Bob Alberti 15741 Words

11/30/2010 ed.

The Scepter of Goth

Program for Individualized Learning

PIL 3251 – Project in Prior Learning

University of Minnesota

Table of Contents

Introduction

You walk into a dank, stone-walled room and find a hungry tiger chained to the far wall. It paces back and forth in front of a crude shelf upon which you can see the glittering jeweled wand that you have been sent to retrieve. But how to get past the tiger?

Suddenly Artemis arrives in the room beside you. Where he can see that you have the armor and sword of a warrior, you see that he wears the robes of a wizard.

You've heard of Artemis before, but you don't know him, and you're not sure whether he is going to attack you or not. Before you can decide whether to launch a pre-emptive attack of your own, he speaks.

“How do we do this?” he asks.

“Well, I can distract the cat while you grab the wand,” you say after a moment's thought, “but we can't both retrieve the wand.”

“I just need money,” Artemis says, “I'll give you the wand if you pay me for it.”

You quickly agree on a modest sum, and put your plan into action. As you attack the tiger, Artemis attempts to dodge past and grab the wand. But the tiger wheels and claws the wizard, hurling him across the room with a powerful swipe.

“Are you okay?” you ask, backing away from the angry tiger..

Bloody but unbowed, the wizard holds up the jeweled wand. You offer him the agreed upon amount, but he declines to accept it.

“You were supposed to keep me safe,” he says, “the deal is off.”

Angry at this betrayal, you attack the wizard, but he vanishes in a cloud of greasy black smoke. After he disappears you can hear his laughter echoing across the land.

MMORPG's

This kind of scenario is typical of the kind of interaction you might experience when running an MMORPG. The unwieldy acronym stands for Massively Multiplayer Online Role-Playing Game, an interactive computer adventure game for multiple players. The most famous examples of MMORPG's are *World of Warcraft*, or perhaps the older *Everquest*. Nowadays the experience described above would be conveyed on a computer screen with animated images of warriors, tigers, wands and wizards, and text messages or audio channels might carry the dialog between the characters. The obsessive, near-addictive appeal of such games is so widespread that it has even been comically portrayed in four seasons of the popular web video series “The Guild” created and written by Felicia Day[1].

But more than a quarter century ago this same interaction was already possible, albeit without graphics. Multiple players could share a game space (the room), communicate with each other, exchange items (money, the wand), fight a common enemy (the tiger), and engage in complex social interactions (bargaining, betrayal). Lacking today's computers with high speed graphics and communications, the players used text-only computer terminals and exchanged information at the rate of about 30 characters per second over the telephone.

For calendar year 2010 the market for MMORPG games is estimated between $8 billion[2] $11 billion[3] of revenue, but the industry has its roots in a number of parallel developments that began in the 1970s and spawned *Scepter of Goth.* (*Scepter*) the first commercial MMORPG. *Scepter* was the primary offering of the GāmBit MultiSystems company created by Alan Kleitz, Gerry Leone, Robert Alberti, Sr. and myself in 1983. *Scepter* was based on one of the earliest non-commerical MMORPG games, *Milieu* which had been created in 1978 by Alan Kleitz.

Along with other early innovations in online gaming such as Control Data's PLATO games and *MUD1* by Richard Bartle, *Scepter* helped launch the present-day multi-billion-dollar MMORPG industry..

Background

My involvement in computers began during my freshman year of high school in 1975, when I sat down at a computer terminal in my outstate Minnesota school connected via a telephone modem[a] to a mainframe computer in St. Paul run by Technology Information Educational Services, now known as TIES. One of the older students showed me how to play a game called *Lunar Lander*. Only six years after the first moon landing, the objective of *Lunar Lander* was to safely guide your descent module to a landing on the Moon. Since this was a text based game printed on a lengthy scroll of cheap yellow paper, the game display was simply this:

```
M-----------------L
```

As the lander, represented by the "L", descended toward the moon ("M") this became:

```
M---L
```

A second line of text indicated descent velocity in meters per second. Your only input was simply a number representing the desired thrust for the next turn. The goal was to descend quickly enough to avoid exhausting your fuel and crashing, but slowly enough to survive impact.

I played a few times and crashed a few times, but something was bothering me. I didn't like the "M--L" display. I wanted a curve for the lunar surface and a little rocket ship for the lander, like this:

```
)------>
```

Also, I thought that when you fired your engines, the flames should appear:

```
)----*>
```

Curious about how to make this happen, I typed LIST to inspect the BASIC language source code for the program...

There is a scene in J. K. Rowling's "Harry Potter and the Sorceror's Stone" where the protagonist first rides a flying broomstick.

---

[a] "Modem" stands for "modulator-demodulator," a device for translating digital signals into audio tones for transmission across telephone lines.

> "He mounted the broom and kicked hard against the ground. Up, up he soared. Air rushed through his hair and his robes whipped out behind him, and in a rush of fierce joy he realized he had found something he could do without being taught. This was easy! This was wonderful.[4]"

I've always enjoyed this scene, because this is how I felt when I started making my changes to the *Lunar Lander* game. I needed no instruction to understand and change the BASIC language source code. I changed the display to show the characters I preferred, and spent the rest of my computer time adding new and more gruesome types of crashes to the unsuccessful endgame. With a thrill like the one experienced by the young wizard, I knew I had discovered my career in computers.

Minnesota's 'Silicon Valley'

All around Minnesota, other young students were experiencing the same thrill of discovery on computers that had suddenly appeared in their schools. TIES was initially funded by a federal Title III grant to promote computers in Minnesota schools.[5] Emerging from the University of Minnesota's Education and Research Development Council, (ERDA) in 1967, TIES was established under the joint powers act as a cooperative of state school districts[6]. Today TIES is owned and run by 41 member Minnesota school districts, to provide instructional and administrative computer services.

At the time, Minnesota was a world leader in computer technology. Local industry pioneer and TIES past president Dale LaFrenz said, "Minneapolis-St. Paul was the computer capitol of the world; this was where all the action was as we made most of the computers here. Univac's home-base was here; Control Data's home-base was here; Honeywell was here and in the computer business at the time; IBM was down the road in Rochester.[7]" Minnesota in the 1960's and 1970's was Silicon Valley, before Silicon Valley.

TIES used three Hewlett-Packard mid-range computers to serve its member districts, and the statewide system was on a Sperry Univac.[8] Soon after discovering the computer terminal in our school I was programming on both the TIES Univac mainframe, as well as a similar mainframe provided by MECC – the Minnesota Educational Computing Consortium. MECC was created in 1973 out of the state's desire to consolidate and manage the rapidly expanding field of educational computing in Minnesota.[9] These organizations were leaders among a number of organizations all working to place computer terminals in Minnesota's schools in order to introduce computers to the student curricula.

By 1976 students from all over the state could dial in to the MECC and TIES mainframes over banks of specially multiplexed phone lines. Using system commands such as “WHO” these students were able to see other accounts that were logged in, as well as some minimal information about what those accounts were doing. Between these clues as well as word of mouth, students soon established communications by writing some early examples of on-line forums. Chat rooms and interactive games soon followed.

Peer Pressure

The influence of this interactive capability, primarily available on the MECC Cyber 73 timesharing system (which provided more interactive options than its TIES counterpart) cannot be overstated. Populated largely by teenage males, the competitive drive spurred intensive innovation among students.

Speaking of a similar contemporary programming environment, former PLATO[a] programmer Don Gillies said, “Programs would get deleted, and then some other person would go in and try to 'out-do' the previous game, and so in the space of about 4 years we probably went through 20 different variants of [D&D] and sorcery-like games. This was very healthy and kept people playing the games, which were always changing.[10]" A similarly competitive environment existed on MECC.

Within a few months, a number of programs were available on the MECC system that have come to be considered standard on-line fare: FORUM*H7LT309 offered topical discussion conference boards. Early chat programs had been developed, such as *Captain Collins Talk* (*CCT*) written by Stephen E. Collins. *CCT* included private channels and specialized verb commands (comparable to contemporary

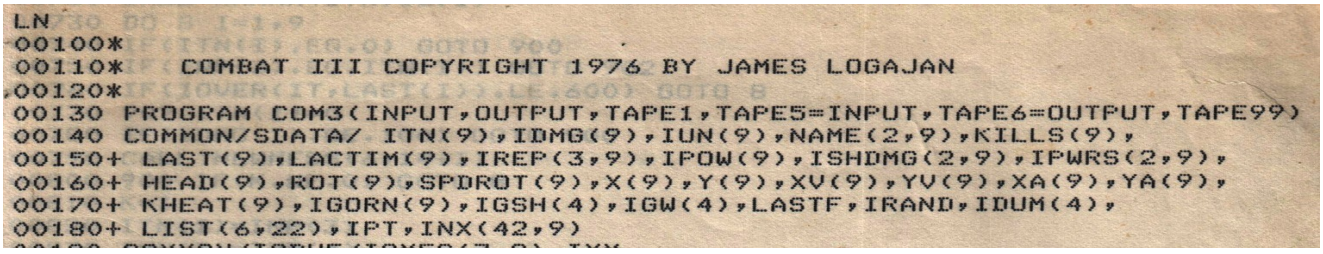


Illustration 1: Combat' source code, first page

[a] More about PLATO below.

MMORPG 'emotes') such as “wave” and “laugh.” Sophisticated interactive games were available such as *Combat* written by James Logajan (Illustration 1).

Combat was a multi-ship space-fighter game played in two dimensions, and employing a text display of azimuth, velocity, and heading to describe the positions of the various players:

```
# Dmg S1  S2 Heading Spd  Dist Azimuth Course
1 17%  5% 30%   0.0   55  2312   33.2  -45.3
2 29% 42% 17% -33.2  117
>>PING<< Laser hit from ship 1 on shield 2 caused 5% damage.
```

Table 1: 'Combat' sample display

Because the game was designed to be played on a very slow-printing TTY 33 teletype, the game display (Table 1) was kept as terse as possible.

The Heading of 0.0 indicates that Ship #1 is pointing directly at the player's Ship #2. It is moving at a Speed of 55 km/sec at a Course -45.3 degrees to the direction of its nose: in other words, it's slewing sideways (to its left and ship #2's right) while approaching, a course that would describe a curve on an X-Y grid. The Azimuth means that Ship 1 is 33.2 degrees the right of the nose of Ship #2. All speeds are relative to a fixed center point of the playing field.

Because Ship #1 is pointed directly at Ship #2, it has fired its laser and hit. Since a ship's front shield, S1, covers a 60-degree arc centered on her nose, and the second shield, S2, covers the rest of the circumference of the ship, the laser has hit the second shield resulting in the “>>PING<<” message. (Had the damage from a single shot exceeded 10%, that message would have read “**BLAM**.”) It is likely that Ship 2 has intentionally rotated to take the laser hit on its rear shield in order to distribute damage across both shields, since her first shield is already damaged. Ship 2 may now rotate to point at Ship 1 to fire its own weapons, but may have trouble aiming: Ship 2 is, as mentioned, moving somewhat sideways relative to Ship 1, so Ship 2 will have to rotate *more* than 33.2 degrees to complete its rotation pointing directly at Ship 1. Determining exactly how far to rotate to fire the next shot is part of the skill that the pilot of Ship 2 must exercise.

This brief example illustrates that, despite the lack of graphics capability, a text-only game such as Logajan's *Combat* could nonetheless convey a compelling sense of competition and gamesmanship. *Combat* could accommodate more than half a dozen simultaneous players as well as four aggressive robot bases called “Gorns” (named for an alien species on *Star Trek*). The availability of games such as

these on the MECC system as early as 1976 offers some idea of the kind of intense software development being practiced at the time by the students using the MECC system.

MECC's Influence

Describing its influence on the state, MECC's former president Joshua Coventry said, “Although its efforts are largely forgotten today, MECC was perhaps one of the most influential and important organizations in the field of educational computing to have ever operated. It was also more successful and had a much larger impact than earlier projects and initiatives to improve education and increase the awareness of computers in education.[11]”

It was in this environment that the game *Milieu*, written by Alan Klietz, first appeared.

A Short History of Computer Games circa 1978

The first computer game was *Nimrod*[12], a computer built to play the game *Nim* by John Bennett in 1951, which debuted at the 1951 Festival of Britain. *Nim* is a puzzle game, and more puzzle games followed as computers evolved, including *Noughts and Crosses* (tic-tac-toe) written by Alexander Douglas in 1952[13], *Tennis for Two* by William Higginbotham in 1958[14], and *Mouse Maze* in 1959 from Doug Ross and John Ward at MIT[15].

Arguably the first “reality simulation” program was *The Management Game*, created in 1958 at the Tepper Business School at Carnegie Mellon University[16]. This game allowed students to take the role of companies to compete against one and other, using a Bendix G-15 computer and a weekly run of 3000 punched cards to carry out the simulation[17].

The 1960's and 70's saw a number of games by a number of authors, including Baer's Magnavox Odyssey console[18], the *Galaxy Game* by Pitts and Tuck[19], *Computer Space* by Atari founder Bushnell[20] and finally his *Pong* in 1972.

