Quake3 was the last id software engine to be written in C (the development team moved to C++ with Doom3).

Quake III codename Trinity does not come from the Matrix movie but rather from the "Trinity River in Dallas" (source: http://www.firingsquad.com/features/carmack/page12.asp)

Almost no usage of libraries: huffman.c md4.c etc...

```
$ cloc code common

http://cloc.sourceforge.net v 1.53  T=6.0 s (93.2 files/s, 59328.3 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C                              340          38573          60515         196318
C/C++ Header                   157           6594           7975          24670
Objective C                     12            886            744           3128
Perl                             4           1278           2954           2798
C++                              8            725            724           2438
make                             3            218            227           1791
Assembly                         7            219            282           1150
DOS Batch                       13             39             10            475
Bourne Shell                     8             61             48            305
HTML                             1              6              0            277
CSS                              1              0              0            207
XSLT                             1             28              2            202
XML                              1              0              0             95
Teamcenter def                   3              0              0              8
-------------------------------------------------------------------------------
SUM:                           559          48627          73481         233862
-------------------------------------------------------------------------------
```

- Twice the LOC of Quake2
- No asm optimized routines (since no software renderer).

Load Visual Studio 2008
Remove source control solution binding
Switch to release
Compiled:

Workspace projects:
===================

```
   - botlib     botlib\Release\botlib.lib
   - cgame      Release\cgamex86.dll
   - game       Release\qagamex86.dll
   - q3_ui      Release\uix86.dll
   - quake3     Release\quake3.exe
   - renderer   renderer\Release\renderer.lib
   - splines    splines\Release\Splines.lib
   - ui         Not building (would build: uix86_new.dll) this project is build via cpp.exe,lcc.exe and q3asm.exe instead
```

Note: The demo comes with quake3.exe and 3 bytecode packages in pak0.pk3: vm/cgame.qvm vm/qagame.qvm vm/ui.qvm

Speed up building process ?

Quake3 first contact and attempt to compile:

```
2>.\win32\winquake.rc(10) : fatal error RC1015: cannot open include file 'winres.h'
```

- winres.h cannot be located: Express Version has no MFC/ATL, this is where winres.h is.

```
Build all qvm files:
        QVM: game,cgame, ui


Now if you want to build

Using nmake from Visual Studio 2008 Command Prompt

Build cpp
        - "outp" is a microsoft function today, this will lead to an error from the compiler
        -> #define outp win32outp in cpp.h

        - memmove is part of win32 environment
        -> #ifndef _WIN32 in unix.c



make lburg //Help to generate the Backend code generator md -> c
make rcc //build rcc, tranform all md to c with previously built lburg
vi s
```

Q: What is the video format and how it is played.
A: RoQ file format from "The 11th hour" game. Used after Graeme Devine, the creator of the format joined id Software,
 the RoQ file format has been in use in every game the company has released such as Quake III, Return to Castle Wolfenstein
  and DOOM 3.

  While the format is limited and much lower quality than MPEG and Indeo Video, it was presumedly preferred by id Software
  because of the lack of royalties, the lack of patent liability that presents a serious problem with most video formats,
  and the absence of complex platform-specific APIs.

More here: http://www.modwiki.net/wiki/ROQ_(file_form

Quake3 VM, funny comments:

```
vm.c:
/*
VM_DllSyscall
Dlls will call this directly
 rcg010206 The horror; the horror.
*/
```

Explanations by Brian Hook
http://www.gamers.org/dEngine/quake3/bwh_gdc99.txt

Brian Hook quake3 gdc

Quake III Arena Shader Manual
http://graphics.stanford.edu/courses/cs448-00-spring/q3ashader_manual.pdf

Quake III makes extensive usage of Cyclic Redundancy Check, this article is amazing to help understanding it:
http://www.ross.net/crc/download/crc_v3.txt

