

The Internet and the Aspiring Games Programmer

Ian Parberry*

Department of Computer Sciences
University of North Texas

Abstract

The Internet is an important tool for aspiring computer game programmers, providing access to information, advice from peers, and electronic publishing. We examine employment prospects in the computer game industry, resources available on the Internet, electronic publishing modes, and computer games at the University of North Texas.

1 Introduction

The computer games industry, although still in its infancy, is one of the major growth areas in computing. Current generation computer games use stunning graphics, high-fidelity stereo sound, and sophisticated scenarios. Until recently, however, the programming has been of low quality. There is an expanding market for qualified games programmers, but very little opportunity for new programmers to learn the trade.

The Internet is probably the most important tool for a novice game programmer by providing access to a large:

- repository of information about games programming tools and techniques,
- peer group of established and aspiring games programmers,
- community of end-users via shareware and freeware.

The purpose of this paper is to document the use of the Internet for computer games by the aspiring games programmer. It is divided into five main sections, covering respectively employment prospects, game information available on the Internet, electronic publishing modes, and computer games at the University of North Texas.

2 Employment

Since the computer games industry is in its infancy, games companies looking to hire programmers are interested in

experience rather than college degrees. Submitting a polished resume is less important than submitting a disk of computer games that the applicant has written or collaborated on. But this situation will probably change as the industry matures. It is plausible to expect that within a decade, game companies will be looking for employees who have produced great games *and* have college degrees. The type of skills that a prospective games programmer needs that can be gained at a university include:

C/C++ Programming: For portability, games should be programmed in a high-level language. C and C++ seem to be the most popular.

Assembly Level Programming: For speed, the low-level aspects of high-performance games have to be programmed in assembly code.

Computer Architecture: Games programmers need to take advantage of advanced hardware features, for example, clocking, caching, DMA, interrupts, and RISC.

Software Engineering: Games programmers seldom work alone. This can cause major problems for programmers who are not experienced in producing commercial quality software to a deadline. Modern software engineering techniques can address some of these problems.

Computer Graphics: Stunning computer graphics and animation are a major factor in selling a game. A syllabus containing elementary 2d material plus advanced 3d material including shading and rendering is most useful.

Algorithms and Data Structures: A knowledge of standard algorithmic techniques and data structures will save the game programmer from having to constantly reinvent the wheel.

Communication Networks: This class is a must for multiplayer networked games, which are becoming among the most popular.

The additional game-specific material that games programmers need can be obtained from one of the recent spate of books on the subject, such as Gruber [2], Hook [3], Howard [4], LaMothe [5, 6], Lampton [7, 8], and Robert [12]. The Internet also provides many important resources, which are discussed in the next section.

*Author's address: Department of Computer Sciences, University of North Texas, P.O. Box 13886, Denton, TX 76203-3886, U.S.A. Electronic mail: ian@ponder.csci.unt.edu. URL: <http://hercule.csci.unt.edu/~ian>.

3 Game Information on the Net

3.1 Newsgroups

There are more than a hundred newsgroups that are relevant to games, mainly in the `alt.games.*`, `rec.games.*`, and `comp.sys.*.games` hierarchies. Also useful are the newsgroups in the `comp.graphics` hierarchy for computer graphics techniques. The most important newsgroup for the games programmer is `rec.games.programmer`. However, the traffic in that newsgroup is high, and much bogus and misleading information is posted there. Source code for some games are posted to `comp.sources.games`.

3.2 ftp Sites

Much information about computer games is available by anonymous `ftp`. Table 1 lists a few common `ftp` sites, the best of which is `x2ftp.oulu.fi`. Be warned however that some of the material out there is pirated, copyrighted, or just plain illegal. Games programmers should not include code, images, or sounds taken off the net in their game unless they are certain of its provenance. Big companies can and do sue to protect their rights to the material regardless of the monetary cost or expected return.

For the games programmer, the biggest resource is the PC Games Programmers Encyclopedia [1] which is a collection of text files of varying quality written by many different authors covering various aspects of programming games for the PC.

3.3 WWW Sites

Some game information is available on the World-Wide Web. Table 2 gives some useful URLs.

4 Publishing Modes

Electronic publishing of computer games typically utilizes bulletin boards, Internet `ftp` sites, and commercial online services (such as CompuServe). Several different modes of publishing have emerged:

Freeware

A *freeware* program is one that is distributed electronically with no payment required. This avenue can be used by a beginning games programmer who is building a portfolio of games that can be used to impress a prospective employer. The term “freeware” has been trademarked by Andrew Fluegelman, an early shareware pioneer.

Shareware

In contrast with freeware, a *shareware* program includes a legalistic or pseudo-legalistic request or demand for payment. The user who sends in a payment is said to have

registered their copy of the game. Game authors can actually make a living doing this, but registration rates are typically very low. Registration is encouraged by offering upgrades or full printed documentation (see also *nagware*, *crippleware*, and *heroinware* below).

The shareware user is encouraged to share the unregistered version of the game with friends, thus building up a loyal customer base without the expensive overhead of advertising, packaging, or negotiating legal contracts with distributors. The downside is that many shareware authors report that registration rates are low. Reported figures range from 1% to 80%, but there is typically little evidence on which to base these conjectures.

