Understanding Best Practices in Free/Open Source Software Development

Walt Scacchi
Institute for Software Research
School of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
Wscacchi@ics.uci.edu

18 October 2003

http://www.ics.uci.edu/~wscacchi/Presentations/COSST/BestPractices.pdf

What is free/open source software development?

- Free (as in “freedom”) vs. open source
  - Freedom to access, browse/view, study, modify and redistribute the source code
  - Free is always open, but open is not always free
- F/OSSD is not “software engineering”
  - *Different*: F/OSSD can be faster, better, and cheaper than SE
- F/OSSD involves *more* software development tools, Web resources, and personal computing resources

Who is investing in OSSD?

- Large corporations: (IT and Financial)
  - IBM-*Eclipse*, Sun-*NetBeans* and *OpenOffice*, HP-*Gelato*, Apple-*Darwin*, Microsoft Research-*Rotor*, SAP-*DB*, etc.
  - Barclays Global Investors, DKW
- Mid-size corporations:
  - RedHat, Novell
- Small (start-up) companies:
  - ActiveState, Collab.Net, Jabber, Ximian, JBoss, Compiere, etc.

Sample practices for F/OSSD

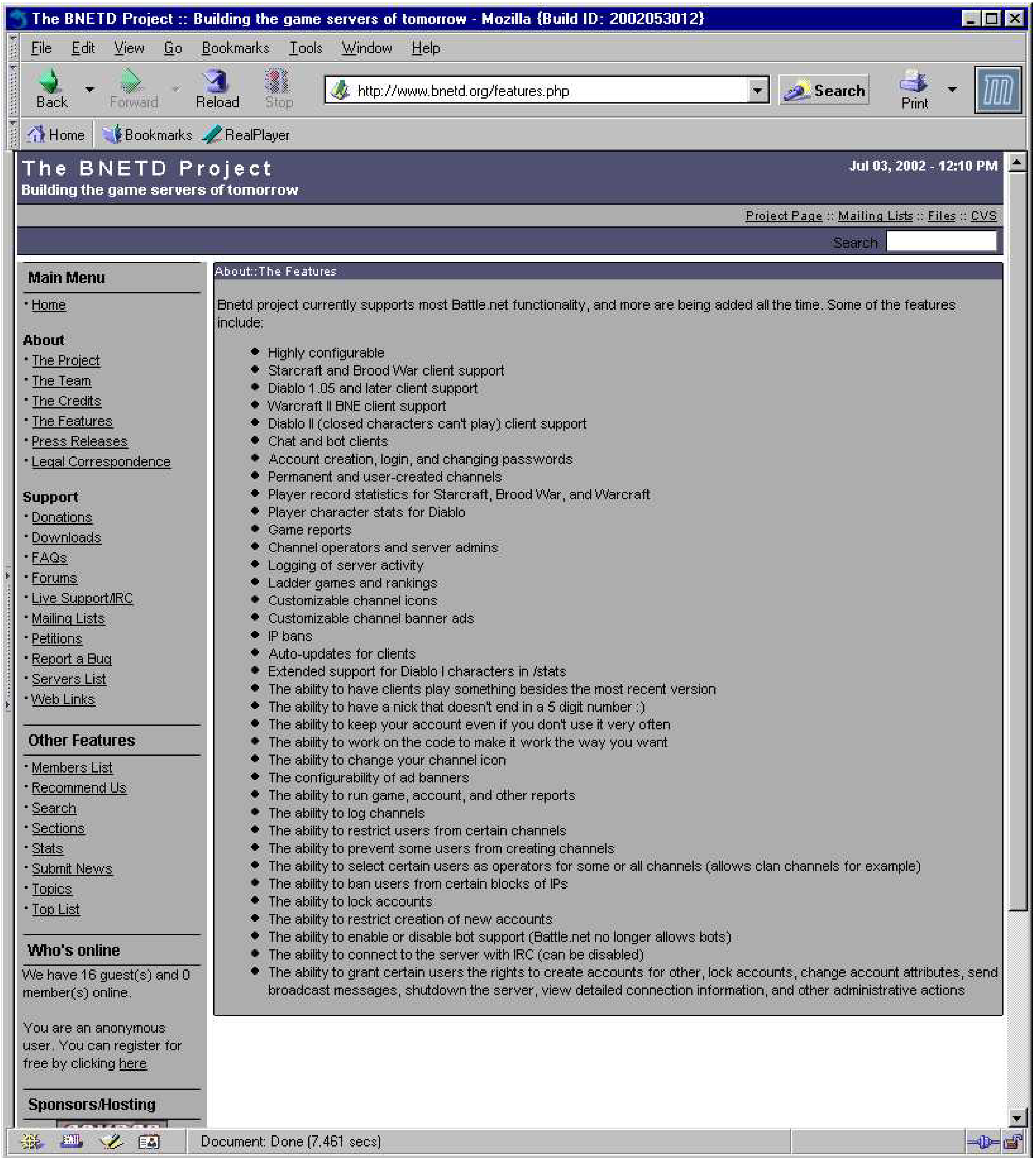
- Requirements and design
- Configuration management and work coordination
- Maintenance/Evolution
- Project management/career development
- Software technology transfer and licensing

F/OSS Processes for Requirements or Design

- F/OSS Requirements/Designs
  - not explicit
  - not formal
- F/OSS Requirements/Designs are embedded within "informalisms"
  - Example OSS informalisms to follow (as screenshot displays)
- F/OSS Requirements/Design processes are different from their SE counterparts.

SE vs. F/OSS processes for Requirements

• Elicitation	• *Post-hoc* assertion
• Analysis	• Reading, sense-making, accountability
• Specification and modeling	• Continually emerging webs of discourse
• Validation	• Condensing and hardening discourse
• Communicating and managing	• Global access to discourse


The BNETD Project :: Building the game servers of tomorrow - Mozilla {Build ID: 2002053012}
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://www.bnetd.org/features.php
Search
Print
Home Bookmarks RealPlayer
Document: Done (7.461 secs)

The BNETD Project

Building the game servers of tomorrow

Jul 03, 2002 - 12:10 PM

Project Page :: Mailing Lists :: Files :: CVS

Search

Main Menu

- Home

About

- The Project
- The Team
- The Credits
- The Features
- Press Releases
- Legal Correspondence

Support

- Donations
- Downloads
- FAQs
- Forums
- Live Support/IRC
- Mailing Lists
- Petitions
- Report a Bug
- Servers List
- Web Links

Other Features

- Members List
- Recommend Us
- Search
- Sections
- Stats
- Submit News
- Topics
- Top List

Who's online

We have 16 guest(s) and 0 member(s) online.