PLATO

In parallel with many of these efforts, although less well known at the time, were games developed on the Control Data PLATO computers in the 1960's and 1970's. PLATO systems developed a number of

fascinating features years ahead of their time, including graphic monitors, and the first touch screen interface[21], PLATO Notes (the predecessor to Lotus Notes)[22], and a graphic multi-user interactive game space battle game called *Empire*. PLATO's architecture was designed for multi-user programming primarily focused on educational applications, and among the multi-user role playing games under development in 1975 were *dnd* by Gary Whisenhunt and Ray Wood, *PEDIT5* by Rusty Rutherford, and *Moria* by Kevet Duncombe. Many of these games are available for review on http://www.cyber1.com, a site devoted to the history of PLATO software. These games were influential within their own sphere, and led to the development of later games such as *Airfight*, which eventually became Microsoft Flight Simulator, and the *dnd*-based *Telengard* produced for home computers by Avalon Hill[23].

Early Role Playing Games

The first "role-playing" computer game was *Hunt the Wumpus*, written by Gregory Yob in 1972. Like many of its predecessors it is a puzzle game, but uniquely for its time it is a first-person puzzle, placing the player in the maze and employing text descriptions of visual, tactile, and scent clues that help the player achieve the goal of destroying the dread Wumpus from the safety of an adjacent room.

Also known as "The Colossal Cave Adventure," *ADVENT* (developed on a PDP-10, filenames were limited to six characters[24]) was written by Internet pioneer William Crowther in 1975. Many of the characteristics of *ADVENT* were carried forward into *Milieu*, *Scepter*, and subsequent games.

ADVENT is a single player game in which you are the protagonist in an underground adventure. You may move freely through the rooms of the game, restricted only by such obstacles as locked grates, crevasses, and monsters. You may take, drop, and in a limited sense manipulate the objects found within the game. Most of the objects have a particular purpose to which they apply, and failing to do so can render the game unwinnable.

The connection between Wumpus and subsequent programs can only in inferred. "Crowther's program [owed] its turn-based conversational style to... tabletop wargaming. Nor was the program without precedent – *Hunt the Wumpus* by Gregory Yob in 1972 (*Wumpus*) was a textual maze game, while *SHRDLU* (Terry Winograd, 1972) had a recognizably adventure-like parser[25]."

Many similarities suggest a connection between *Wumpus* and *ADVENT*. In *Wumpus* one moves through passages with directional exits, encountering monsters and pits, and seeking a goal (the

Wumpus), as in *ADVENT*. Like *ADVENT*, rooms in *Wumpus* are modeled on a real space – the surface of a dodecahedron.

Unlike *ADVENT*, rooms in *Wumpus* are described only by their number, and by the clues offered by adjacent rooms. There are items in *Wumpus*, but they are not manipulable: e.g. you may take coins, but you may not later drop them, nor may you drop the arrows with which your character is initially equipped. And where *Wumpus* can be considered a single-puzzle game with the goal stated in its title, part of what made *ADVENT* more appealing is that it is made up of multiple puzzles scattered across a common gamespace.

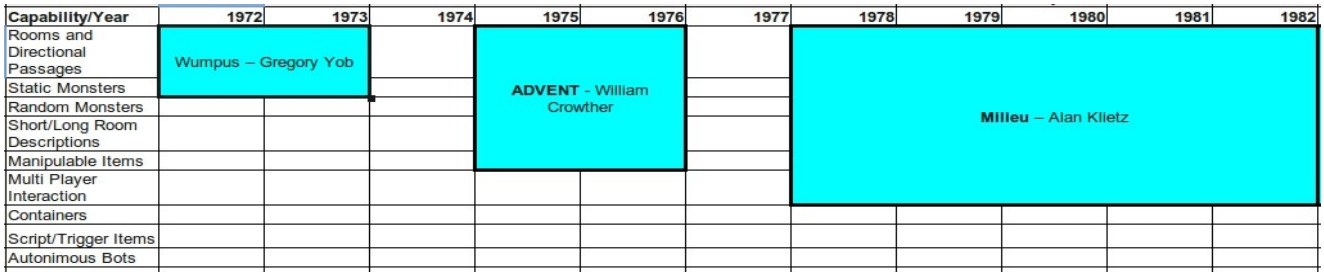


Illustration 2: Feature comparison between Wumpus, ADVENT, and Milieu

According to Klietz, he was inspired by *ADVENT*. "I wondered how to make *ADVENT* multiplayer. How would interaction work? There needed to be some sort of rule system, one that was simple enough to program into the computers of the day[26]. A student named Mike Pritchard wrote... a 'Talk' program... and it all clicked: I could write a 'super Talk' too, have it show prompts like *ADVENT*, but be multi-player using the *AD&D*[a] rule-set."

ADVENT featured elements subsequently found in *Milieu*, including: text-described "rooms," some of which were programmed for unique behaviors; "monsters" which appear in a given room; "items" which can be manipulated by the player; and the freedom of the player to move at will between rooms and to freely manipulate items. Illustration 2 summarizes the relationships between the programs.

[a] TSR, Inc. published *D&D* in 1974, and *AD&D* in 1977.

Table 2: ADVENT room descriptions

AT END OF ROAD
You are standing at the end of a road before a small brick building. Around you is a forest. A small stream flows out of the building and down a gully.

> south

IN A VALLEY
You are in a valley in the forest beside a stream tumbling along a rocky bed.

> north

AT END OF ROAD

> south

IN A VALLEY

> look

IN A VALLEY
You are in a valley in the forest beside a stream tumbling along a rocky bed.

Most notably, *Milieu* and *ADVENT* both feature rooms with long and short room descriptions. Upon entering a room for the first time, a complete room description is provided. Thereafter a brief description is provided unless the player deliberately re-examines the room with a LOOK command. Table 2 demonstrates this by showing how the room description of *ADVENT* changes as one moves north and south between two rooms.

While the PLATO programs *dnd* and *PEDIT5* programs predated *Milieu*, Klietz had never heard of these programs at the time that he developed *Milieu*. "Our school didn't give us access to PLATO. My only access was on MECC[27]." Inspiration for *Milieu*'s operations and commands was drawn directly from the *ADVENT* command structure and parser.

Milieu

Klietz's roleplaying game on the MECC system went through a series of names: *E*M*P*I*R*E*, *Scepter of Goth*, *Milieu*, and *Ghost*. For simplicity's sake all of the MECC program versions are referred to in this document under the single name *Milieu*, and *Scepter of Goth* is applied only to the commercial version.

While *ADVENT* provided the style for the *Milieu* suite of games, its single, nondescript protagonist offered insufficient guidance for a game intended to be multi-user and interactive. That *ADVENT* portrays a human protagonist is clear: the character appears in a mirror as a human figure, has a limited number of items it can carry, and both eats and drinks. But for an interactive, multi-user game – in which one's protagonist character must be viewed from the perspective of the other players – more detail is required. Klietz decided to base his character rules on *Advanced Dungeons and Dragons* (*AD&D*) from TSR, Inc[28]., and offered a variety of classes of characters from which players could select, offering the players variety and a sense of control.

Milieu was the most popular game on MECC, with some users logged in eight hours per day. After receiving complaints from the parents of some users,[29] Klietz eventually had to modify the game to shut down betwee midnight until five a.m. to ensure his users got a night's sleep.

Despite its popularity, all was not well with *Milieu*: according to Klietz the game was "awash with gold pieces," and high-level players were casually handing out powerful weapons to new players, so Klietz decided to reset the game database. With the assistance of his dungeon masters (DM's were volunteers who worked within the game to assist players and maintain game balance) Klietz engineered a dramatic game reset.

High level characters were offered a series of challenging adventures, the goal of which was to seize the magical Scepter of Goth. When a character finally succeed, he unleashed a devastating cataclysm. Klietz personally issued the game-wide text broadcasts that described the sky darkening, thunder, and earthquakes that led to the database reset, in what was possibly the first example of interactive multi-user live theater.

Scepter of Goth

In 1983 the State of Minnesota decided to shut down the MECC mainframe in favor of purchasing stand-alone Apple II computers for every school. As far as MECC's users were concerned, this was a catastrophe. An analogous contemporary example might be if the United States government announced that, with regret, the Internet would be turned off, and instead the public was urged to watch new three-dimensional televisions that would be made available. Few would be satisfied with such a trade-off.

Amid the outcry over the State's decision, I saw a business opportunity. I had begun using MECC in 1976 and was still a regular user seven years later, so I was aware that hundreds of teenagers and college students like me around the state would be suddenly cut off from their computer-based community. There was a significant business opportunity for whoever could step up and replace the e-mail, chat, and game services to which the MECC users had become accustomed.

I contacted Alan Klietz, who I knew as the author of the most popular game out of the many available on the MECC system, *Milieu*, and together with two other partners Klietz and I formed GāmBit Multi Systems (GāmBit).

Fortunately for our venture, Klietz was not merely an exceptional programmer, but is arguably a genius with every aspect of computer operations. In order to make his *Milieu* program work on the very limited MECC mainframe, for example, Klietz had manually created a memory paging system using the NOS overlay loader[30]. This allowed portions of his program to be swapped in and out of the limited memory space as needed, an advanced architecture not generally employed in computing until Sun introduced dynamic shared libraries in the late 1980's[31].

Additionally, while CPU time was limited, the peripheral processing units or PPU's – sub-processors that handled disk input and output – did not count against the CPU time limits. Klietz's design allowed him to take advantage of the "free" PPU time and quadruple the program's player-handling capabilities. Klietz kept his source code off-line, so other programmers could not figure out his unique architecture, and some baselessly accused him of gaining assistance from MECC staff. However there was no sign that MECC staff were capable of implementing this architecture themselves.

When GāmBit was under development, Klietz was not only prepared to rewrite the *Milieu* program in a different language on another computer, he was also able to configure the architecture of the computer itself to improve its ability to run his program.

Scepter Architecture

In retrospect, it seems incredible that so much could be accomplished with computers whose processing power was less than that found in most present-day digital watches. In 1985 the GāmBit computer system cost $1353 and was comprised of the following parts:

- 1 Seagate ST225 20MB hard drive
- 640K Dynamic XT Motherboard
- 2 256K RAM chips
- 1 8088 CPU Chip
- 1 8087 Math Chip
- Floppy controller and 5 1/4" floppy drive
- 1 Monochrome graphics video card
- 1 XT I/O card
- 1 8-port Hostess board (later upgraded to 2).

In addition to the mainstay program *Scepter of Goth*, a number of other services were available, including

- E-mail

- Online topic based forums
- Interactive chat
- *Foreign Intrigue*

The interactive chat program included private channels, private person-to-person communications, emotes, and eventually "Jeeves," an automated butler who could assist guests and new users when no system operators (sysops) were available by answering frequently-asked questions.

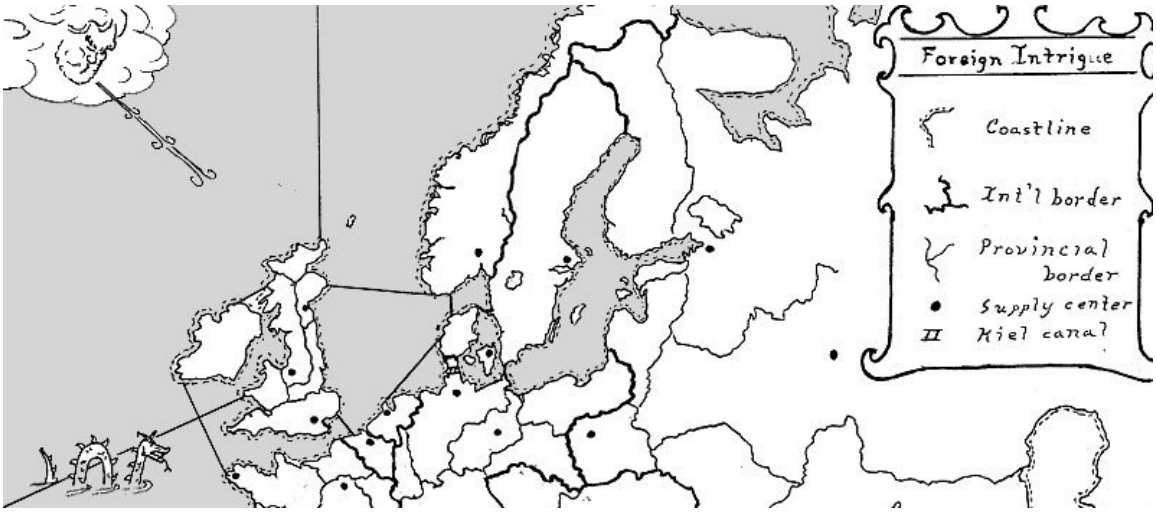


Illustration 3: Map for Foreign Intrigue drawn by Bob Alberti

Foreign Intrigue was based on the *Diplomacy* board game published by Avalon Hill[32], and was a turn-based interactive game that allowed players to compete to conquer Europe. Messaging features in the GāmBit forum software allowed for the negotiations and backstabbing for which its parent game was renowned. In order to avoid copyright problems, I drew the game map by hand from a turn-of-the-century atlas, which gave *Foreign Intrigue* a different set of countries from *Diplomacy*. The game appealed to a different set of users than *Scepter,* and could be played without extensive periods online. Subscribers on a budget could play *Foreign Intrigue* very inexpensively.

