m
Where to? 2
Zap -- Super Bat Snatch! Elsewhereville

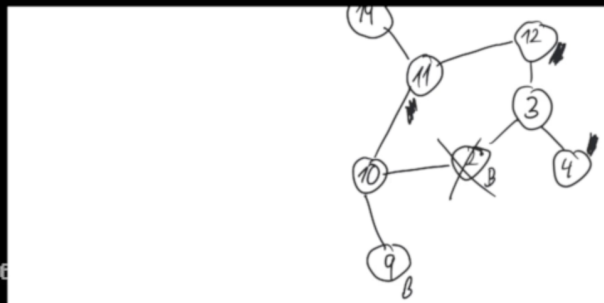
Bats nearby.
You are in room 10
Tunnels lead to 2 9 11

Shoot or move (s/m)? m
Where to? 11

You are in room 11
Tunnels lead to 10 12 19

Shoot or move (s/m)? _

```



Immerhin ein etwas besserer Überblick

Nichts passiert. Auf der Karte kann ich jetzt den Warnhinweis bei Raum 11 entfernen und die neuen Ausgänge eintragen. (Man kann übrigens mit etwas Phantasie schon die Form eines Dodekaeders erkennen, den man von unten oder von oben betrachtet.) Der Warnhinweis bei Raum 9 bleibt stehen – dort könnten ja immer noch Fledermäuse lauern.

Die erste Spur des Wumpus!

Ich beschließe, die Erforschung der Höhle vom Ausgangspunkt aus fortzusetzen und nehme den sicheren Weg durch die Räume 12 und 3 bis nach Raum 4. Und dort habe ich die Witterung meiner Jagdbeute – ein Wumpus!

```
You are in room 11
Tunnels lead to 10 12 19
```

```
Shoot or move (s/m)? m
Where to? 12
```

```
You are in room 12
Tunnels lead to 3 11 13
```

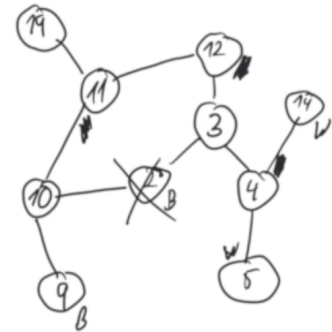
```
Shoot or move (s/m)? m
Where to? 3
```

```
Bats nearby.
You are in room 3
Tunnels lead to 2 4 12
```

```
Shoot or move (s/m)? m
Where to? 4
```

```
I smell a Wumpus!
You are in room 4
Tunnels lead to 3 5 14
```

```
Shoot or move (s/m)? _
```



Diesen stechenden Geruch kenne ich doch...

Noch riskiere ich keinen Schuss, denn der Wumpus könnte in Raum 14 oder in Raum 5 lauern. Wenn ich ihn mit der Wahrscheinlichkeit von 1:2 verfehle, könnte ich ihn eventuell aufscheuchen und müsste ihn dann mühselig neu aufspüren. Oder er kommt in meine Höhle und frisst mich eventuell auf!

Das Risiko ist zu groß, ich beschließe, weiter durch das Labyrinth zu schleichen und mehr Hinweise auf seinen genauen Aufenthaltsort zu sammeln.

Das Netz zieht sich zu...

Meinen Weg nehme ich wieder über die sichere Route 3 und 12 nach Raum 13, der oben in der Karte fehlt, weil ich ihn in der nervenaufreibenden Spannung des Spiels einzutragen vergessen habe. Und jetzt kommt der Durchbruch:

```
You are in room 4
Tunnels lead to 3 5 14
```

```
Shoot or move (s/m)? m
Where to? 3
```

```
Bats nearby.
You are in room 3
Tunnels lead to 2 4 12
```

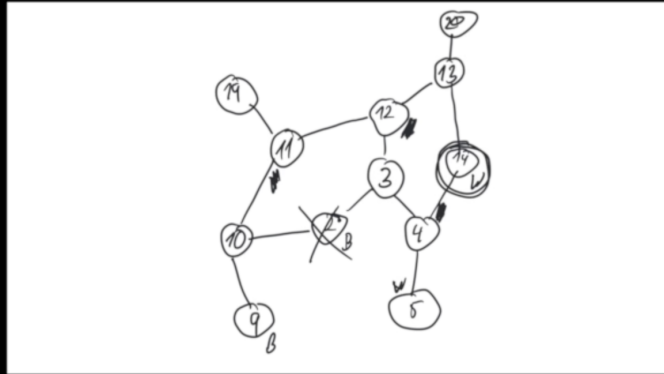
```
Shoot or move (s/m)? m
Where to? 12
```