You are an anonymous user. You can register for free by clicking here

Sponsors/Hosting

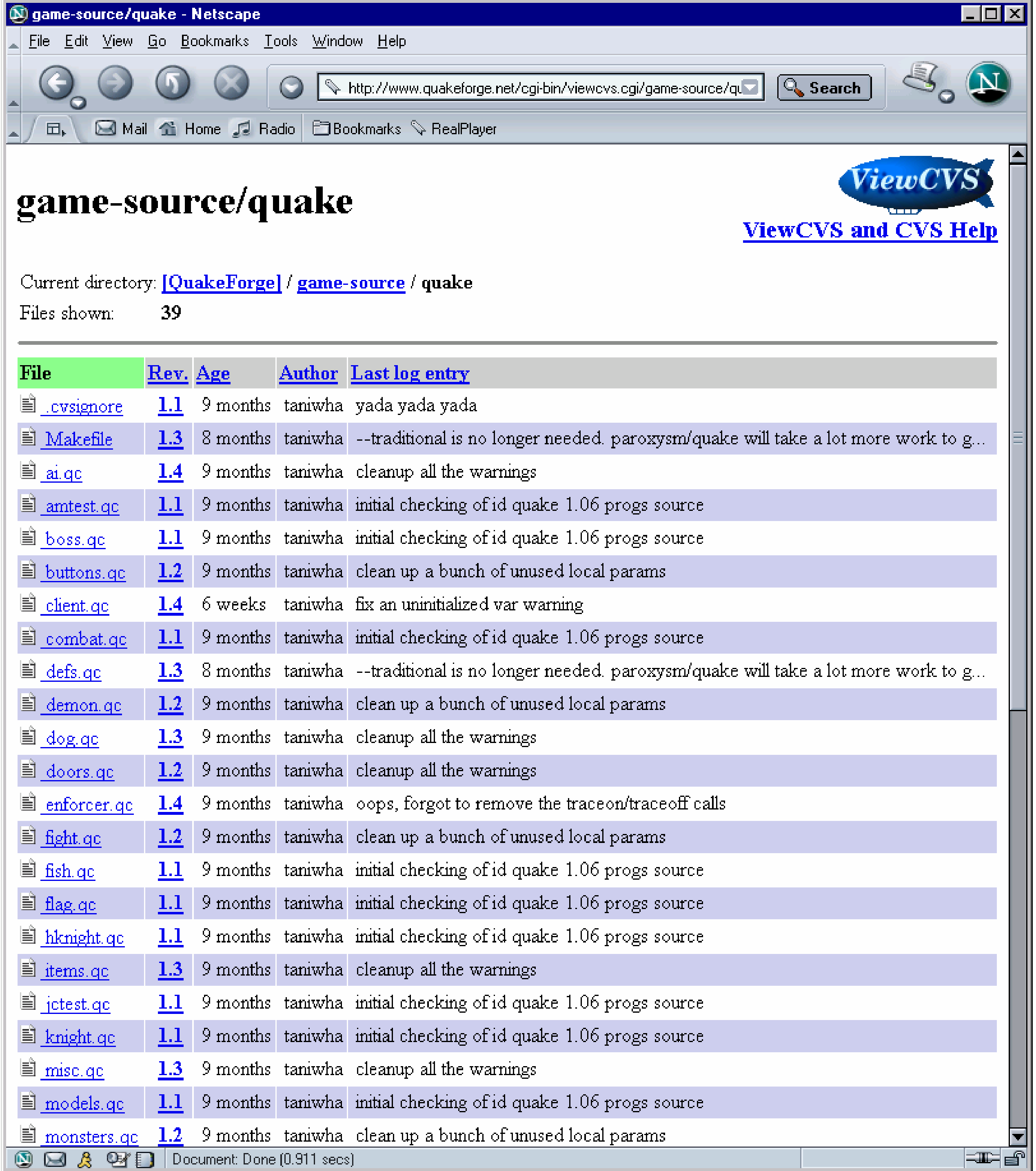
About::The Features

Bnetd project currently supports most Battle.net functionality, and more are being added all the time. Some of the features include:

- Highly configurable
- Starcraft and Brood War client support
- Diablo 1.05 and later client support
- Warcraft II BNE client support
- Diablo II (closed characters can't play) client support
- Chat and bot clients
- Account creation, login, and changing passwords
- Permanent and user-created channels
- Player record statistics for Starcraft, Brood War, and Warcraft
- Player character stats for Diablo
- Game reports
- Channel operators and server admins
- Logging of server activity
- Ladder games and rankings
- Customizable channel icons
- Customizable channel banner ads
- IP bans
- Auto-updates for clients
- Extended support for Diablo I characters in /stats
- The ability to have clients play something besides the most recent version
- The ability to have a nick that doesn't end in a 5 digit number :)
- The ability to keep your account even if you don't use it very often
- The ability to work on the code to make it work the way you want
- The ability to change your channel icon
- The configurability of ad banners
- The ability to run game, account, and other reports
- The ability to log channels
- The ability to restrict users from certain channels
- The ability to prevent some users from creating channels
- The ability to select certain users as operators for some or all channels (allows clan channels for example)
- The ability to ban users from certain blocks of IPs
- The ability to lock accounts
- The ability to restrict creation of new accounts
- The ability to enable or disable bot support (Battle.net no longer allows bots)
- The ability to connect to the server with IRC (can be disabled)
- The ability to grant certain users the rights to create accounts for other, lock accounts, change account attributes, send broadcast messages, shutdown the server, view detailed connection information, and other administrative actions