Scepter *Classification*

In his book of 2004 "Designing Virtual Worlds,[33]" Richard Bartle provided the terminology with which to describe the construction of games such as *Scepter of Goth*, the "virtual worlds" of the title.

In Bartle's terms, then, *Scepter* was similar in architecture to the later "DikuMUD" class of programs, in which the program mechanics (the "driver"), the routines that described the physics of the world (the

“mudlib”) and the world model were all coded directly into the *Scepter* runtime engine. As an original program, all of the code was custom written by Klietz, first in Multi-Pascal (as *Milieu* on the MECC system) and then in 'C' (on the 8088-based PC). There was no scripting language in *Scepter,* an issue that Klietz addressed in the subsequent development of his *Screenplay*.

Scepter Design

The game of *Scepter* was centered around the town of Boldhome, a northern coastal city-state surrounded by the untamed wilderness of a collapsed empire. This placement allowed for a variety of game environments, from frozen wastes just to the north to mountainous caves to the east, and even underwater settings were envisioned.

Practically speaking these settings had to be implemented within the existing game engine, through such gimmicks as forcing players to dig their way out of a snowdrift by creating a passive blocking monster with a lot of hit points called 'a snowdrift,' rather than with the implementation of a new game command for “dig” that could be applied to a new item called a “snowdrift.”

This is not to say that new features were not being consistently coded, but such new capabilities were expensive in terms of programming and debugging time, particularly since GāmBit did not have a pre-production testing environment. Instead, “playtesting” accounts were auctioned off to subscribers who wanted to be the first to explore new features of the game.

The Story

Based on Klietz's experience with M*ilieu, Scepter* was intended to have a “story arc” leading towards another eventual reset of the character database. While Klietz was diligently studying game theory at the time, neither of us were sure it would be possible to prevent the game economy from going seriously out of balance again, and foreshadowing another reset seemed like a good idea.

To that end a master story arc was constructed to lead to this next reset, which could be held off indefinitely with additional modules, or hastened if the game's balance went badly awry.

Notes from the original game design include the idea of the Phoenix, a powerful entity neither good nor evil, that has somehow obtained the 'Scepter of Goth,' which is intended to protect Boldhome. Without

the Scepter's protection, evil will inevitably encroach upon Boldhome (thus explaining the monsters all about). Arching over all of the various puzzles and quests in *Scepter*, then, is the quest to restore the game's eponymous MacGuffin.[a]

Original handwritten notes on the game's design indicate the structure of the overall story arc:

1) Nobody knows where the Phoenix is (it isn't anywhere to start with!)
2) Clues among 3-6 (key) quests
3) When final battle occurs, player who destroys Phoenix will likely be evil. When Phoenix dies, move to Phase II, 'Thralldom of Evil.'
    1. As Phoenix becomes more wounded, will smoke and burst into flames (treasure from previous encounter, "Armor of Flame Protection,' get from a dragon?)
    2. Beginning of next phase, castle will be named for evil character, objective of phase: destroy the castle.
    3. Change character class names to Thralls (fighters,) Morlocks (wizards,) Ur-viles (clerics), etc.

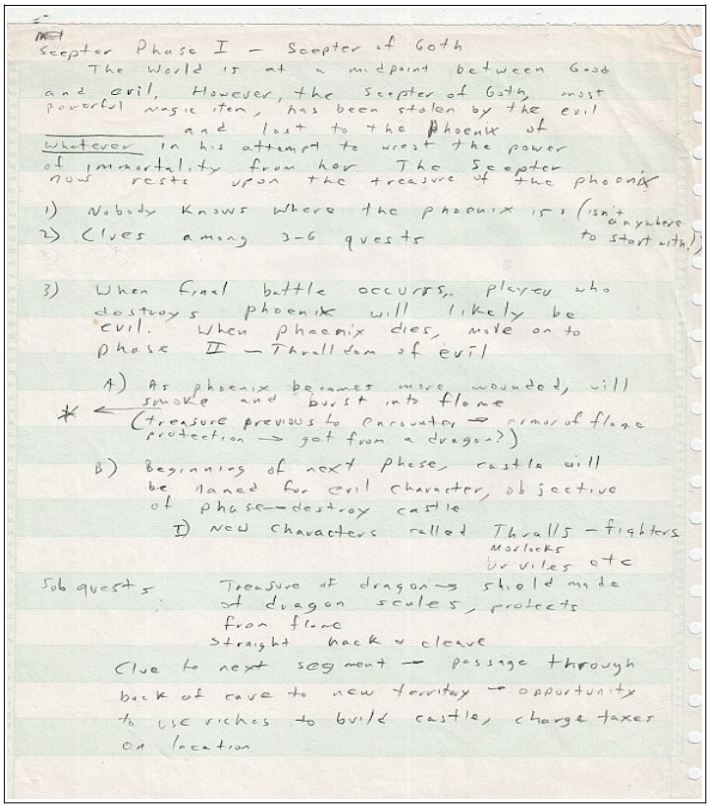

Scepter Phase I – Scepter of Goth
The World is at a midpoint between Good and evil, However, the Scepter of Goth most powerful magic item, has been stolen by the evil _____ and lost to the Phoenix of _____ in his attempt to wrest the power of immortality from her. The Scepter now rests upon the treasure of the phoenix
1) Nobody knows where the phoenix is (isn't anywhere to start with!)
2) Clues among 3-6 quests
3) When final battle occurs, player who destroys phoenix will likely be evil. When phoenix dies, move on to phase II – Thralldom of evil
A) As phoenix becomes more wounded, will smoke and burst into flame
(treasure previous to encounter → armor of flame protection → get from a dragon?)
B) Beginning of next phase, castle will be named for evil character, objective of phase – destroy castle
D) New characters called Thralls – fighters Morlocks Ur viles etc
Sub quests Treasure of dragon – shield made of dragon scales, protects from flame straight back v cleave
Clue to next segment – passage through back of cave to new territory – opportunity to use riches to build castle, charge taxes on location

Illustration 4: Original Scepter game design notes

While *Scepter* as run by GāmBit never did run all the way through to the Phoenix endgame, this story structure helped inform and guide the various quests and puzzles created, and provided a thematic background to every element of the game.

Associate DM's

Before discussing the *Scepter* database, it will be helpful to discuss the role of the Associate DM's (ADM's) who helped write it. On MECC, Klietz had brought on board volunteers to help him run *Milieu,* and GāmBit followed suit. The work involved in developing *Scepter* was considerable, and

[a] A plot element that catches the attention, and which the characters in a story are willing to do almost anything to obtain.

both Klietz and I wore many hats in our small company, so even assistance correcting typos was welcome.

However as a for-profit enterprise, GāmBit could not afford to be reckless about selecting its ADM's, and so to that end I created an Associate DM questionnaire, which I distributed at an in-person meeting of those desirous of the role. (A pity that we weren't as careful about our programmers, really.)

The one-page application included six questions, to be answered in essay form:

1) Explain briefly what you believe is the function of a DM in a fantasy role playing system. What should or shouldn't he do?
2) Explain what might be the differences between the DM of a standard FRP game and a Scepter DM. Take into consideration the fee the players pay, and the fact that there is rarely face-to-face contact.
3) In ONE SENTENCE of 25 words or less, explain why you wish to be a Scepter DM. (Bonus: reduce this one sentence to two words, verb and noun).
4) Player AAA plays Scepter a lot, and frequently makes requests of the DM's for restoration of his character (always with a good reason). Player BBB plays less heavily, but regularly, and has never requested restoration when killed. AAA is fighting a monster called an "AAArachnid" when BBB arrives and accidentally kills the already-wounded AAA by typing "THRUST AA". BBB runs from the room, erasing AAA's items, and AAA (at half his original level) finds and kills BBB, taking his items. Player AAA requests his original level and items back. BBB requests his original level and items back. What do you do?
5) Player Manfred has been playing Scepter for several hours when the system crashes and his character is erased. Due to the crash, however, he was not charged any credits for the time he was on. He requests that his character be restored to where it was before the crash. What do you reply?
6) You have noticed that player Bluto spends a lot of time fighting Wazoo monsters, which he claims are his favorite. One day you notice a message of the DM board which mentions that Wazoo monsters were found to have a typographical error, and were worth five times the experience points that they should have been. The message says that this was discovered after a DM heard Bluto bragging to another player about "easy experience points." Do you penalize Bluto, and if so, how and why?

There were no fixed "right" answers for these questions. Since much of a DM's work involved writing, spelling and grammar were as important as sound judgment and an eye for the commercial realities of serving paying customers rather than other scholastic mainframe users (as had been the case with *Milieu*.) Answers, for example to #4, ranged from the unacceptably stern ("I wouldn't restore either of them and explain to both of them that if BBB would have been more careful and AAA hadn't flown off

the handle this wouldn't have happened") to the impractical ("I would tell BBB to tell AAA he was sorry for killing him.") to the vague ("Not restore character.").

I designed these quizzes with the thought in mind that I was going to have to work closely with my ADM's, and I wanted to assess candidates for their overall suitability with my own approach. While others might have designed their tests differently, I am an intuitive decision maker and was not interested in any kind of scored testing. I decided to simply go with the candidates whose answers I liked best overall. My approach was rather successful – not only did I select some good Associate DM's, but a number of my ADM's have remained friends for the past quarter-century.

ADM Instructions

Once selected, ADM's received free access to the game, but could no longer participate as players while remaining ADM's (hence selecting ADM's also entailed balancing skilled *Scepter* players against the loss of income). In addition to a thick technical instruction manual, they also received a two-page set of behavioral guidelines. Including some of the following advice:

> Associate DM's repair spelling errors they discover, flawed or "zapped" rooms, erase unused rooms or indexes, and notify Master DM's of balance problems. Associate DM's "fine tune" new areas and dungeons, role-play monsters, observe the activities of players, and add "spice" to the game through editing.
>
> ADM's must be fair in their response to player comments, striking a compromise which is firm but satisfying to the customer.
>
> DO WORK – Have something you can point to as an accomplishment, however small, every time you log off.
> DON'T PLAY SUPERMAN – ADM's are something between a god and a janitor, avoid playing your ADM as a character. You are there to serve and entertain, not interfere.
> DO PARTICIPATE – remain apart, but try to alter the destinies of players for the good.
> DON'T HEX ANYONE – even with the most annoying players, don't be "out to get" anyone.
> and DO REPORT YOUR ACTIONS – whenever you make a major change, report it to the DM board.

DM's

DM's emerged from the pool of ADM's quite naturally, with the most committed, reliable, and creative rising to the top. Nothing functional separated DM's from ADM's (or "Master DM's" as I and the other

owners were described) except the organizational hierarchy imposed by the owners. As a rule, ADM's were supposed to remain visible as often as possible, while DM's were expected to be available for pages but usually invisible, and Master DM's to be invisible most of the time. This freed the DM's and Master DM's from most interruptions, and helped keep the ADM or DM from having to justify tough decisions to angry subscribers by passing the buck up to a superior (a DM or Master DM). Being invisible, the DM or Master DM could delay responding in order to allow an angry customer to cool down.

In addition to adjudicating disputes for Associates, DM's were responsible for all aspects of creating Dungeon adventures (see below). DM's were responsible to design everything, starting with the concept, creating the rooms, monsters, and treasures, and coming up with innovative tricks and traps based on the capabilities of the game's parameters (see for example “Game Object Characteristics” below). Every room could access a single index of six random monsters, each of which could drop an item from a random six-object list of treasures, meaning that any room could generate between 1 and 36 different treasures. Additionally, DM's could create “permanent” monsters, intended to occupy a room until destroyed, and these were usually very powerful “boss” characters guarding customized, powerful, permanent treasures as a reward (such as a tiger guarding a wand).

Boldhome

Scepter had three distinct although arbitrary map scales, “dungeon,” “town,” and “countryside.” Two of these can be seen in Illustration 5, where the map of Boldhome is bordered by squares 150 through 153 of the surrounding area. Each 'room' of the town was assumed to be about the size of a city block. At ten blocks to a mile (the scale of the Minneapolis streets where I lived at the time) Boldhome is about 1.5 miles long on a side. Each “countryside” square was assumed to be about a mile across. These scales were arbitrary, insofar as no effort was made to make these mathematically precise.

As a nod to the subscribers (and to relieve my already heavily-taxed imagination) all of the streets of Boldhome are named after the actual Twin Cities (Minneapolis and St. Paul) metropolitan area street addresses of GāmBit's initial subscribers. This was intended to give subscribers a pleasurable little flash of recognition as they moved through the game. In the end, however I and other local *Scepter* DM's are instead periodically reminded of Boldhome as we travel around the Twin Cities.