Nagware

Nagware is a version of shareware that encourages registration by popping up a nag screen that reminds the user to register when they first start up the game. Typical approaches include locking the computer for a small period of time (typically 5 to 30 seconds), or by hiding the button or key-sequence that the user needs to make the nag screen go away, with the intention of making the user read or scan the whole screen in order to find it. The downside is that nag screens are either so innocuous that users ignore them, or so obnoxious that users remove the game from their hard drives. The middle ground is very hard to find.

Crippleware

Crippleware is a version of shareware in which the program is crippled in some way, usually by disabling play features, or by preventing the user from saving a game. The consumer must register the game in order to get a fully functional copy. The consensus of opinion among game players is that crippleware is so annoying that they never play more than once, and never register the copies.

Heroinware

Heroinware is a new variant of shareware pioneered by *id Software* with their game *Doom*. Unlike crippleware, the executable code for the entire unaltered game is distributed electronically, with enough levels to allow the user to play for an extended period of time. The user gets the remaining levels by registering the game. The rationale is that users play for long enough to get hooked on the game, and purchase the remaining play levels. Where crippleware is designed to prevent the user from enjoying the game without paying for it, *heroinware* encourages the user to enjoy the free version, in the hope that he or she will want more. *id Software* have reported that 127,000 registered copies of *Doom* have been sold, and that approximately 10 million unregistered copies exist.

In yet another marketing coup, *id Software* have released *Doom 2* as a full retail program without using shareware. An estimated 600,000 copies were ordered before the

<code>cs.columbia.edu</code>	game FAQs
<code>fau143.informatik.uni-erlangen.de</code>	Unix games
<code>ftp.cica.indiana.edu</code>	microsoft windows games
<code>ftp.uml.edu</code>	DOS games
<code>ftp.uwp.edu</code>	id archives; some development tools
<code>infant2.sphs.indiana.edu</code>	doom
<code>nic.funet.fi</code>	Amiga, DOS and Unix games
<code>rtfm.mit.edu</code>	game FAQs
<code>sumex-aim.stanford.edu</code>	Macintosh games
<code>wuarchive.wustl.edu</code>	dos, general
<code>x2ftp.oulu.fi</code>	the best general site

Table 1: Some game ftp sites.

<code>http://wcl-rs.bham.ac.uk/GamesDomain</code>	the best place to start
<code>http://www.cm.cf.ac.uk:/Fun/</code>	pointers to games (esp. unix)
<code>http://www.fokus.gmd.de/minos/employees/hgs/audio/audio.html</code>	PC audio hardware
<code>http://obsidian.math.arizona.edu:8080/netrek.html</code>	Netrek, net-based Star Trek
<code>http://hercule.csci.unt.edu/larc</code>	UNT LARC

Table 2: Game-related URLs.

game was released, generating over \$41 million in revenue at the \$69 per copy list price. It may be hypothesized that this incredible success (in the games industry, 200,000 copies is considered a blockbuster) is due to the large customer base built up through the use of heroinware.

5 Computer Games at UNT

5.1 The LARC Project

The author of this paper established the Laboratory for Recreational Computing at UNT in 1993. Membership in LARC is open to undergraduate and graduate students at UNT, and is on a voluntary basis. Current membership is almost exclusively undergraduate, and includes computer scientists, artists, and musicians. The group meets formally for two hours once a week. UNT has provided laboratory space and five computers with sound cards and joysticks. LARC members have keys to the laboratory and have exclusive access to the equipment at all times. Membership figures are shown in Figure 1. More information is available on LARC on the World-Wide Web [10].

5.2 A Computer Games Class

A senior level undergraduate course on Game Design and Programming was offered at UNT in Fall 1994 under an experimental CSCI 4980 course code. Preliminary announcements on the Internet in `unt.general`, `rec.games.programmer`, and `dfw.general` attracted 90 queries for information, and 24 students were enrolled, (8 of them already LARC members). The course was taught

without a textbook, relying heavily on resources found on the Internet. More information about CSCI 4980 is available on the World-Wide Web [9]. It will be offered again in Fall 1995 under the CSCI 4330 course code.

Material covered in the course fell into two categories, nontechnical and technical. The nontechnical material included:

- game genres
- marketing and copyright
- introduction to the internet
- choosing a name
- sex, violence, and political correctness
- what makes a successful game
- game reviews
- storyboarding
- the game proposal
- team composition

The technical material included:

- introduction to 32-bit programming
- graphics file formats
- offscreen buffers and blitters
- sprite animation
- palette manipulation
- 2d graphics
- input devices: keyboard, mouse, and joystick
- 32-bit protected mode assembly programming
- the physics of sound sampling
- sound effects using VOC files and DMA
- general MIDI
- 3d graphics techniques
- code optimization