Configuration management and work coordination

- Use CM to coordinate and control who gets to update what part of the system
  - Many F/OSSD projects use **CVS** (single centralized code repository with update locks) and frequent releases (*daily releases* on active projects)
  - Linux Kernel: **BitKeeper** (multiple parallel builds and release repositories)
  - Collab.Net and Tigris.org: **Subversion** (CVS++)
  - Apache: Single major release, with frequent "*patch*" releases (e.g., "A patchy server")


game-source/quake - Netscape
File Edit View Go Bookmarks Tools Window Help
http://www.quakeforge.net/cgi-bin/viewcvs.cgi/game-source/qu
Search
Mail Home Radio Bookmarks RealPlayer
ViewCVS
Document: Done (0.911 secs)

game-source/quake

ViewCVS and CVS Help

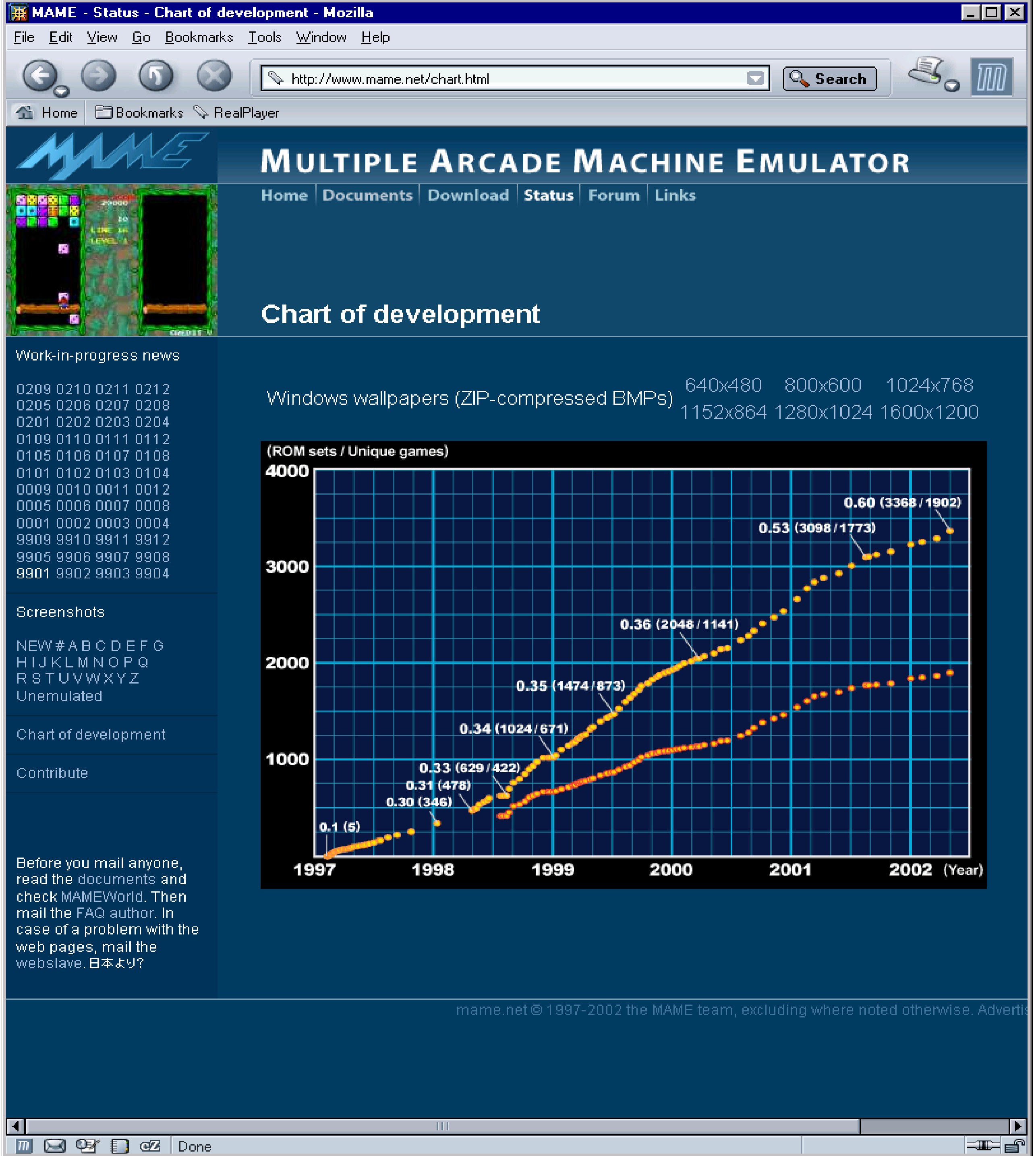
Current directory: [QuakeForge] / game-source / quake

Files shown: 39

File	Rev.	Age	Author	Last log entry
.cvsignore	1.1	9 months	taniwha	yada yada yada
Makefile	1.3	8 months	taniwha	--traditional is no longer needed. paroxysm/quake will take a lot more work to g...
ai.qc	1.4	9 months	taniwha	cleanup all the warnings
amtest.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
boss.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
buttons.qc	1.2	9 months	taniwha	clean up a bunch of unused local params
client.qc	1.4	6 weeks	taniwha	fix an uninitialized var warning
combat.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
defs.qc	1.3	8 months	taniwha	--traditional is no longer needed. paroxysm/quake will take a lot more work to g...
demon.qc	1.2	9 months	taniwha	clean up a bunch of unused local params
dog.qc	1.3	9 months	taniwha	cleanup all the warnings
doors.qc	1.2	9 months	taniwha	cleanup all the warnings
enforcer.qc	1.4	9 months	taniwha	oops, forgot to remove the traceon/traceoff calls
fight.qc	1.2	9 months	taniwha	clean up a bunch of unused local params
fish.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
flag.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
hknight.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
items.qc	1.3	9 months	taniwha	cleanup all the warnings
jctest.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
knight.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
misc.qc	1.3	9 months	taniwha	cleanup all the warnings
models.qc	1.1	9 months	taniwha	initial checking of id quake 1.06 progs source
monsters.qc	1.2	9 months	taniwha	clean up a bunch of unused local params

Evolutionary redevelopment, reinvention, and redistribution

- Overall evolutionary dynamic of F/OSSD is *reinvention*
  - Reinvention enables continuous improvement
- F/OSS evolve through minor mutations
  - Expressed, recombined, redistributed via incremental releases
- F/OSS systems *co-evolve* with their development community
  - Success of one depends on the success of the other
- Closed legacy systems may be *revitalized* via opening and redistribution of their source
  - When enthusiastic user-developers want their cultural experience with such systems to be maintained.



MULTIPLE ARCADE MACHINE EMULATOR

Home | Documents | Download | Status | Forum | Links

Chart of development

Work-in-progress news

0209 0210 0211 0212
0205 0206 0207 0208
0201 0202 0203 0204
0109 0110 0111 0112
0105 0106 0107 0108
0101 0102 0103 0104
0009 0010 0011 0012
0005 0006 0007 0008
0001 0002 0003 0004
9909 9910 9911 9912
9905 9906 9907 9908
9901 9902 9903 9904

Screenshots

NEW # A B C D E F G
H I J K L M N O P Q
R S T U V W X Y Z
Unemulated

Chart of development

Contribute

Before you mail anyone, read the documents and check MAMEWorld. Then mail the FAQ author. In case of a problem with the web pages, mail the webslave. 日本より?