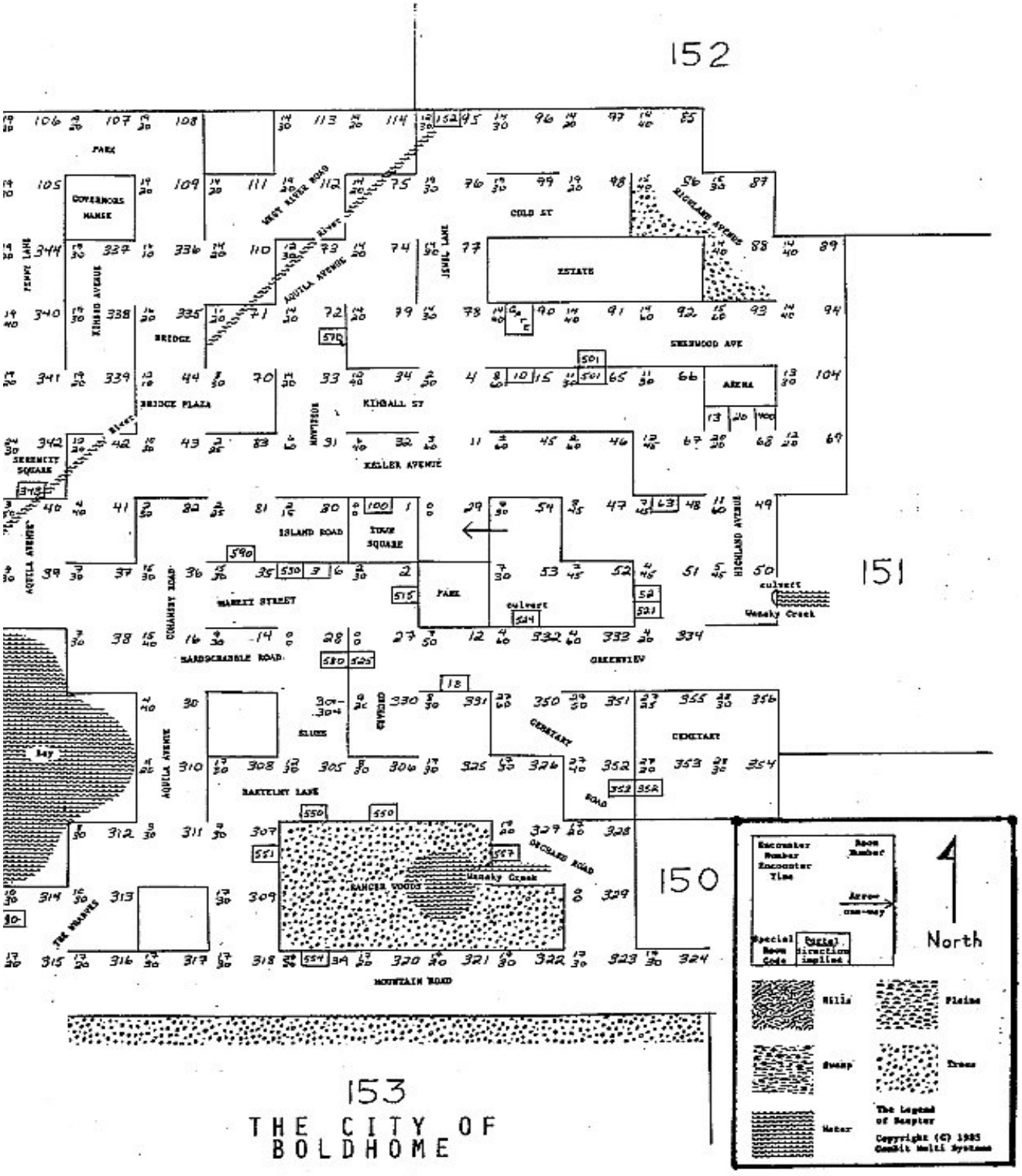


Illustration 5: Original Scepter DM's Boldhome map with room numbers and legend

Boldhome was designed to be the center of essential services. The shops, armor-repair room, or the level-advancement rooms could have been distributed anywhere, but for the sake of lower-level

characters they needed to be located where such characters could shop and train for advancement without being killed by a passing monster along the way.

Boldhome was designed to provide sufficient access to adventures for low-level (1-5) characters. The streets of the town were designed to offer a challenge only to level 1 and 2 characters, in the form of the occasional rogue thief or rat. Level 3-5 characters could find adventures in areas accessed from Boldhome, but designed to be still "in town," such as the culvert to the underworld in room 53, or the Ranger's Woods at the south end of the city.

While it was perfectly possible for any room in *Scepter* to link to any other room, DM's were instructed to maintain verisimilitude by ensuring that rooms and areas had exits that were logically consistent. Additionally, exits into and out of Boldhome were kept to a minimum as can be seen in Illustration 5. The only exits are along the riverbank from room 95 to 152, through a culvert from room 50 to 151, and through town gates in rooms 334, 324, and 315, exiting to 151, 150, and 153 respectively. Due to the differences in scale, it was impossible to offer a 1:1 correspondence of exits to entrances, and so the concept of a city wall was employed to provide logical consistency for these travel restrictions.

Boldhome itself was outfitted with an array of "neighborhoods" to provide variety to lower level players including the wealthy "governor's" area northwest across the river to tempt thieves. The northeast quadrant included the jewelers of Gold St., and the haunted Estate for clerics to get in their exorcise, and an arena to the east was intended for interactive gladiatorial games although it was never implemented, but did house the Fighter's training area. A culvert just east of Town Square provided access to rat-infested sewers, and a cemetery to the southeast was chock full of ghosts and ghouls. To the south were the dangerous slums, beyond which the Ranger's Woods included some basic animals and the Ranger training area. Southwest along the bay were the wharves, where low-level fighters could find drunken dockworkers to beat up. And all throughout Boldhome different buildings and underworld tunnels were created to provide variety and challenges to up and coming players.

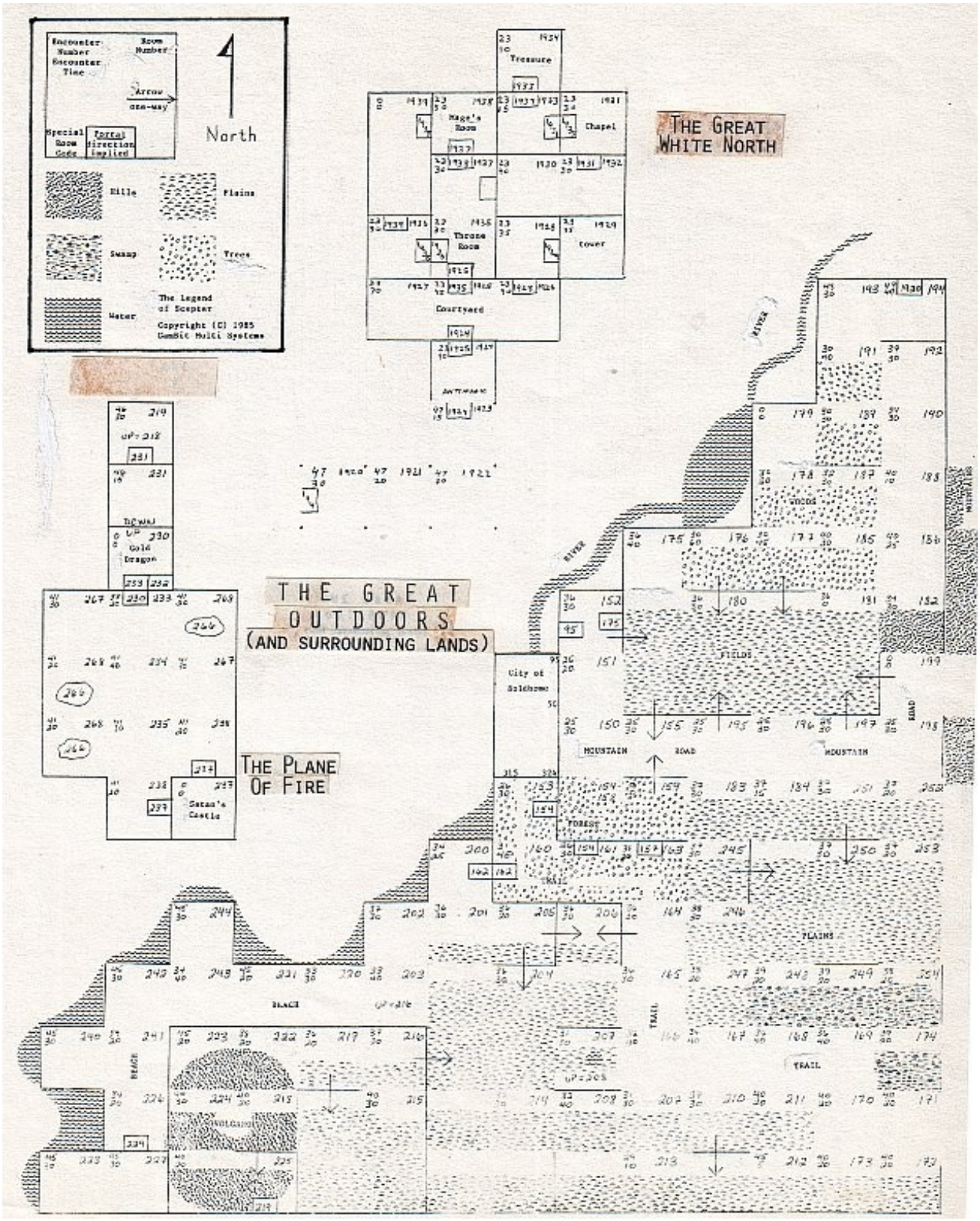


Illustration 6: Original 'Great Outdoors' map, complete with glue discolorations

The Great Outdoors

Surrounding Boldhome at a much larger scale were The Great Outdoors. These offered a variety of terrain types, including forests, grasslands, mountains, and tundra. While they were designed with challenges and riddled with random encounters, they were intended as the commons between various adventure areas created by the DM staff.

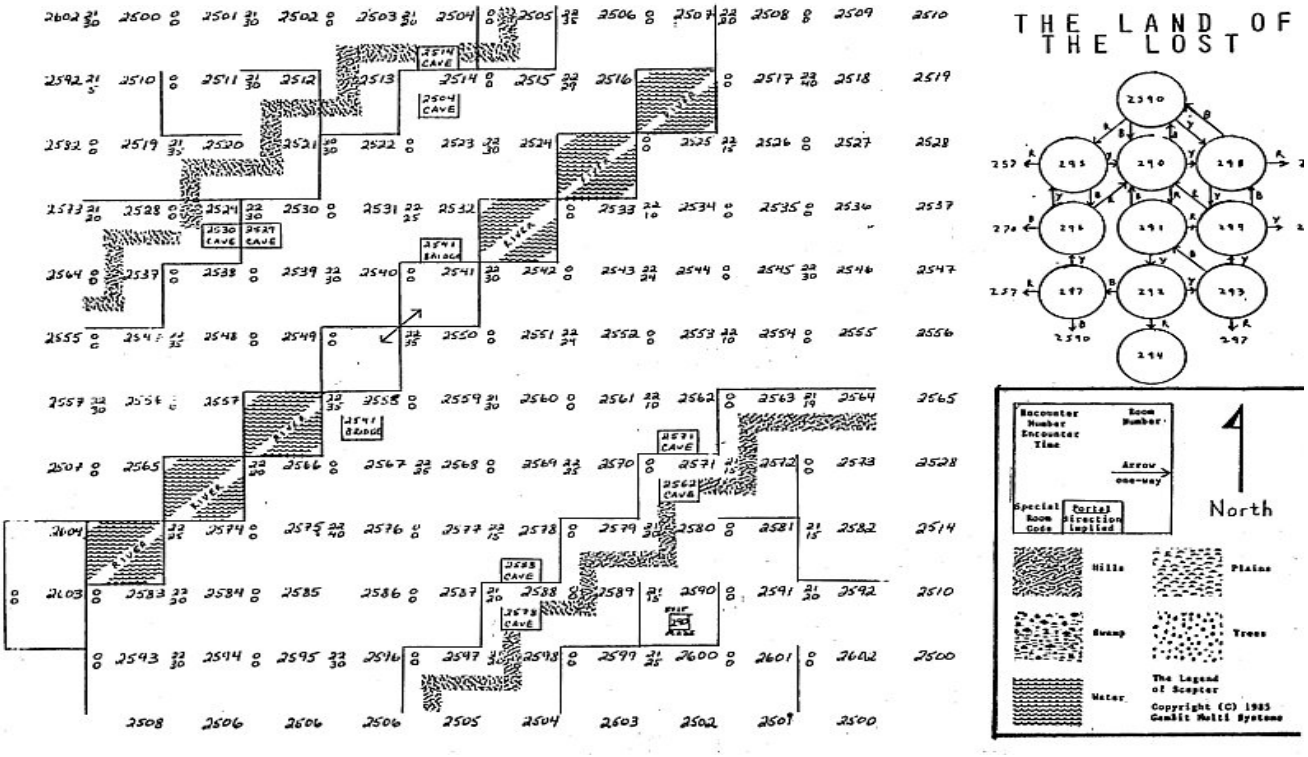


Illustration 7: Lands to the east of the Great Outdoors

The areas were built with many one-way exits spread across areas, designed to channel players back to common central areas (e.g. "The Land of the Lost," upper-right of Illustration 7). While it was possible to map these areas by dropping permanent items (and many players eventually did), the initial effect was to create the impression of a much larger space. I drew this design concept directly from the "maze of little tiny passages, all alike" in ADVENT, which appeared dauntingly large until mapped by dropping items.