```
You are in room 12
Tunnels lead to 3 11 13
```

```
Shoot or move (s/m)? m
Where to? 13
```

```
I smell a Wumpus!
You are in room 13
Tunnels lead to 12 14 20
```

```
Shoot or move (s/m)? _
```



Gefunden!

Erneut nehme ich die Witterung des Wumpus auf. Und da es nur einen Wumpus im Höhlensystem gibt, habe ich jetzt Gewissheit. Das Untier kann weder in Raum 5 noch in Raum 20 sein sondern muss sich in Raum 14 befinden. Es gibt logisch keine andere Möglichkeit.

Ich kann also meinen Pfeil ohne Risiko und mit 100% Trefferwahrscheinlichkeit abschießen und erlege den Wumpus erwartungsgemäß:

```
Shoot or move (s/m)? s
No. of rooms (1-5)? 1
Room #? 14
Aha! You got the Wumpus.
Heeheehee - the Wumpus will get you next time...

Same setup (y/n);
? _
```

Viel Lob gibt es nicht...

Wie in Spielen dieser Zeit üblich wird das Resultat in kargen Worten gegeben. Weder wird ein “Wumpus tot” noch ein “Halali” geblasen. Fast könnte man allerdings eine gewisse Häme zwischen den Zeilen lesen, etwa so, als ob das Programm selber überrascht wäre, dass der Spieler gewonnen hat. Aber das soll uns nicht stören, denn wir wissen, dass uns der Sieg aufgrund unserer überlegenen Strategie gewiss war...

Wenn gewünscht lässt sich das Spiel mit der gleichen Verteilung von Fallgruben, Fledermäusen und Wumpus nachspielen. Man kann so eventuelle Spielfehler nachvollziehen und überprüfen, ob die eigenen Vermutungen bei der Erkundung richtig lag. Oder der Wumpus, die Fallgruben und die Fledermäuse werden erneut im Höhlensystem verteilt.

Wie hält sich ein Computerspiel nach fünfzig Jahren?

Wie ist “Hunt the Wumpus” gealtert? Hat es heute überhaupt noch einen spielerischen Wert? Ursprünglich bin ich aus eher narratologischem Interesse an das Programm herangegangen – wegen seines Wertes als wichtiger Schritt in der Entwicklung interaktiven Erzählens, als der es allgemein angesehen wird. Wie das Spiel als erstes eine Spielwelt als erforschbare Karte definiert, habe ich beschrieben.

Andere Kommentatoren lesen “Wumpus” als den ersten Vertreter des [Horror-Survival-Genres](#), eine Einschätzung, die ich nur bedingt teile, denn der Fokus der Interaktion im Spiel liegt nicht beim Grusel. Unstrittig ist aber, dass *Wumpus* eine entscheidende Rolle als Präform der textbasierten Interactive Fiction spielt.

Das ursprüngliche Textadventure *Advent*, bzw. *Colossal Cave Adventure* von Will Crowther (1976) vereint die Idee der Erforschung eines vernetzten Systems von Räumen aus *Wumpus* mit einer Dialogsimulation in natürlicher Sprache, wie sie von *Eliza* (1966), einer “virtuellen Psychiaterin”, und *SHRDLU* (1970), das die Manipulation von Objekten mit Sprachbefehlen simulierte, vorgemacht worden war.

Allein schon deshalb verdient Yobs Spiel eine Würdigung noch im Jahr 2023.

Außerdem habe ich beim Experimentieren mit *Wumpus* festgestellt, dass das Spiel tatsächlich immer noch Spaß macht. Obwohl es nur ca. 300 Programmzeilen in wenig dichtem Basic-Code umfasst, gelingt es Yobs Programm, spielerische Tiefe zu erzeugen.