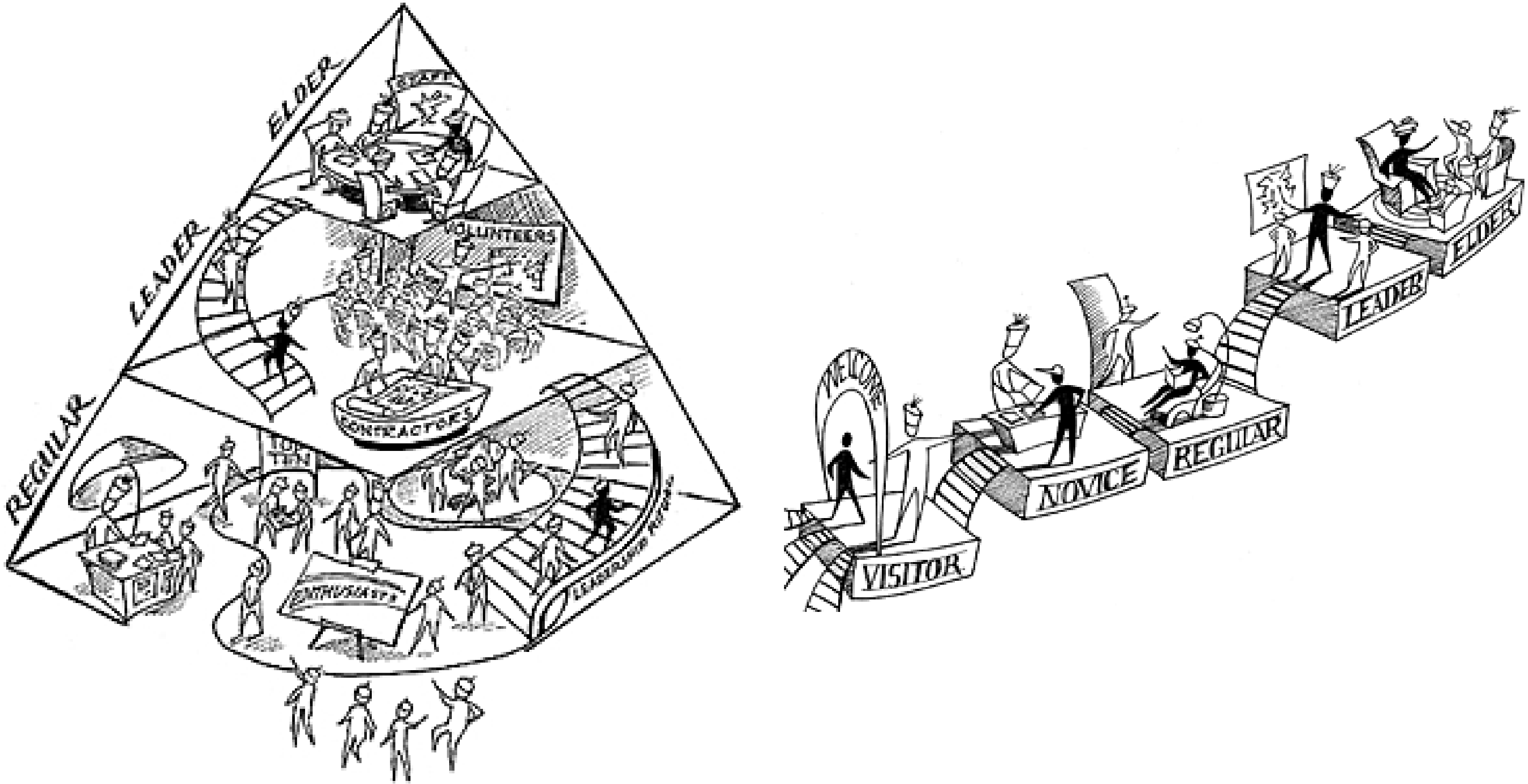
Windows wallpapers (ZIP-compressed BMPs) 640x480 800x600 1024x768 1152x864 1280x1024 1600x1200

mame.net © 1997-2002 the MAME team, excluding where noted otherwise. Advertis

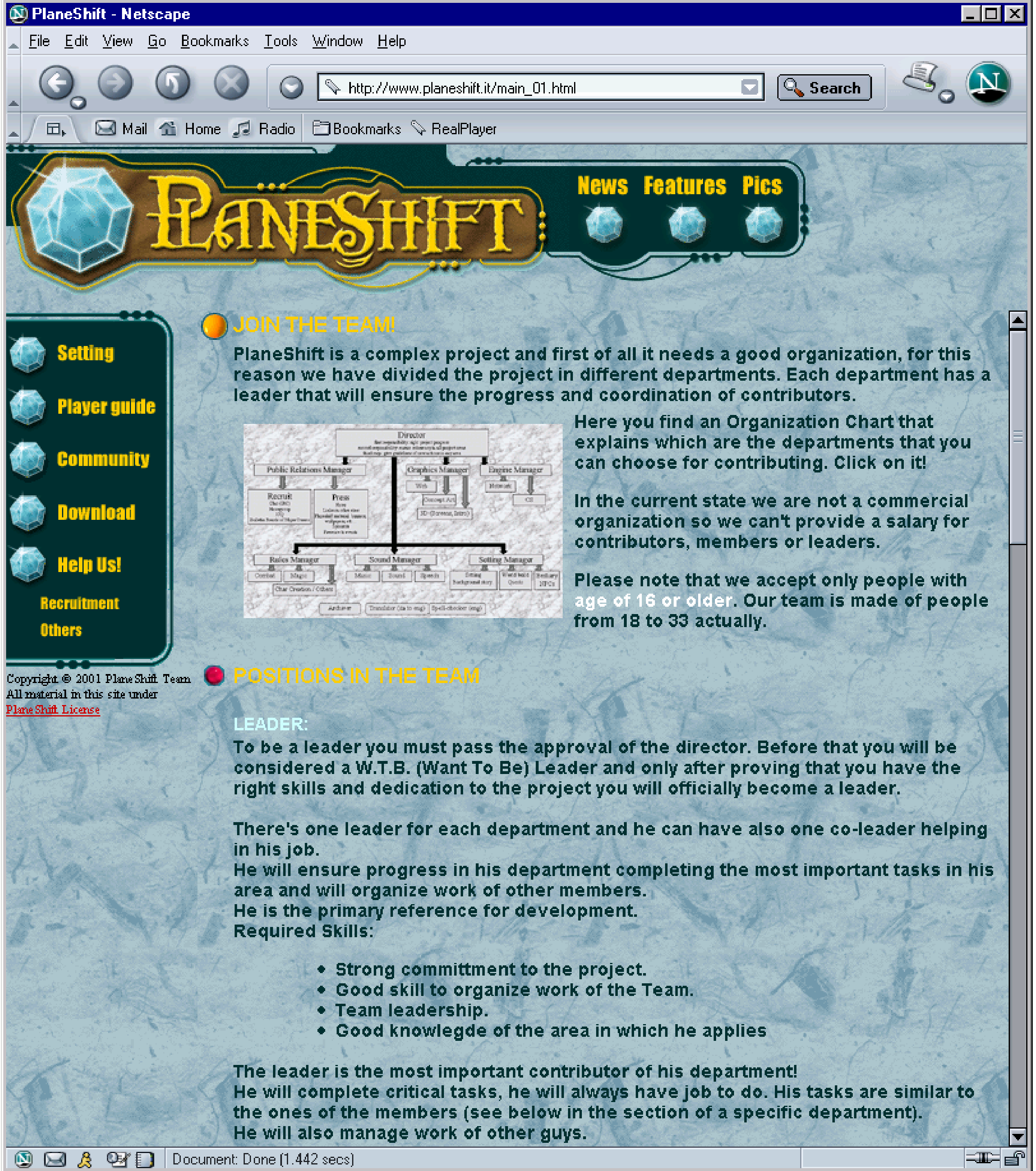
Project management and career development