In certain areas of *Scepter's* Great Outdoors one could travel almost indefinitely in a given direction while only moving between three or four actual rooms, often becoming astonishingly lost. Most new ADM's were amazed to learn how small was the actual gamespace.

Dungeons

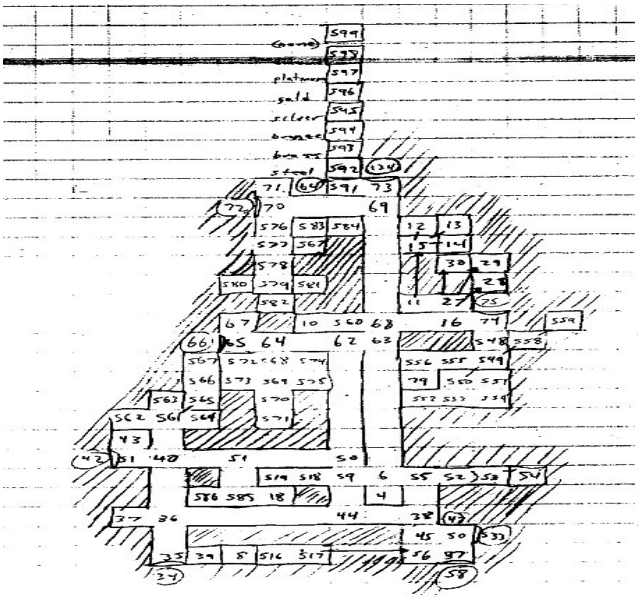


Illustration 8: A "dragon" based dungeon

Standalone, modular game areas were generally referred to as "dungeons," and were subject to only a very few limitations. They needed to fit logically into one of the existing game maps (Boldhome or the Great Outdoors) and have exits that made physical sense (e.g. a complex of caves stretching north-south, with exits to a hilly area of the Great Outdoors in adjacent north-south squares 2562-2571).

Most of these dungeons had a single-goal "minons-boss-treasure" structure like that in Illustration 8. Adventurers can enter and leave through one of several openings (circled, around the perimeter), but the ultimate treasure was only available at the end of a series of increasingly-powerful dragon caves toward which players were funneled.

Frequently the treasures available in such dungeons, whether part of the random treasure indexes or the permanent treasures held by the boss, were designed to address particular balance problems in the game. If clerics were complaining that the blunt weapons to which they were restricted were weaker than their non-clerical counterparts, then the next dungeon built would be likely to include a stronger blunt weapon among the random treasures. Or possibly the boss at the end would have a particularly powerful and permanent blunt weapon.

While most dungeons were standalone, there were plot threads to the overall "Phoenix" story arc that needed to be planted throughout the game (Illustration 4). Most of these I wrote myself, such as the Ice Castle located in the Great White North, which contained clues that were to lead to the discovery of the Phoenix.

It was while writing the Ice Castle that I developed the notion of “inimical rooms” that was later incorporated into *Screenplay*, rooms where the player took damage just for being in the room. The Ice Castle could then have had rooms that cost a player four health points a turn unless they wore “fur” armor specifically designed to warm a character. Or underwater, a player might lose ten health points a turn unless they wore “scuba gear.” Unfortunately these capabilities were not present in *Scepter*.

Database Structures

There were several program databases associated with *Scepter*.

The Off-line Character Database

This stored all characters and their possessions. It was possible for a DM to alter offline characters, to create objects in their inventory, restore characters who were “dead,” etc. Characters that died rolled against their constitution: if they made the check they lost a single level, but if they failed they lost half their levels and a random characteristic score (Strength, Intelligence, etc.). Players who successfully appealed a “death” usually wanted their levels, possessions, and statistic restored.

The Active Character Database

This was stored in the computer's memory, and represented those characters whose owners were active in the game.

The Monster Database

The templates for all monsters appearing in the game. Each monster was linked to an index of objects, and a random item from that index would be dropped when the monster was defeated.

The Oindex Database

The object index database was an arbitrary list of six item numbers which could be 'dropped' by a defeated monster as a random treasure. Item numbers could be repeated in the list to increase the chance of their appearing. So if a DM wanted most orcs to drop silver, but a few to drop gold, the list

would include the index number for silver coins five times and that for gold coins once. DM's were instructed to use these lists thematically, configuring the items to be dropped according to the levels of the players intended for the encounters with the monsters to be encountered. For example, orcs should drop orc-armor, not dwarven armor, unless the setting were perhaps a dwarvish hall that the orcs had presumably raided.

The Mindex Database

Lists of six monsters from the Monster Database. DM's were instructed to group monsters according to appropriate anticipated player levels that would encounter the monsters, and thematically according to the setting of the game. However there was nothing preventing any monster index from being used in any room. As with objects, monster index values could be repeated in the list of six to increase the chance of that monster appearing in an associated room.

The Edescript Database

Arbitrary long descriptions for items. Most items did not require long descriptions, but for those which did these numbered index lines were available, and followed the same display rules as room descriptions (see below).

The Object Database

The numbered templates for all objects appearing in the game.

Generic object characteristics

All objects shared the following characteristics.

Object names were set according to a 'C'-like descriptor of the format “noun, description %s”. For example, a gold sword would bear the description “sword, gold %s”. In order to take the sword, the player would have to refer to it by its noun, e.g. “take sword” would work, but “take gold sword” would not. This could be used by DM's for game purposes – for example were a sword described as being embedded in a stone, the DM could set the description to “Excalibur, sword embedded in a

stone.” Without the %s, the player would see only the description “You see a sword embedded in a stone.” However, the command “take sword” would not work: the only way to take the sword would be for a player to learn the sword's name (possibly as the goal of a quest) and use the command “take Excalibur.”

Each object had a configurable “article” parameter, e.g. “a,” “an,” “some” etc. So players would appropriately see “a sword,” “an axe,” or “some corn”.

Objects could be hidden by setting them to be “invisible.” They would be found only if a player used a 'search' command in the room.

Objects could be set to be permanent, meaning they would be stored as part of the room database if the program shut down. Normal, impermanent items would vanish during periodic garbage collection if left alone in a room.

Each object could be set to be 'takeable' using the “take” command, or not. So a DM could set Excalibur to be untakeable (even if its proper name were known to a player) until such time as the DM was sure that someone had completed the quest. This would prevent lucky guesswork from bypassing the quest, but required DM's to be very alert to activities within *Scepter*. As we shall see, the program had features facilitating such monitoring.

Every item also had a weight, which was summed with other carried items in order to limit the objects a character could carry, based on the character's strength.

Object types

There were twelve regular object subtypes: armor, shields, and weapons; treasure and money; doors, portals, and teleporters; magic items and scrolls; chests and keys.

Armor, shields and weapons

Armor and shields had a parameter indicating the amount by which they increased a player's defense.

Weapons had upper and lower ranges of damage, and (if they were set to be “magical”) the amount that they increased a player's chance to hit successfully.

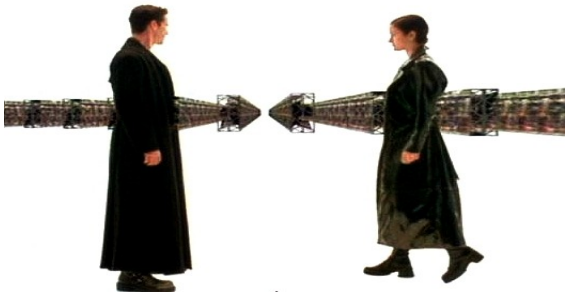


Illustration 9: Neo and Trinity inspect the Object Database of 'The Matrix'

All of these objects had a total number of hits they could take, or strikes they could deliver, before breaking. This latter was added to reduce the proliferation of armor and shields within the game, and allowed players to spend money to repair items with the help of the game's repairman (located in a customized room, see below).

Treasure and money

Treasure and money had a value parameter in "shillings," the base currency of the game. Money had a "multiplier" by which an item could multiply its value. So one silver coin would have a multiplier of 20 and a value of 20 shillings, 2 such coins would have a value of 40 shillings. One copper coin had a multiplier of 1 and a value of 1 shilling.

Doors, portals, and teleporters

Doors could be closed or open, locked or not, and could be trapped or not, and were linked to a destination room. For lock functionality, seen "Chests and keys," below.

Portals were filters for player level: they could not be locked, closed, or trapped, but they had upper and lower limits for the levels allowed through to the destination. Their utility in restricting areas of the game to players of the appropriate level is obvious, but no particular effort was made to gracefully introduce them into the game environment – a player attempting to pass through a portal restricting their level simply received the message "Only players of levels 3 to 5 may pass through this portal."

Teleporters were objects that, when invoked, would transport a player to a set destination. Teleporters could optionally be restricted to work only from a particular room so, for example, an object called "Rapunzel's hair" might be set to work only from the room at the base of Rapunzel's tower.

Magic items and scrolls

Magic items and scrolls were linked to the spell that would be executed when they were used, and for magic items, the number of charges remaining. Potions were simply another kind of magical item.

Chests and keys

Containers were objects that could contain other objects, usually referred to as Chests. Chests could be closed or opened, locked or trapped, just like doors.

Locks on doors and chests bore three-digit codes, e.g. 854. The third digit ("4") indicated the odds in ten of failing to pick the lock (40%). A lock ending in "0" could not be picked (100%).

Keys were numerically coded to open doors or chests. The three-digit key value had to match the lock to be opened. Keys ending in "0" (e.g. "850") were master keys for all locks 850-859.

DM-only objects

DM's also had access to objects called "trump cards" (a la Roger Zelazny's "Amber" series). Using a Trump moved a DM directly to the room where a given player was located with a command such as "use trump Merlin." A trump was a separate object that could be dropped, picked up by other characters, etc., but as with level-restricted portals, trumps simply informed any other player that "Only DM's can use trumps."

The Room Database

The "map" of the game, this was the list of numbered rooms in the game. All rooms 1-9999 were always considered to exist: DM's did not 'create' rooms, but 'modified' them to prepare them for use. Each room had an associated set of descriptions, a monster index, and available fixed directional exits. Other exits could be provided by the use of "door" and "portal" objects.

Descriptions

Several features existed to control what was displayed to the user. First, "slashed" descriptions were multiple descriptions of the same location, provided for the sake of variety. For example a frequently visited room might have a description that reads:

/You're in a green meadow through which runs a creek./You're in a green meadow with a gurgling creek./A gurgling creek runs through this green meadow.

Any one of the three descriptions will be randomly displayed.

The second type of description was the “short” description. This was provided by “commenting out” portions of the long descriptions with the hash (“#”) sign. So the same room description might read:

/You're in a green meadow# through which runs a creek#./

If the user has never entered the room before, or if the user employs the “look” command, they will get the full description (sans the hash marks).

If the user has visited the room before, they will only see “You're in a green meadow.”

Additionally the C-language “newline” convention of using “\n” to insert a carriage-break-and-return was employed for lengthy descriptions.

As one might imagine, this could get quite complex. For example:

/You're in a green meadow# through which runs a creek.\nThe creek gurgles over the rocks#./You're in a green meadow# with a gurgling creek.\nThe creek splashes over a rocky bed#./

The result of which could be:

You're in a green meadow through which runs a creek.
The creek gurgles over the rocks.

And upon re-entering the room:

You're in a green meadow.
> look
You're in a green meadow with a gurgling creek.
The creek splashes over a rocky bed.

Other Parameters

Rooms had several configurable characteristics in addition to their directions and descriptions. The “encounter type” was the Mindex entry tied to the room, and “encounter time” was the frequency with which new encounters were generated. Rooms could be set as Anti-Magic wherein no magic would work, and as Safe Havens where guards would intervene to prevent violence.

Finally, rooms could be flagged to Notify any DM's who happened to be on line when a character entered the room. This was set in the Guest area (below) to alert a DM to attend a visitor, but could also be set in special areas of the game where the invisible intervention of the DM's offered oversight or might enhance the experience.

Hells

Some of the collections of rooms were not “connected” to the primary map of the game, but represented maps under development, private DM storage areas, and special game areas to which a character needed to be manually transferred by a DM. Characters could be “imprisoned” in these disconnected rooms (known as “hells”) for committing infractions of the game rules, or simply for annoying the DM's. Since room location was stored with the character data, even logging out and back in would not free a player from a DM hell.

Guest area

One isolated room collection was the “Guest Area.” This was where visitors who were not GāmBit subscribers could get a tour of the game, speak with a DM, and eavesdrop on game broadcasts (such as arrivals and departures of players, and “yells” that players issued.) Guests could be transferred into the main play area by a generous DM, but this was discouraged, as weak unarmed Guests were frequently abused by players and this did little to encourage subscriptions.

Players of recently deceased high-level characters quickly learned to log back in as Guests in order to draw a DM's attention and explain how a cosmic ray or apocryphal program “glitch” had caused their high-level character to unjustly be defeated in battle.

Specialized rooms

Certain rooms had special features which were hard-coded into the game engine: Room 1 was the Town Square, a safe room where newly created subscriber characters arrived; room 9 was where guests arrived to be greeted by DM's; and room 18 was the repair shop. Additionally rooms 20-26 were the Training (level advancement) rooms for the various character classes. Rooms 1 through 30 were “fast attach” rooms, frequently-used Boldhome town locations held in memory at all times to reduce disk accesses and speed up the game.