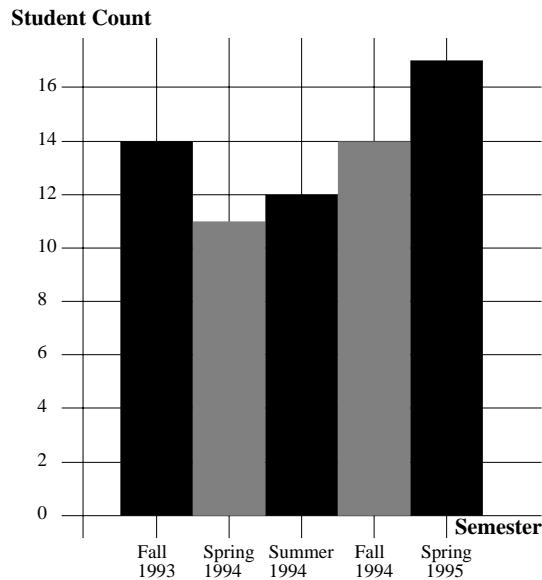


Figure 1: Student membership of LARC. Figures for Spring 1995 are anticipated from a poll of students.

- hooking to timers
- mode X graphics

Grading was based on two simple programming assignments, and the major team project. The first program was to add features to a simple sprite-based game engine, and the second was to add sound effects to the same program. The team project required the design and implementation of a prototype for a new computer game. Students worked in teams of one to three members to submit a proposal, a storyboard, a progress report, and the final program.

5.3 Feedback

LARC has been in operation for long enough that feedback from students, administrators, and faculty can be evaluated. Each of these will be discussed in a separate subsection. Faculty planning to duplicate LARC should be aware that although students will be in favor of it, faculty and administrators may be unwilling support it.

5.3.1 Student Feedback

Student reaction to LARC has been exclusively positive, even from students who have no interest in computer games, and those who have an interest in computer games but choose not to participate. Student reaction to the CSCI 4980 class offered in Fall 94 was enthusiastic, including statement such as the following (an unedited transcript of anonymous student comments is available on the World-Wide Web [11].)

“The Entertainment Industry has long been a closed market to those who follow the educational route career, one this class succeeds in opening up these doors, and giving insight into one of the fastest growing software markets.”

“This was an excellent and informative class.”

“This class (and the chance it provides) are, without a doubt, absolutely necessary.”

“This is one of the most interesting and practical classes that I have ever taken.”

“I have already learned more in this class than I would have expected to learn in any class I had taken.”

“This is by far the most informative computer science course (or any other type of course) that I have taken anywhere.”

“Actually, though I had researched this topic for some time before taking this class, my eyes were opened to the world of techniques, tools, and methods used to create software in this field (games programming). This class has armed me with information extremely well, and pointed out where to look for more.”

5.3.2 Reaction from Administrators

Reaction from administrators at UNT, from department chairs to Deans to the Chancellor, has also been very positive and encouraging. The Computer Science department has been generous in providing laboratory space and funds for an initial purchase of computer equipment. Higher administrators have been supportive of the idea of using LARC for promotional purposes, which benefits LARC directly by attracting more students.

5.3.3 Reaction from Faculty

Reaction from faculty has been more guarded. Curiously, faculty outside the Computer Sciences Department have been more supportive of LARC than those inside it. The major criticism from faculty in Computer Sciences is that games are not appropriate to the Computer Science curriculum, which as we argued in Section 2, is not true. Games programming is an expanding part of the employment market, and can be integrated into a traditional Computer Science curriculum as a capstone course that integrates knowledge gained at all levels.

References

- [1] The PC Games Programmers Encyclopedia. Version 1.0, available by anonymous ftp from `teeri.oulu.fi` in `/pub/msdos/programming/gpe`, 1994.
- [2] D. Gruber. *Action Arcade Adventure Set*. Coriolis Group Books, 1994.
- [3] B. Hook. *C++ Game Programmer's Secrets: What the Game Companies are Afraid You'll Find Out*. Sams Publishing, 1994.

- [4] C. Howard. *Programming Games for Beginners: Visual Basic for Windows for Fun and for Profit*. Sams Publishing, 1993.
- [5] A. LaMothe. *Teach Yourself Games Programming in 21 Days*. Sams Publishing, 1994.
- [6] A. LaMothe, J. Ratcliff, M. Seminatore, and D. Tyler. *Tricks of the Game-Programming Gurus*. Sams Publishing, 1994.
- [7] C. Lampton. *Flights of Fantasy: Programming 3-D Video Games in C++*. The Waite Group Press, 1993.
- [8] C. Lampton. *Gardens of Imagination: Programming 3D Maze Games in C/C++*. The Waite Group Press, 1994.
- [9] I. Parberry. CSCI 4980: Computer Game Design and Programming. A WWW document with URL <http://hercule.csci.unt.edu/larc/4980.94f>, 1994.
- [10] I. Parberry. The Laboratory for Recreational Computing. A WWW document with URL <http://hercule.csci.unt.edu/larc>, 1994.
- [11] I. Parberry. Student Comments from CSCI 4980, Fall 1994. A WWW document with URL <http://hercule.csci.unt.edu/larc/4980.94f/feedback.html>, 1994.
- [12] D. Roberts. *Easy PC Game Programming*. Coriolis Group Books, 1994.