- F/OSSD projects self-organize as a *pyramid meritocracy* via *virtual project management*
  - Meritocracies embrace incremental mutations over radical innovations
  - VPM requires people to act in leadership roles based on skill, availability, and belief in project community
- F/OSS developers want to have fun, exercise their technical skill, try out new kinds of systems to develop, and/or interconnect multiple F/OSSD projects (*freedom of choice and expression*).

A pyramid meritocracy and role hierarchy for F/OSSD



(images from A.J. Kim, *Community Building on the Web*, 2000)


PlaneShift - Netscape
File Edit View Go Bookmarks Tools Window Help
http://www.planeshift.it/main_01.html
Search
Mail Home Radio Bookmarks RealPlayer
PLANESHIFT
News Features Pics
Setting
Player guide
Community
Download
Help Us!
Recruitment
Others
Document: Done (1.442 secs)

Copyright © 2001 Plane Shift Team
All material in this site under
Plane Shift License

JOIN THE TEAM!

PlaneShift is a complex project and first of all it needs a good organization, for this reason we have divided the project in different departments. Each department has a leader that will ensure the progress and coordination of contributors.

Here you find an Organization Chart that explains which are the departments that you can choose for contributing. Click on it!

In the current state we are not a commercial organization so we can't provide a salary for contributors, members or leaders.

Please note that we accept only people with age of 16 or older. Our team is made of people from 18 to 33 actually.

POSITIONS IN THE TEAM

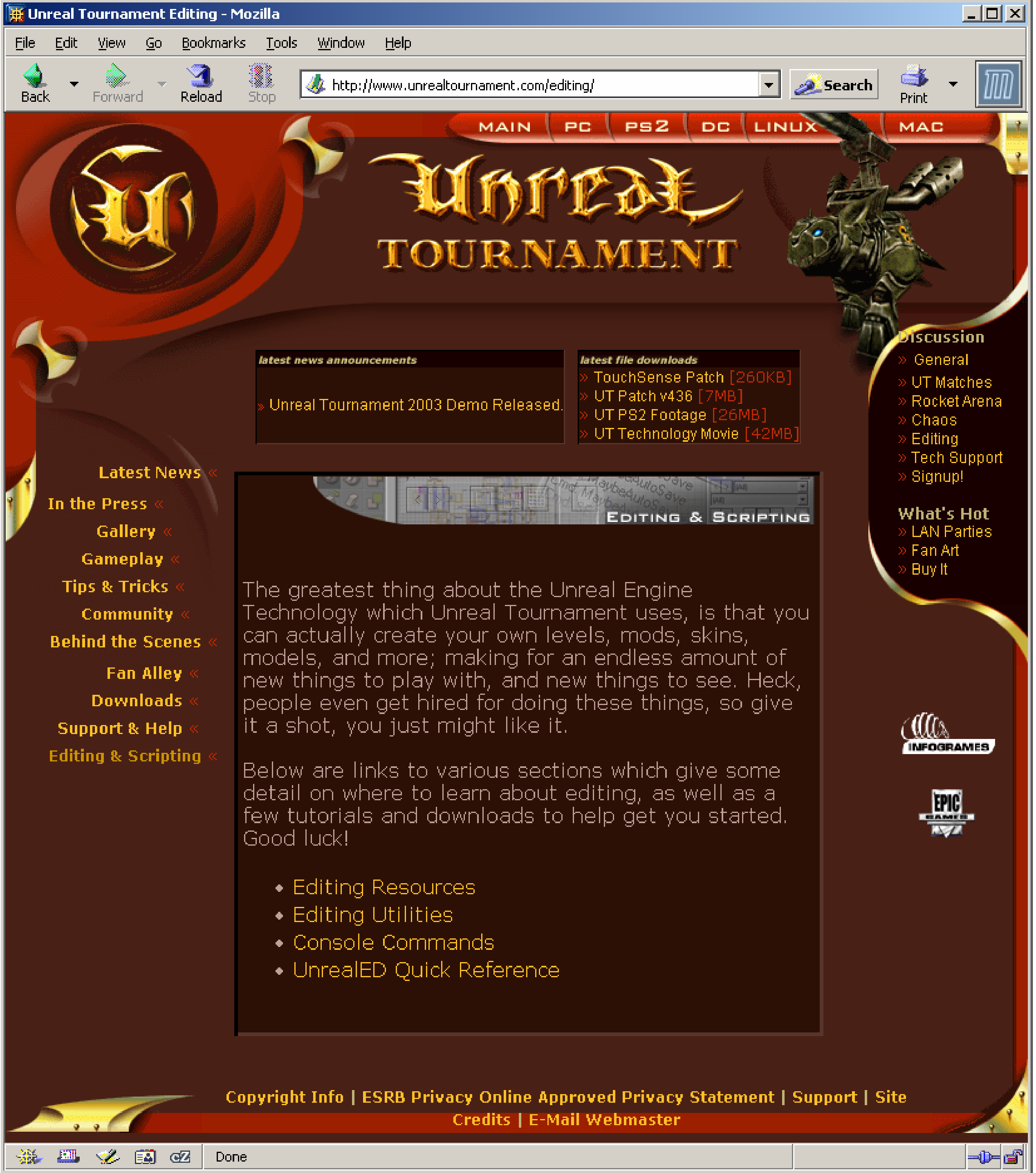
LEADER:

To be a leader you must pass the approval of the director. Before that you will be considered a W.T.B. (Want To Be) Leader and only after proving that you have the right skills and dedication to the project you will officially become a leader.