Characters

Based on *AD&D*, character statistics such as Strength, Intelligence, and Dexterity have become conventional MUD fare, and will not be reviewed here. However, a few of the more unusual features shall be explored.

Classes

In addition to the commonplace character classes derived from *AD&D* which users could select:

- Cleric
- Fighter
- Magic-user
- Paladin
- Ranger
- Thief

Scepter also included some unique classes that could only be set by a DM:

- Assassin
- Barbarian
- Caretaker
- Lady
- Town Mayor

Assassins benefited from increased odds of successfully hiding and backstabbing, and did not suffer penalties for killing other players (see below). This class required the DM to also set the player characteristic of "assassins guild member" which charged the player a daily fee based on their level for continued use of the character class.

While player killing was discouraged, there were times when it was convenient or simply politic to allow players to administer punishment to other players, rather than requiring the DM's to always intervene. If one player was deliberately instigating or annoying another, he might find his target temporarily reset to Assassin class and freed of player-killing restrictions by a sympathetic DM.

The Barbarian class was restricted because it was too strong. Barbarians that teamed up could often defeat anything else in the game. This is the same lesson Rome learned, really.

The Caretaker class was most often used for DM test accounts. The Town Mayor was simply a character type that could make "official town announcements."

The "Lady" character class was a bald-faced attempt to lure female players to the game[a]. Lady characters had the advantages in dexterity, intelligence and magic commonly attributed to elves (in later MUDs that featured different character species.)

Other Characteristics

In addition to the standard characteristics of Strength, Intelligence, etc., *Scepter* offered some (for their time) unique characteristics:

Hidden, Invisible, and Nonexistent – successively more concealed. Hidden and Invisible characters could still be seen in the "Who is online" status. Hidden characters could be found through searching, Invisible characters through guessing the character's name and searching for or talking to them. Nonexistent characters were completely concealed, not even appearing in the system-wide status display, available if players exited the game to the Main Menu. Nonexistent characters could only be detected if they spoke or broadcast a message. Players could Hide themselves with that command, and could use magic to become Invisible but Nonexistent was restricted to DM's.

Playtester and DM – Scepter quite successfully auctioned off playtesting roles to customers willing to play while the game was subject to crashes and resets. Other players could not enter the game at these times.

Thieves Guild, Assassin's Guild – bonuses to pick locks, steal, and other advantages already discussed, in exchange for a daily fee.

Evil – Character would not be attacked by aggressive monsters, but *would* be attacked by town guards and townspeople. Could only be set by a DM and was discouraged from frequent use.

Piety – Piety was decreased by killing other players, and lower piety aggravated town guards the way the Evil setting did, but also made the character more likely to be attacked by aggressive monsters, and reduced chances of successful revivification[34].

---

[a] As we shall see in the "Demographics" section, this was an utter failure.

Talk Limit – Imposed on the characters of players who abused the “Say” and “Yell” commands.

Monsters

In addition to commonplace statistics such as hit points and level, Monsters in *Scepter* had the following interesting characteristics:

Parley – One of forty different values guided monster responses to attempts by players to speak (parley) with the monster. One setting caused the monster to offer to sell an item from its Oindex treasure list, making the monster a merchant or shopkeeper. Other settings were negative comments, positive comments, or random of either. Some settings prompted the monster to flee, others to attack, or pick any response at random. Finally, one option caused the monster to teleport the player randomly to a location in the wilderness outside Boldhome.

Monster responses to Parley were processed like room descriptions, so multiple responses could be coded, with the first occurrence of “%s” being monster name, and the second %s the player name. For example a barkeep with a parley descriptor of

/%s says, “Hello %s”/%s growls, “So it's %s!”

would alternately produce

> say “Hello”
Tom says, “Hello”
The barkeep says, “Hello Tom”
> say “Greetings”
Tom says, “Greetings”
The barkeep growls, “So it's Tom!”

The Assistance setting caused the town guard to intervene in a player attack against a character, relieving the player of their primary weapon. More obtrusive was the No Kill characteristic that simply told the player that they had changed their mind about attacking a monster.

If set to Flee, the monster would always flee if attacked.

The “Follow” setting meant that a player who fled might be pursued by the monster.

“Attack Last Aggressor” set to true responded to whoever attacked most recently, but set false caused the monster to only attack whoever hit it first regardless of the actions of the rest of the party. When “Attack Last Aggressor” was set false and combined with “Follow” set true, a player who struck first and fled might lose the support of their group and have to fight the pursuing monster alone.

“Block” caused a monster to prevent players from leaving the room.

“Defend” set true meant a monster only attacked back if “Attack last aggressor” was true. Otherwise a monster could be set to not fight back at all. For example, DM's could make passive monsters with a lot of hit points such as “a snowdrift,” which players could then attack until it was destroyed, dropping an exit portal called “a path through the snow” as a treasure.

Set to “Guard,” a monster would prevent players from picking up any objects in a room (for example, a tiger set to guard a wand on a shelf).

“Moral Reaction” set to true meant that a monster would attack a player with low Piety, or with Evil set to true.

If set to “Kidnap,” a defeated player would not be killed, but would instead be placed in a prison room, with the monster on guard in the next room. Rescue adventures followed.

Monsters set to “Anti Magic” were immune to spells, and set to “Magical” were immune to non-magical weapons (weapons without a +1 or better to hit).

Monsters set to Steal would steal items from a character, and could build up quite a treasure trove before being defeated and dropping all items.

Other monster characteristics included Level Drain, Invisible, Poisonous, Spell Casting, Undead (vulnerable to Clerics), Rust (monster strikes damage weapons and armor) and Regenerate Hit Points.

The Customers

In an early attempt to categorize the kinds of players who participated in MUDs, Richard Bartle *didn't* invent the 'Bartle Test[35].' Rather, Bartle wrote a paper on the theory[36], and the test was developed by Andreasen and Downey.

The Bartle Test categorizes players into Achievers, Explorers, Socializers and Killers (A, E, S, and K, respectively).

Bartle Types

As a commercial venture, *Scepter* strongly discouraged player-killing, which tended to drive off customers. In order to maximize interest and online time the fee-per-hour game featured many puzzles based on the interactions of the various room, object, monster, and player characteristics already discussed. Meanwhile, a chat room was available at a much lower per-hour rate for users who wished merely to have discussions.

As a result many forces existed to exclude Killers and Socializers and shape the *Scepter* player base into Achievers and Explorers.

Demographics

Only one undated copy of the subscriber list remains, but based solely on the apparent genders of the first names of the subscribers, the *Scepter* database appears to have contained 114 males and 9 females.

Murderers

Two of *Scepter*'s subscribers were defendants in murder cases. Bill Couture[37] subscribed with the handle Ziggy and was a stalker, who killed the boyfriend of his obsessive interest. He escaped justice for several months, until he was caught setting fire to the home of his interest's next boyfriend. Couture was convicted in criminal court. Norm Wartnick (Fuzzybear) was accused in 1986 of murdering a business rival in 1973, the civil case continued for ten years.[38] Wartnick settled the civil suit in 1996 for $1.3 million.

Wartnick was, incidentally, one of the most vocal and hard to satisfy subscribers, actually phoning me at home at 3:00 a.m. to insist that his character had been killed by a computer bug and must immediately be edited back to life. Due to his temper and the exceptional amount of money he spent on *Scepter*, requests such as these were usually granted, and as it turns out this was probably just as well[a].

---

[a] This is actually the reason that I have since that time had an unlisted home telephone number.

GāmBit's Development

So involved were GāmBit's operators at simply getting *Scepter* and the other games up and running, advertising and marketing to customers, writing the game database, and keeping the customers happy, that very little attention was paid to the books. It came as rather a surprise when, at the end of the first year, the hired accountant recommended that the $24,000 in profits be distributed, lest they be taxed at corporate rates. While GāmBit was beset with more troubles in its short existence than most small businesses, it never actually did operate in the red. By 1985 GāmBit was generating profits after expenses but before payroll of approximately $70,000/year.

The First Programmer

While Alan Klietz was, as mentioned, an exceptionally skilled programmer, and I was a competent one, there was a lot of work to be done both keeping the operating system functional, fixing bugs in various programs, and writing new programs. So when the budget allowed, GāmBit attempted to hire programmers to assist with growth.

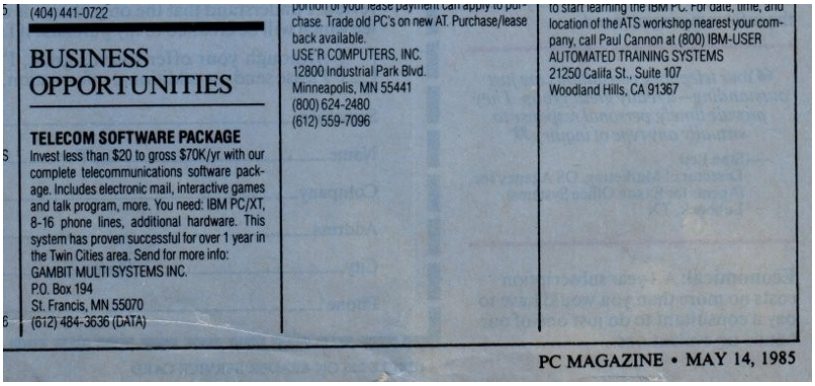
The first programmer, hired in late 1984, copied all the system's software and attempted to sell it by using GāmBit's own chat room. When caught “red handed” and fired in early 1985, he immediately used the stolen software to set up a competing system at the rate of $10/month (in contrast with GāmBit's approximately $2/hour rate.) GāmBit's partners, perhaps unwisely, did not believe taking legal action was advisable, expecting that the pirated system would fail without ongoing programming support before any legal action could even commence. And indeed, the pirated system lasted less than a year, which was nonetheless enough time to put a significant dent in GāmBit's finances and local reputation.

Many of GāmBit's customers abandoned it for the new system. Regrettably, several of my closest friends were among GāmBit's subscribers who left for the rival system. As his employer I was legally prohibited from discussing the programmer's dismissal, and my friends decided to believe his version of events, unrestricted as it was by law or fact.

Franchising

Faced with the sudden collapse of the subscriber base, the decision was made to reclaim the idea back from the fired programmer, and sell copies of the GāmBit system as franchises to other cities.

At about $10,000/franchise for 13 franchises, GāmBit went from reduced revenues to about $130,000 in a very short time. At the time, merchant licenses to process credit cards were hard to come by, but GāmBit had an advantage – it had been established by renaming a prior corporate shell ("Restaurant Concepts Ltd.") that already possessed a merchant license so that credit card subscriptions could be accepted from day one. In addition to their franchise fees, several franchises needed to use GāmBit's merchant account in order to process their subscriptions, and GāmBit took a percentage of every transaction.


(404) 441-0722

BUSINESS OPPORTUNITIES

TELECOM SOFTWARE PACKAGE
Invest less than $20 to gross $70K/yr with our complete telecommunications software package. Includes electronic mail, interactive games and talk program, more. You need: IBM PC/XT, 8-16 phone lines, additional hardware. This system has proven successful for over 1 year in the Twin Cities area. Send for more info:
GAMBIT MULTI SYSTEMS INC.
P.O. Box 194
St. Francis, MN 55070
(612) 484-3636 (DATA)

chase. Trade old PC's on new AT. Purchase/lease back available.
USE'R COMPUTERS, INC.
12800 Industrial Park Blvd.
Minneapolis, MN 55441
(800) 624-2480
(612) 559-7096

location of the ATS workshop nearest your company, call Paul Cannon at (800) IBM-USER
AUTOMATED TRAINING SYSTEMS
21250 Califa St., Suite 107
Woodland Hills, CA 91367

PC MAGAZINE • MAY 14, 1985

Illustration 10: This PC Magazine ad brought in ~$130,000.

The Second Programmer

Using the franchising funds, GāmBit had purchased a powerful Unix minicomputer, and Alan Klietz had begun work on *Scepter's* successor, *Screenplay.* Programming support for GāmBit was needed more than ever.

We set about hiring another programmer to help support the system, undeterred by losing our customer base (including as mentioned a number of good friends of mine) to the first programmer. The second programmer stole my girlfriend of three years, but at least he didn't set up a competing company.

Buy Out

At this point GāmBit's management was starting to falter. Alan Klietz was focused on *Screenplay* and GāmBit's new server, named Unison. In addition to e-mail and online games, Unison offered Unix shell

access and training tutorials to aspiring programmers, in the hope of capturing some of the competitive creativity that had been the hallmark of GāmBit's inspiration, MECC. Plans were in place to allow developers to recoup software development time by putting their own games up for operation on Unison.

Illustration 11: Advertisement for Unison timeshare

But by this time I was burned out. I had lost an important relationship and a number of longtime friends. I was conducting customer support and marketing while maintaining the *Scepter* database and fixing program bugs. And due to the time spent operating GāmBit, my academic career was a shambles.

So when one of our franchises, Interplay, Inc. from Falls Church, Virginia, offered to buy out GāmBit MultiSystems, we were ready to sell. Interplay had big plans, plans to go national using the "Tymnet[39]" network. As the offer included partial ownership in Interplay, GāmBit's owners would retain influence while gaining assistance, guidance, and resources from Interplay's staff and programmers. The new arrangement seemed poised for success.