Eine gute Strategie und sorgfältige Überlegungen machen das Spiel gewinnbar und Zufallsfaktoren

sind zwar spielrelevant aber nicht alles bestimmend, so wie in vielen anderen Games dieser Zeit.

Auch für heutige Spielerinnen und Spieler lohnt sich noch ein Blick auf das Programm und es kann für einige Runden “Casual Gamings” unterhaltend sein – sofern man zufällig Papier und Bleistift parat hat. Wer *Hunt the Wumpus* ausprobieren möchte, kann es am einfachsten [über eine Online-Variante tun](#), die dem Originalspiel wohl am nächsten kommt.

Wie ging es weiter?

Gregory Yob entwickelte noch eine Fortsetzung seines Spiels, die einige andere Formen von Höhlensystemen anbot, sich aber konzeptuell wenig von seiner Originalidee unterschied.

Nachdem er den Meilenstein von *Wumpus* gesetzt hatte, trat er aber nicht mehr als Spielenentwickler in Erscheinung sondern lebte ein Hippie-Leben auf der Suche nach philosophischer Erleuchtung, nannte sich augenzwinkernd einen “Neo-Neuro-Cyber-Schamanen” und war unter dem Namen “Hara Ra” bekannt.

2005 starb er, [nachdem er zuvor verfügt hatte, seinen Körper kryogenisch einzufrieren](#), in der Hoffnung irgendwann wiederbelebt zu werden, wenn die Medizin für ihn hilfreiche Heilungsverfahren entwickelt hat.

“Hunt the Wumpus” fand noch eine kommerzielle Bedeutung als Cartridge für den hierzulande relativ unbekannten 8-Bit Computer TI99/4A.

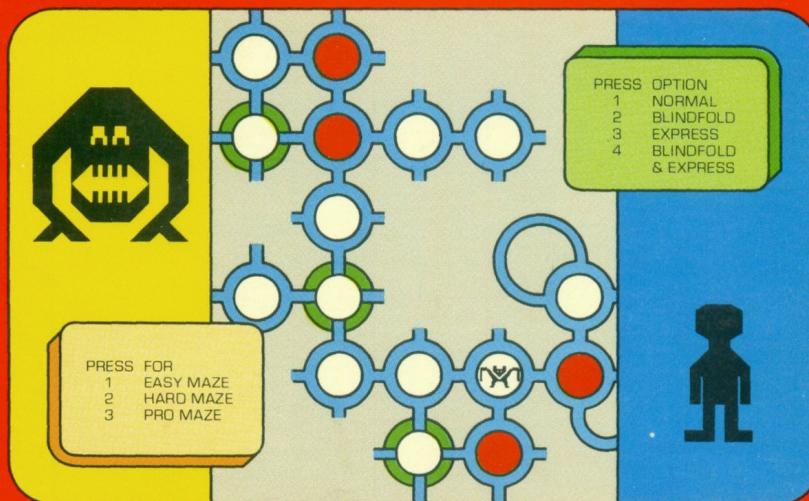
Texas Instruments Home Computer



Hunt the Wumpus

SOLID STATE
SOFTWARE™**COMMAND MODULE**

An exciting simulated hunt in a hidden maze of caverns and twisting tunnels! Seek out the lair of the Wumpus, while avoiding perils along the way!



Cover der Cartridge-Version für den Ti99/4a (1981)

Diese Version gab das Konzept eines Dodekaeders zugunsten eines kartesischen Netzwerkes auf, das sich zwar grafisch gut auf dem Bildschirm darstellen lässt, was aber gegenüber der originalen Idee Yobs dem Spiel meiner Meinung nach viel von seinem Charme nimmt.

Wie es leider oft Praxis in der Wildwest-Ära der Computergeschichte war, wird Gregory Yob als Spieldesigner auch noch einfach unterschlagen und in der Anleitung nicht genannt.

Heute gibt es, wie gesagt, zahllose Adaptionen und Variationen – auch ich habe eine [Version in Python](#) als Fingerübung geschrieben. Ein Lösungsalgorithmus für das Spiel ist letztendlich ein [Szenario in der KI-Programmierung](#).

“Hunt the Wumpus” – ein kleines Programm mit einer großen Wirkung. Über Jahrzehnte.

Dieser Beitrag wurde publiziert am Samstag den 18. Februar 2023 um 11:00 in der Kategorie: [Spielebesprechungen](#), [Videospielgeschichten](#). Kommentare können über den [Kommentar \(RSS\)](#) Feed verfolgt werden. Du kannst zum Ende springen und ein Kommentar abgeben. Pingen ist momentan nicht erlaubt.