There's one leader for each department and he can have also one co-leader helping in his job.
He will ensure progress in his department completing the most important tasks in his area and will organize work of other members.
He is the primary reference for development.
Required Skills:

- Strong committment to the project.
- Good skill to organize work of the Team.
- Team leadership.
- Good knowlegde of the area in which he applies

The leader is the most important contributor of his department!
He will complete critical tasks, he will always have job to do. His tasks are similar to the ones of the members (see below in the section of a specific department).
He will also manage work of other guys.



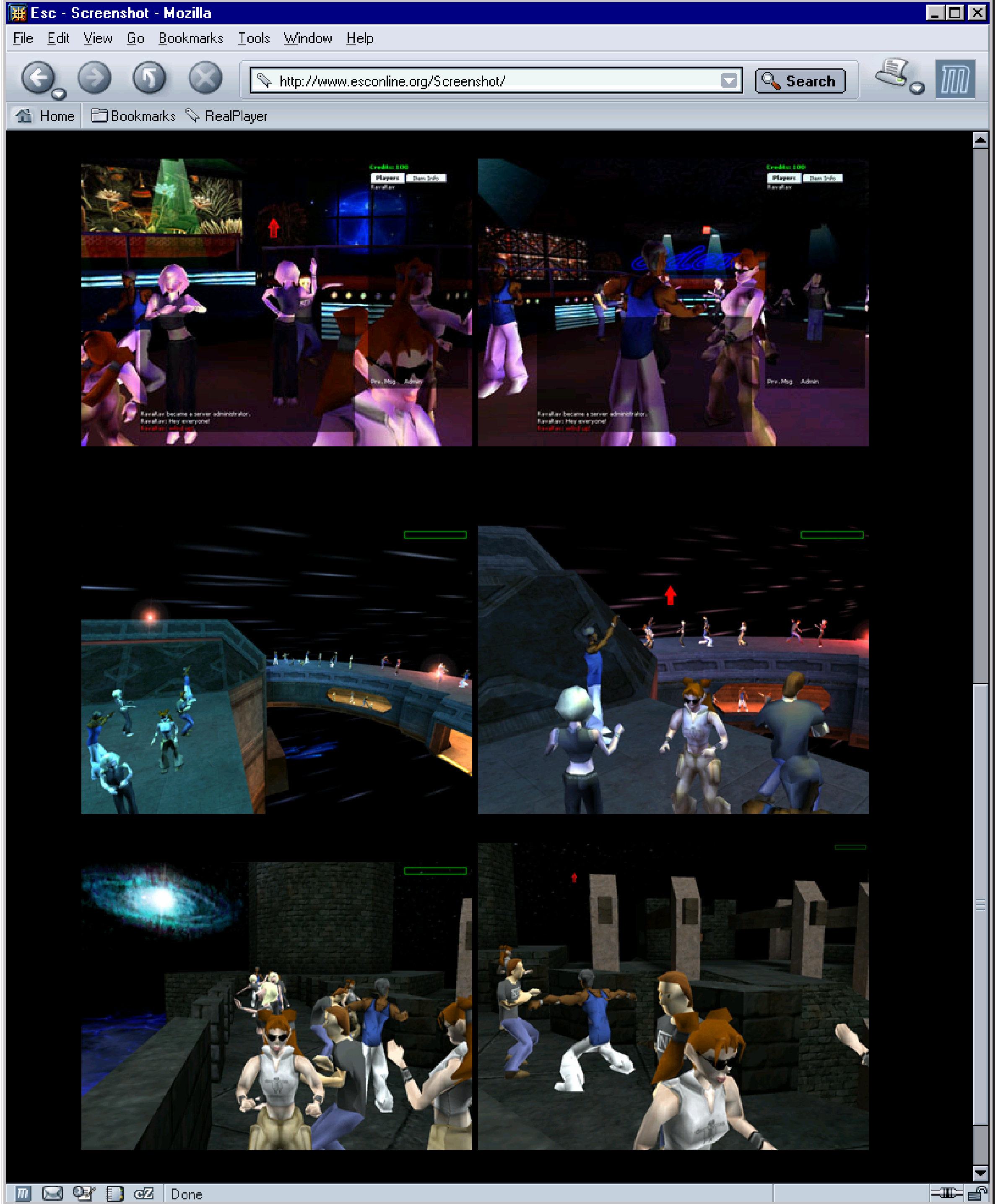
The greatest thing about the Unreal Engine Technology which Unreal Tournament uses, is that you can actually create your own levels, mods, skins, models, and more; making for an endless amount of new things to play with, and new things to see. Heck, people even get hired for doing these things, so give it a shot, you just might like it.

Below are links to various sections which give some detail on where to learn about editing, as well as a few tutorials and downloads to help get you started. Good luck!

- Editing Resources
- Editing Utilities
- Console Commands
- UnrealED Quick Reference

Software technology transfer and licensing

- F/OSS technology transfer from existing Web sites is a *community and team building process*
  - Not (yet) an engineering process
  - Enables unanticipated applications and uses
  - Enables F/OSSD to persist *without* centrally planned and managed corporate software development centers