Screenplay

Written in 1985 for Unison, a Charles River Data Systems 68000 Unix system, *Screenplay* was designed to address the primary shortcoming exhibited by *Scepter*: the inability to create scripts and programs within the game itself. Much had been accomplished in *Scepter* through the creative use of portals, monster characteristics, object indexes, etc., but Klietz wanted bigger things. Not only did he want to surpass the dynamic limitations of the *Scepter* software, he wanted to write a game engine that would allow for a variety of different types of game.

In a message on March 27, 1992 in the Usenet group rec.games.mud, Alan Klietz described *Screenplay* and its capabilities[40]

```
Screenplay was a deliberate effort to get away from RPG cliches
of its predecessor such as levels, hit points, and faux medievalism.
```

```
The game featured various themes - wild west, science fiction, mystery/detective,
and so on but with much less violence than Scepter.  Here the monsters were
intelligent -- you could carry on a rudimentary conversation similar to an Infocom
game. The object of the game was to cooperatively solve puzzles and gain knowledge
rather than to accrue levels by killing things.
```

In order to allow its authors (as DM's were renamed) to transcend the conventions of *AD&D* that had been programmed into *Scepter*, (such as experience only gained through killing, hit points, etc.) *Screenplay* offered the ability to program automatons (replacing the term "monsters"), rooms and objects.

Key to this ability were "action objects" and "timed events." An example would be a door, to which could be added an action object called "knock," a customized command just for that door. When the player typed "knock" a programmed set of events could follow. An immediate event would print the message "You hear distant footsteps approaching within," while a timed event would open the door five seconds later. At that time, the state of the door would change from closed to open (allowing passage to the next room), and a greeting would be printed saying "A butler opens the door."

Using action items, almost any action could trigger events, and using timed and immediate event delays almost endless possibilities were available. As described in the 1986 Interplay Inc. company prospectus[41]:

```
Action objects may create/delete/modify other objects, automatons, rooms, or
players, including objects in other rooms. For instance a lever pulled in one room
may slide back a bookcase in the next room to reveal a secret door.

It is possible to trigger an event that takes place ten minutes later, in another
room. Or program bad food that when eaten causes a player to later contract a
disease. Or a magic sword that, when wielded, creates a hostile automaton in
another room that begins searching for the sword-wielder.

An action can even be triggered by placing one object within another: a triangular
crystal placed in the triangular socket on the forehead of a statue causes the
statue to come to life!  The possibilities are endless.
```

An example of the underlying programming language is below, an example of code controlling eavesdropping. Note that the C-based *Scepter*-style description strings remain[42].

```
Let eavesdropper be the node's content's first_thing.
  While the eavesdropper exists, do
    Begin
      If the eavesdropper's type is user and the eavesdropper
        is not me and the eavesdropper is not the other_person
        and (random 1 to 10) is 10, then
      Begin
```

```
      write eavesdropper's mask, "You overhear %s",
        (show_name me, definite, caps).
      write eavesdropper's mask, " whispering to %s!\n",
        (show_name other_person, definite, caps).
      write eavesdropper's mask, "\"%s\"\n", txt.
    End.
    Let the eavesdropper be the next eavesdropper.
  End.
```

Screenplay replaced the term "monster" with "automaton," due in part to the fact that the same data structure was used for players and "monsters." One ramification of this was that automatons could be controlled by authors as if they were players. The town guard who normally gave only pre-programmed responses could suddenly begin behaving spontaneously and originally when taken over by an author.

Given that scripts could add to player experience points, they allowed for games that could be based on achievements other than killing, and genres of any kind. To do this, Klietz had created a high level programming language within the *Screenplay* application itself[43]:

```
The high-level language was [named] Hi, an object-oriented database query/edit
language with embedded actions.

Hi scripts were checked for correctness at run time.  Given that
the scripts were crash proof, authors were free to and indeed
encouraged to experiment with the live database.  The authors created
some pretty neat stuff, such as a sadistic doctor's office like in
Little Shop of Horrors and a play-for-your-life game show a la Smash TV.
```

Despite these features, *Screenplay* did not progress far past the prototype stage. Before going on to become a notable figure in the field of Linux security[44], David Wheeler was a programmer for InterPlay, and he posted his evaluation of *Screenplay* to rec.games.mud in 1992[45]:

```
Screenplay had a number of excellent features & good potential, but it was very
incomplete and only partially documented. It did run & could show off its
potential, but it was decided (after creating a monstrous to-do list, including
some stability and language problems) that it would take too long to finish.
```

Returning to the table of Illustration 2, we can measure *Screenplay's* capabilities against those of the programs that preceded it in its own evolution:

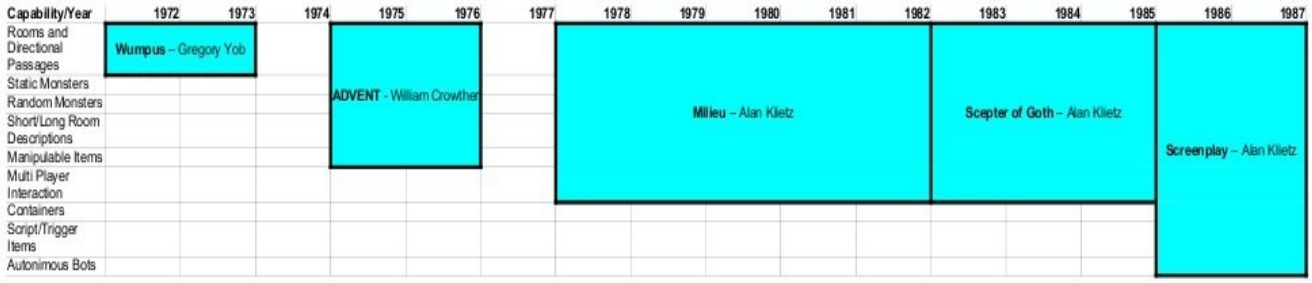


Illustration 12: Feature Comparison between early games, Wumpus through Screenplay

We can see that while *Scepter* largely repeated the capabilities of *Milieu, Screenplay* added generalized containers, scripted and triggered items, and automatons. The next logical set of features would have been graphics capabilities.

Whether the prototype Screenplay might have ever have been successful or eventually added graphics will remain unknown, because at this point Interplay collapsed under the weight of its own expenses, and the conviction of its president Denny Flanders on multiple counts of tax evasion.

After GāmBit

The collapse of GāmBit and its successor Interplay was personally quite devastating. I had lost many friends, an important relationship, and my entrepreneurial business, and it took over a year for me to get back on my feet both emotionally and financially. Even then it wouldn't have happened if I hadn't been fortunate enough to meet the woman I would eventually marry, and who helped straighten me out. Thoroughly burned out with entrepreneurship, I was recruited into a job which paid a considerably more money than I had earned through GāmBit. So for many reasons I put *Scepter* behind me.

After getting a job with the Army High Performance Research Center, Alan Klietz put GāmBit and his notoriety behind him as well.

“My co-workers ribbed me about *Scepter* and it really annoyed me, so I tried to forget about it[46].”

Meanwhile others who had been involved with GāmBit and *Scepter*, either as players or franchisees, were moving on as well. A number were inspired by Scepter to become involved with the growing online-gaming industry.

Illustration 13: Brett Vickers

Brett J. Vickers (Illustration 13) was inspired by *Scepter* in his 1988 creation, along with Brooke Paul, of *Quest for Mordor*[47], which remained a fundamental open source MUD for many years. Working for ArenaNet in 2005, Vickers was programming team lead for *Guild Wars*[48], a commercial MMORPG funded without a subscription fee[49]. Unaware of this connection, *Guild Wars* turned out to be my eldest son's favorite MMO, and he spent a lot of time at the *Guild Wars* booth when we attended the 2006 Leipzig Games Convention.

Scott Hartsman was "about 15" when he became involved in the Milwaukee franchise of *Scepter*, following which he worked with former InterPlay vice-president Tom Zelinski and David Whatley on Simutronics' *GemStone.* Hartsman's resume[50] reads like a history of MMORPG's, beginning with *Scepter,* followed by *GemStone, Darkness Falls, Everquest* and *Everquest II*. Now Chief Operating Officer for San Francisco-based Trion, Inc. this is Hartman's reply to the question of *Scepter's* influence on his career:

> "Influence? Hell - I'd say that it practically created it. If anything, it was the first time I could see how *I* could actually, **maybe,** if things worked...make games for a living.
> Up until then, [computer games] were just cool things that you bought in the store, made by people far away. It was always a "Man, it'd be awesome to do that." But never even seemed possible until *Scepter*."

In 2010 Simutronics split into HeroEngine and Simutronic Games, the latter still listing *GemStone IV* among its products after nearly a quarter century. *Scepter* player David Whatley remains President and CEO of Simutronic Games, and HeroEngine is used as an underlying technology in a number of popular MMORPG games, such as the forthcoming *Star Wars: The Old Republic*.

Illustration 14: Morgan Gerhard with Michael York

Morgan Gerhard (left side of Illustration 14) was a subscriber to the original *Scepter* site in Minnesota. In an unsolicited e-mail to me[51] he wrote,

> "Your *Scepter of Goth* game... was instrumental to me ending up in Hollywood and recording a lot of the biggest games on the market today (*Gears of War*, *Ratchet & Clank*, *Resistance, Uncharted*, etc.) I'm recording all the voices for the games. I work for Technicolor in Burbank. I

> remember my folks would yell at me when they would catch me up at 3 a.m. playing your game."

J. Todd Coleman, Josef Hall, and James Nance founded two MMORPG companies, Wolfpack Studios (now part of Ubi Soft Entertainment) and Kingsisle Entertainment.

Mark Peterson, whose interest in computers also began on MECC and TIES[52], was an original *Scepter* subscriber who went on to produce the commercial *Swords of Chaos* system.

Andrew and Chris Kirmse created *Meridian 59*[53], the first 3D massively multiplayer game, which was acquired by 3DO who in turn licensed it to Computec in Germany. Andrew Kirmse worked as a Lead Programmer at LucasArts and is now a Principal Engineer at Google, where he is responsible for *Google Earth*.

Illustration 15: Matt Firor

Matt Firor (Illustration 15) and Rob Denton attempted to set up a *Scepter* franchise but were thwarted by the collapse of InterPlay. They went on to form Interesting Systems Inc. ISI merged with Mark Jacobs' Adventures Unlimited Software, Inc. (AUSI). Mythic was acquired by Electronic Arts and as EA Mythic produced *Warhammer Online*, and as Bioware Mythic is planning *Star Wars: The Old Republic* for 2011. Bioware Mythic is still headquartered in Fairfax, VA., where Interplay was located. Rob Denton remains General Manager and Vice President of Mythic Entertainment. In 2007 Matt Firor was hired by Zenimax Media Inc., parent company of Bethesda Softworks, as president of ZeniMax Online.

Illustration 16: Mark Jacobs

Mark Jacobs (Illustration 16) was CEO of Mythic until 2009[54], but he began his career in online gaming after being turned down for a job at InterPlay. Further annoyed by their closure of his local *Scepter* franchise (with no recourse to transfer his character or refund his investment in it), Jacobs told InterPlay that he was considering setting up a competing service:

> "[Interplay] laughed at me (big mistake) and said that they had heard it before. Well, I bought a copy of QNX and... dedicated my life to designing and programming my own system. I put it up, charged less than [Interplay] did, and *Gamers World* was born."

Over twenty years later Jacobs' successful career includes the early MUDs *Aradath* and *Dragon's Gate* and, after founding Mythic Entertainment, *Darkness Falls*, *Dark Age of Camelot,* and *Warhammer Online: Age of Reckoning.*

Analysis

The Market

In the generation since *Scepter* was created, the market for MMO games has grown immensely. A 2009 report from Lazard Capital estimated that the market for online games had grown from $3 billion in 2005 to $11 billion in 2009, making it a quarter of the world computer game market[55]. Figures for 2010 claim that Activision Blizzard alone earned over $1 billion on *World of Warcraft*[56]. In China, the two leading MMO game companies, Shanda and NetEase earned 2009 profits totaling nearly $250 million[57]. With a billion citizens and a growing middle class, this market can only grow.

Scepter's Contribution

How are contemporary market figures relevant to *Scepter?* Based on a small nonscientific sampling, drawn from the 2008 survey of the "Top 20 Most Influential People in the MMO Industry[58]" several - Scott Hartsman Richard Garriott, David Perry, and Mark Jacobs – have backgrounds that include *Scepter.* And as we've seen, senior management at ArenaNet (Brett Vickers, *Guild Wars*), Simutronic Games (David Whatley, *Star Wars: The Old Republic*), J. Todd Coleman[59] of Wolfpack Studios (now UbiSoft), Rob Denton, (Bioware Mythic), Matt Firor (Zenimax Online). All across the MMO industry, former *Scepter* players are now senior management.