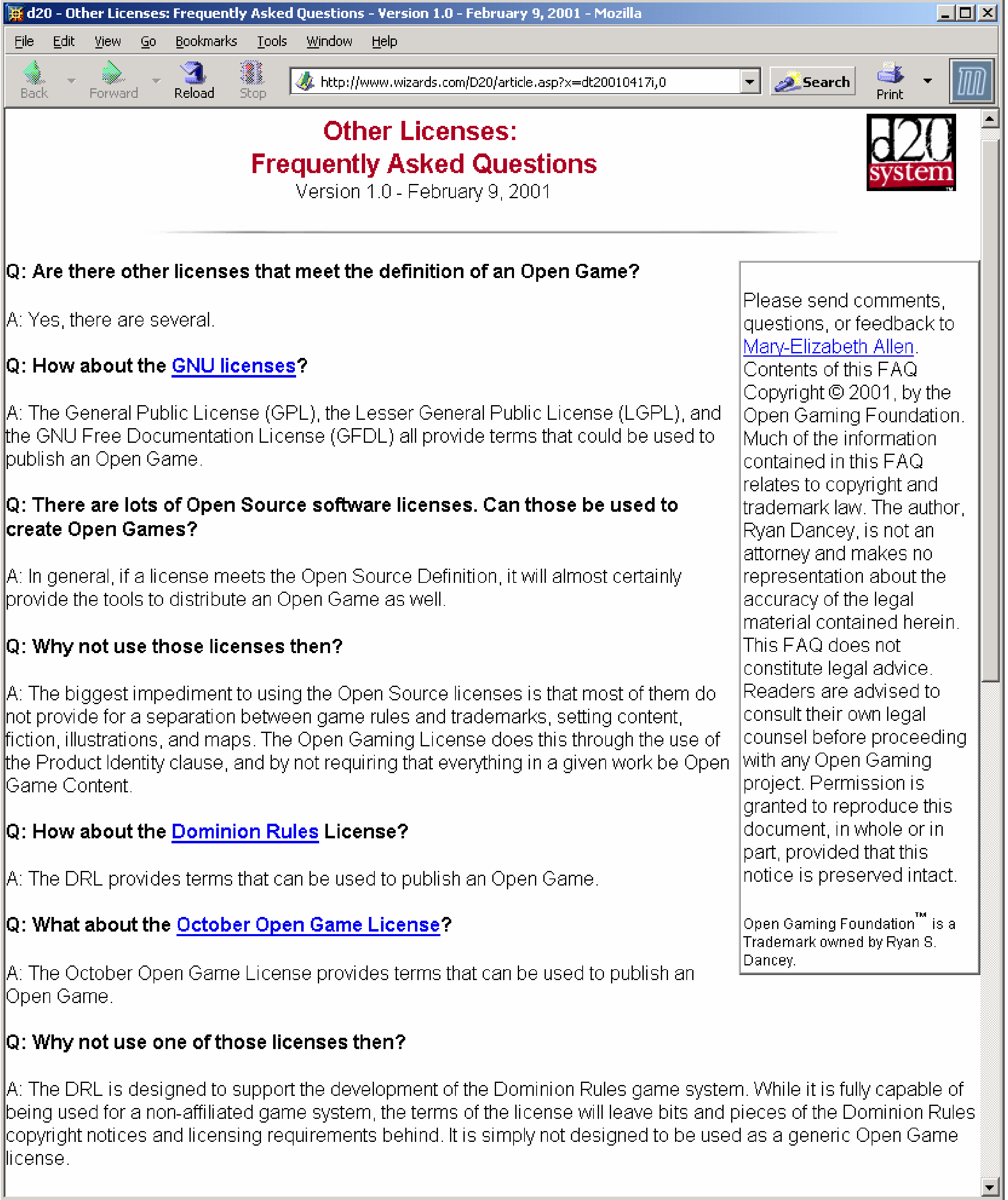
RavaRav became a server administrator.
RavaRav: Hey everyone!

RavaRav became a server administrator.
RavaRav: Hey everyone!

Free/OSS licenses

Reiterate and institutionalize F/OSS *culture* (values, norms, and beliefs)

- GNU Public License (GPL) for *free* software
- More than 35 other open source licenses (http://opensource.org)
- “Creative Commons” Project at Stanford Law School developing public license framework


d20 - Other Licenses: Frequently Asked Questions - Version 1.0 - February 9, 2001 - Mozilla
File
Edit
View
Go
Bookmarks
Tools
Window
Help
Back
Forward
Reload
Stop
http://www.wizards.com/D20/article.asp?x=dt20010417i,0
Search
Print
d20 system

Other Licenses: Frequently Asked Questions

Version 1.0 - February 9, 2001

Q: Are there other licenses that meet the definition of an Open Game?

A: Yes, there are several.

Q: How about the GNU licenses?

A: The General Public License (GPL), the Lesser General Public License (LGPL), and the GNU Free Documentation License (GFDL) all provide terms that could be used to publish an Open Game.

Q: There are lots of Open Source software licenses. Can those be used to create Open Games?

A: In general, if a license meets the Open Source Definition, it will almost certainly provide the tools to distribute an Open Game as well.

Q: Why not use those licenses then?

A: The biggest impediment to using the Open Source licenses is that most of them do not provide for a separation between game rules and trademarks, setting content, fiction, illustrations, and maps. The Open Gaming License does this through the use of the Product Identity clause, and by not requiring that everything in a given work be Open Game Content.

Q: How about the Dominion Rules License?

A: The DRL provides terms that can be used to publish an Open Game.

Q: What about the October Open Game License?

A: The October Open Game License provides terms that can be used to publish an Open Game.

Q: Why not use one of those licenses then?

A: The DRL is designed to support the development of the Dominion Rules game system. While it is fully capable of being used for a non-affiliated game system, the terms of the license will leave bits and pieces of the Dominion Rules copyright notices and licensing requirements behind. It is simply not designed to be used as a generic Open Game license.

Please send comments, questions, or feedback to Mary-Elizabeth Allen. Contents of this FAQ Copyright © 2001, by the Open Gaming Foundation. Much of the information contained in this FAQ relates to copyright and trademark law. The author, Ryan Dancey, is not an attorney and makes no representation about the accuracy of the legal material contained herein. This FAQ does not constitute legal advice. Readers are advised to consult their own legal counsel before proceeding with any Open Gaming project. Permission is granted to reproduce this document, in whole or in part, provided that this notice is preserved intact.

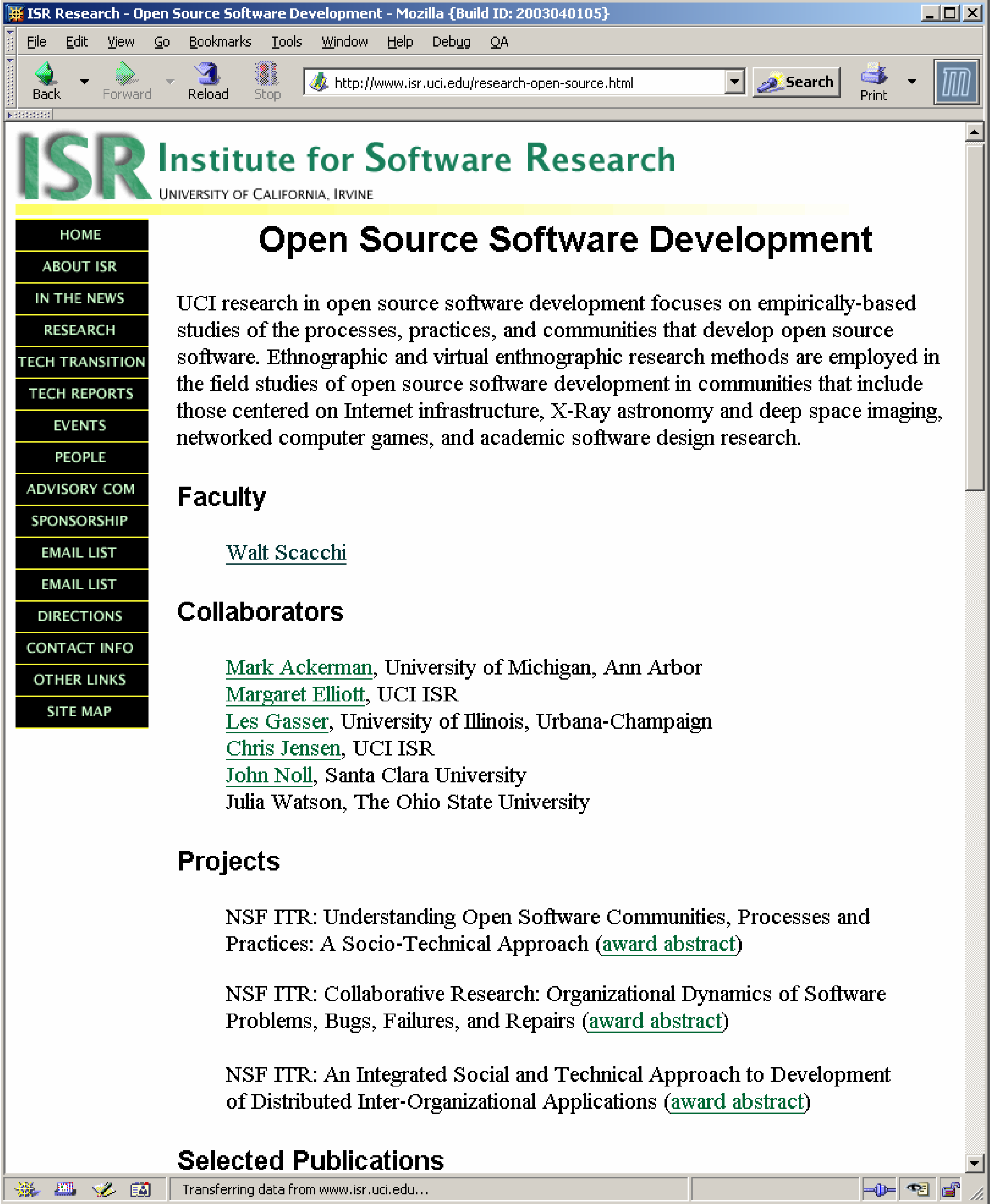
Open Gaming Foundation™ is a Trademark owned by Ryan S. Dancey.

Implications

- F/OSSD is a *community building process*
  - – not just a technical development process
  - – F/OSS peer review creates a *community of peers*
- F/OSSD processes often iterate *daily* versus infrequent singular (milestone) Software Life Cycle Engineering events
  - – F/OSSD: frequent, rapid cycle time (easier to improve) *vs*.
  - – SLC: infrequent, slow cycle time (harder to improve)

Conclusions

- Developing F/OSS is *different* than software engineering
  - not better, not worse, but different and new
  - more social, more accessible, more convivial
- F/OSS systems don’t need and probably won’t benefit from classic software engineering.


ISR Research - Open Source Software Development - Mozilla {Build ID: 2003040105}
File
Edit
View
Go
Bookmarks
Tools
Window
Help
Debug
QA
Back
Forward
Reload
Stop
http://www.isr.uci.edu/research-open-source.html
Search
Print
ISR Institute for Software Research
University of California, Irvine
HOME
ABOUT ISR
IN THE NEWS
RESEARCH
TECH TRANSITION
TECH REPORTS
EVENTS
PEOPLE
ADVISORY COM
SPONSORSHIP
EMAIL LIST
EMAIL LIST
DIRECTIONS
CONTACT INFO
OTHER LINKS
SITE MAP
Transferring data from www.isr.uci.edu...

Open Source Software Development

UCI research in open source software development focuses on empirically-based studies of the processes, practices, and communities that develop open source software. Ethnographic and virtual enthnographic research methods are employed in the field studies of open source software development in communities that include those centered on Internet infrastructure, X-Ray astronomy and deep space imaging, networked computer games, and academic software design research.

Faculty

Walt Scacchi

Collaborators

Mark Ackerman, University of Michigan, Ann Arbor
Margaret Elliott, UCI ISR
Les Gasser, University of Illinois, Urbana-Champaign
Chris Jensen, UCI ISR
John Noll, Santa Clara University
Julia Watson, The Ohio State University

Projects

NSF ITR: Understanding Open Software Communities, Processes and Practices: A Socio-Technical Approach (award abstract)

NSF ITR: Collaborative Research: Organizational Dynamics of Software Problems, Bugs, Failures, and Repairs (award abstract)

NSF ITR: An Integrated Social and Technical Approach to Development of Distributed Inter-Organizational Applications (award abstract)

Selected Publications

Acknowledgements

- *Project collaborators*:
  - Mark Ackerman, UMichigan, Ann Arbor
  - Les Gasser, UIllinois, Urbana-Champaign
  - John Noll, Santa Clara University
  - Margaret Ellliot, Chris Jensen, UCI-ISR
  - Julia Watson, The Ohio State University
- *Funding support*:
  - National Science Foundation, IIS#-0083075, ITR#-#0205679, ITR#-0205724, and IIS#-#0350754.
  - No endorsement implied.

References

see http://www.ics.uci.edu/~wscacchi

- W. Scacchi, Understanding the Requirements for Developing Open Source Software, *IEE Proceedings--Software*, 149(1), 24-39, 2002.

- W. Scacchi, *Open EC/B*: A Case Study in Electronic Commerce and Open Source Software Development, Final Report, July 2002.

- W. Scacchi, Free/Open Source Software Development Practices in the Computer Game Community, *IEEE Software*, Special Issue on Open Source Software, (to appear, 2004).

- W. Scacchi, Understanding Free/Open Source Software Evolution: Applying, Breaking and Rethinking the Laws of Software Evolution, revised version to appear in N.H. Madhavji, M.M. Lehman, J.F. Ramil and D. Perry (eds.), *Software Evolution*, John Wiley and Sons Inc, New York, 2004.