Scepter's Significance

By franchising nationwide, Scepter presented itself to a self-selected community of entrepreneurs. Indeed, as we have seen, many Scepter franchisees went on to start their own companies after InterPlay folded. Describing Scepter's influence, Jacobs wrote,

> "I've always referred to [*Scepter*] as one of the most important online games along with MUD I. One could argue that *Scepter* was more important in that it influenced a lot of [people]. It was

> their first exposure to an online game of any kind, in particular an MORPG. Some of these guys went on to create their own MMO's or at least work in the industry."

And as the first commercial online game, *Scepter* was not only the first online game many people had experienced, it also established immediately the concept of MMO's as a for-profit enterprise in the minds of people who otherwise might not have considered such ventures profitable. Mark Jacobs said,

> "[*Scepter's*] biggest influence on me was the fact that I saw that people would pay to play online games. That was a real game changer."

The end result is that, through its early commercial debut and comparatively wide distribution via franchising, *Scepter* inspired a number of entrepreneurs to create online game companies. Over the course of the next two decades, these individuals helped grow the industry to its present multi-billion dollar size, rising to positions of leadership along the way. Former *Scepter* players are by no means the only leaders in the MMORPG industry. However their early start provided enough of an advantage that *Scepter* seems disproportionately represented among the online gaming leadership.

Conclusion

From its humble beginnings with *Hunt the Wumpus*, online role-playing has blossomed to become a multi-billion-dollar industry. Because of its early start in 1983, and comparatively wide distribution through franchising in 1985, *Scepter* introduced a large number of entrepreneurs to the ideas of both MMORPGs and the possibility of commercializing such endeavors.

According to Richard Bartle,

> *Scepter* had an advantage over the European and academic virtual worlds in that it was based in the USA, which meant phone calls were inexpensive. In the UK, phone calls were per hour rather than flat rate, so the commercial games were charging maybe a quarter of what the phone call itself cost [which] stymied commercial development. In the US... people who started companies were able to make money... long enough to gain the experience necessary to do it large scale[60].

The real question is, what difference did *Scepter* make? If GāmBit had not commercialized *Scepter* in 1983, when would the inevitable creation of a commercial MMORPG game have occurred? Once a commercial MMORPG had been developed, how long would it have taken to become widespread enough to spark an industry?

It is possible that while the development of MUDs, already well underway by 1987, would have continued its robust pace, the idea of commercializing such efforts may have lagged. Indeed, the culture of the Internet discouraged commercialization[61], although CompuServ and America Online operated nascent royalty systems. This was illustrated in 1997, when Tim Berniers-Lee stated,

> "The idea was that everybody would be putting their ideas in as well as taking them out. This is not supposed to be a glorified television channel.[62]"

I learned this to my chagrin when attempting to license Internet Gopher software in 1993; the effort was roundly rejected by the Internet community, which still held commercial use of the Internet to be anathema, and this rejection significatnly contributed to Gopher's decline. It is not impossible that, had a proof-of-concept of commercial MMORPG not emerged during the 1980's, the advent of the Internet with its open source, nonprofit culture and resistance to commercial development, may have significantly changed how commercial MMORPG's evolved and were developed. According to Richard Bartle,

> "People who cut their teeth on *Scepter* helped shaped the way that commercialisation went, and without them we may have had fewer gamers and more opportunists directing the way that virtual world development went, which would have been to the detriment of the players of virtual worlds.[63]"

It seems likely that *Scepter's* early emergence contributed significantly to the development of the present day commercial MMORPG industry.

In financial terms, the MMO industry grew from $5 billion in 2008 to an estimated $8 billion in 2010,[64] a rate of $1.5 billion per year, or over one hundred million dollars a month in growth. Even if Scepter is only responsible for advancing that schedule by a month, the financial ramifications are significant.

Even absent financial considerations, *Scepter* contributed to an early generation of industry leaders. Richard Bartle noted that without *Scepter,* the experiential link between commercial products and the skills being used in non-commercial MUDs might not have been established.

> "...the reduced commercial experience would have [resulted in]... Fewer people who knew how to run a commercial virtual world... more companies would have had to have learn lessons [to which] the answers were already known.[65]"

Whether financially or simply culturally, *Scepter* contributed to a significantly different contemporary MMORPG industry than might otherwise have emerged.

[1] Day, Felicia, "The Guild; Season 1" http://www.watchtheguild.com, 2007.

[2] Wu, J. "The world of MMORPG: a Tale of Two Regions" Strategy Analytics: Digital Media Strategies, http://www.strategyanalytics.com/default.aspx?mod=ReportAbstractViewer&a0=5689. 5 August 2010. As reviewed by Caoili, E. "2010 MMORPG Market To Reach $8 Billion" Gamasutra.com the online service of Game Developer Magazine, part of United Business media LLC: http://www.gamasutra.com/view/news/29835/Study_2010_MMORPG_Market_To_Reach_8_Billion.php

[3] Sebastian, C. *Online games now $11B of $44B Worldwide Game Market*. Gamasutra.com June, 2009. http://www.gamasutra.com/php-bin/news_index.php?story=23954

[4] Rowling, J. K. "Harry Potter and the Sorceror's Stone," Arthur A. Levine, Scholastic. 1997. p. 148.

[5] Dale Eugene LaFrenz, OH 315. Oral history interview by Judy E. O'Neill, 13 April 1995, Minneapolis, Minnesota. Charles Babbage Institute, University of Minnesota, Minneapolis. p.11. http://special.lib.umn.edu/cbi/oh/display.phtml?id=222

[6] Private correspondence with Marilynn Taylor, Communications for TIES, 2010.

[7] LaFrenz, p.5.

[8] Ibid. Marilynn Taylor.

[9] LaFrenz, p. 22.

[10] Koster, Raph. "Online World Timeline." http://www.raphkoster.com/gaming/mudtimeline.shtml Pub. 2002, retrieved August, 2010

[11] Coventry, J. "Educational computing for the masses/" SiliconUser website. 8 June 2007. http://web.archive.org/web/20070628002639/http://siliconuser.com/?q=node/12

[12] J.M.Bennett. "Computing in Australia - The development of a profession" Hale and Iremonger, Sydney, 1994 p. 55

[13] Winter, David, "Pong-Story: Noughts and Crosses, the Oldest Graphical Computer Game" 1996. http://www.pong-story.com/1952.htm.

[14] Rabin, S. "Introduction to Game Development (2nd Ed.)" Charles River Media, 2005.

[15] "Computer Museum Report" Vol 8 http://www.bitsavers.org/pdf/mit/tx-0/TX-0_history_1984.txt

[16] "How College Students Influenced Gaming," http://www.slideshare.net/wuzziwug/how-college-students-influenced-gaming, slide 4.

[17] "Timeline," Tepper School of Business Website circa 2006. Retrieved via The Internet Archive at http://web.archive.org/web/20060309015449/http://business.tepper.cmu.edu/default.aspx?id=142904

[18] Video game odyssey. *Technology Review (Cambridge, Mass.: 1998)*, *105*(2), 96. Retrieved from Applied Science Full Text database.

[19] Purchese, R. "Pitts recalls making America's first coin-op," Eurogamer, 18 May 2010 http://www.eurogamer.net/articles/pitts-recalls-making-americas-first-coin-op

[20] Lemelson-MIT Program, "Inventor of the Week" August 1998. http://weit.edu/invent/iow/bushnell.html

[21] Rabans, Geoff, "Guide to TouchscreenTechnology" Review Explorer. Feb. 2010 http://www.reviewexplorer.com/guides/2474_guide-to-touchscreen-technology/

[22] Wooley, D. "How I Invented Online Conferencing," . http://thinkofit.com/drwool/dwconf.htm

[23] Barton, M. *Dungeons and Desktops: a history of computer role-playing games*. A. K. Peters, Ltd., 2008. p.32.

[24] Kossow, Al."PDP-10 File Specification." PDP-10 Software Archive, http://pdp-10.trailing-edge.com/BB-PBDEB-BB_1990/01/10,7/system/fispec.hlp

[25] Dale Peterson, *Genesis II: Creation and Recreation with Computers*, Reston Publications Co. 1983.

[26] Klietz, Alan, Personal correspondence, August 2010.

[27] Klietz, Alan, Personal correspondence, August 2010.

[28] Arneson, Gygax, *Advanced Dungeons and Dragons*. Tactical Studies Rules, Inc. Lake Geneva, WI. 1977.

[29] Klietz, Alan, Personal correspondence, August 2010.

[30] *Relocatable Program Structure: Compass Version 3 Reference Manual,* p. 3-5. © 1986 Control Data Corp. Http://static.cray.cyber.org/Documentation/COMPASS/3_ProgrStruct.pdf

[31] Levine, John R. *Linkers and Loaders*, Morgan-Kauffman. 1999. Ch. 10.

[32] Calhammer, Allan. *Diplomacy*. Self-published, 1959. Licensed by Avalon Hill, 1976.

[33] Bartle, R. *Designing Virtual Worlds.* New Riders. 2004. p. 46.

[34] http://groups.google.com/group/comp.sys.amiga.games/browse_thread/thread/970ea04ad1857e89/d3610f329497e9f3?q=%22scepter+of+goth%22#d3610f329497e9f3

[35] Bartle. P145.

[36] Bartle, R. A. *Who Plays MUAs?* Comms Plus!, October/November 1990 18-19.

[37] Conover, T. *Hacking.* Wired. August 1996. http://www.tedconover.com/2010/01/hacking/

[38] Rubenstein, B. *Greed, Rage and Love Gone Wrong: Murder in Minnesota*. University of Minnesota Press. 2006 p. 93.

[39] Johnson, Luanne. *Tymnet – post Tymshare*. Computer History Museum. 2006. http://corphist.computerhistory.org/corphist/view.php?s=select&cid=33

[40] Klietz, A. *Re: history: VMS Monster, Sceptre of Goth.* Usenet rec.games.mud, 24 March, 1992. http://groups.google.com/group/rec.games.mud/browse_thread/thread/8b1b97fd477c86a9/3f4c3a30b4b57bea

[41] Interplay Corporation. *General Information as of January 1986: Protocall©, Scepter©, Screenplay©.* 1986.

[42] Klietz, ibid.

[43] Klietz, ibid.

[44] Wheeler, D. *Secure Programming for Linux and Unix.* Http://www.dwheeler.com/secure-programs/

[45] Wheeler, D. *Re: history: VMS Monster, Sceptre of Goth.* Usenet rec.games.mud, 27 March, 1992. http://groups.google.com/group/rec.games.mud/browse_thread/thread/8b1b97fd477c86a9/3f4c3a30b4b57bea

[46] Klietz, Alan. Personal correspondence. August 2010.

[47] Keegan, M. *A Classification of MUDs.* Brandeis University website. 2001.

[48] *Case Study: ArenaNet*, Perforce Software website. http://www.perforce.com/perforce/success/arenanet.html 2006.

[49] Kalning, K. *'Guild Wars': An experiment that worked.* 'On the Level'. MSNBC.com. 2007 http://www.msnbc.msn.com/id/17820122/

[50] Hartsman, S. *Off the Record: About.* Scott Hartsman website. 2010. http://www.hartsman.com/about/

[51] Gerhard, M. Personal correspondence. June, 2010.

[52] Peterson, M. *The Strange Saga of The Realm of Angmar, Elfhelm's Bane, Swords of Chaos, Lords of Cyberspace, Galactic Conquest & The Muinet Entertainment BBS.* Personal website. 2010. http://www.visi.com/~spookshow/muinet.html

[53] Kirmse, A. *History of Meridian 59, 1994-2000*. Meridian 59 History website. 2000. http://sites.google.com/site/meridian59history/

[54] Dobson, W. *Mark Jacobs on his departure from Mythic and EA.* Massively website. 26 June, 2009. http://www.massively.com/2009/06/26/mark-jacobs-on-his-departure-from-mythic-and-ea/

[55] Sebastian, C. Ibid.

[56] Brightman, James. *WoW, Call of Duty Drive Activision Blizzard's Sales to $4.28 Billion in '09.* Industrygamers.com 2010. http://www.industrygamers.com/news/wow-call-of-duty-drive-activision-blizzards-sales-to-428-billion-in-09/

[57] Yang, C. *Booming online game market drives profit up.* Global Times. March 2010. http://business.globaltimes.cn/industries/2010-03/508916.html

[58] Press Release: *Top 20 Most Influential People in MMO's.* Beckett.com. May 2008. http://www.beckett.com/estore/news/?eskin=subMOG&a=9562&s=1

[59] Koster, Raph. *Economic and Political Systems in MMORPGS: Dramatis Personae*. http://www.raphkoster.com/gaming/myschyf2.shtml

[60] Bartle, R. Private correspondence. 4 November 2010.

[61] Salter, L. *Colonization tendencies in the development of the world wide web.* New Media and Society. Sage Publications. 2005. http://nms.sagepub.com/cgi/content/abstract/7/3/291

[62] Berniers-Lee, T. *Realising the Full Potential of the Web*. Proceedings of the W3C, London. 1997.

[63] Bartle, R. Private correspondence, 4 November 2010.

[64] Caoili, E. *2010 MMORPG Market to Reach $8 Billion.* August 10, 2010. http://www.gamasutra.com/view/news/29835/Study_2010_MMORPG_Market_To_Reach_8_Billion.php

[65] Bartle, R. Private correspondence, 4. November, 2010.