Three great things in Quake3:

```
 - Virtual Machine
 - IA
 - Shader system
 - Network system


 Q3 bezier curves: http://www.gamasutra.com/view/feature/131755/curved_surfaces_using_bzier_.php?
print=1

 Amazing how much was pushed in the virtual machine:
  There is nothing in Quake3 loop that triggers the rendition, this is done from the Virtual Machine
calling a system call.

 Trinity idTech3 materialized the lack of interest in building game engine what would be licensed,
leaving it to Epic. idTech3 was an engine with the sole
 purpose to power Quake3, not much was really re-usable. In a lot of regards it may have been the
final act of the vision that John Carmack exposed to Michael Abrasn
 about virtual words.




Q3ASM.EXE  :
===========


lame hashfunction for table symbol

LCC bytecode is CISC which mean that all instruction have different size
depending on the parameters:

Q: Why specify jump offset in terms of "# instruction offset" instead of a
byte offset ? This requires a special operation when loading in order to
transform instruction offset to byte offset :( !

    As a result in Quake3, (vm_interpreted.c) when a VM is loaded and is intended to be interpreted
or compiled
    on loading time, a translation table is also created: vm.instructionPointer[] so instruction
    offset are converted to byteoffset. JMP is then possible.

A: Write the bytecode offset instead of the instruction offset would have prevented compiling to
native platform.


First pass is a big waste of CPU ressource since it is doing all the job and throwing everything
aways except for
the symbol definition.




Q3 RADIANT :
============

Problems :
  - Missing glaux.h
  - C++ scope abuse
                  for(int i=0; i < .....)
                  {
                  }

                  for(i=0 ; i<....) //I is not declared and should not be valide here
                  {
                  }
```

```
Even when quake3.exe needs to refresh the screen, it simply send a message to cgame vm.
Amusingly the cgame vm uses a quake3.exe system call to call the OpenGL rendering routine




3 Virtual machines:
===================

   vm_t                 *cgvm;       // interface to cgame dll or vm
   vm_t                 *uivm;       // interface to ui dll or vm
   vm_t                 *gvm = NULL; // game virtual machine // bk001212 init

   cgvm Client side vm, renderer
   gvm  Bots and Server side vm
   uivm GUI vm

   Q: Where are the virtual machine created ? VM_Create
   A: "cgame" vm  is command triggered
     CL_Snd_Restart_f
      CL_Vid_Restart_f
       CL_InitCGame
        cgvm = VM_Create( "cgame", CL_CgameSystemCalls, interpret );

   A: "ui"vm  is command triggered
       CL_Vid_Restart_f
         or
       Com_Init
        CL_StartHunkUsers
         CL_InitUI
          uivm = VM_Create( "ui", CL_UISystemCalls, interpret );


   A: "game" vm is command triggered
       SV_MapRestart_f
           or
       SV_Map_f
        SV_SpawnServer
         SV_InitGameProgs
          gvm = VM_Create( "qagame", SV_GameSystemCalls, Cvar_VariableValue( "vm_game" ) );


   Note: KEy events are either sent to the cgvm or uivm depending which one is declared as catcher
(cls.keyCatchers) cl_keys.c




   cgvm CG_DRAW_ACTIVE_FRAME path:

          Start on the quake3.exe side
            SCR_UpdateScreen   (client code)
             SCR_DrawScreenField
              CL_CGameRendering
               VM_Call( cgvm, CG_DRAW_ACTIVE_FRAME, cl.serverTime, stereo, clc.demoplaying );

            On the VM side:
                case CG_DRAW_ACTIVE_FRAME:
                   CG_DrawActiveFrame
                   {
                      CG_DrawActive
                      {
                          trap_R_RenderScene( &cg.refdef );
                            syscall( CG_R_RENDERSCENE, fd );

                      }
```

```
                    }

             Back on the quake3.exe

                case CG_R_RENDERSCENE:
                    re.RenderScene( VMA(1) );
                    return 0;
```

Mirrors: Scene is renderered multiple times...

RENDERER:
=========

A good article about Quake3 lightmaps:
http://www.gameversity.com/index.php?action=showtutorial&id=9&PHPSESSID=hj5udtb918gue2hk31kmn921b7

   Seems to be surface based (just like dEngine ;)) !

   re (refexport_t) is initalized in CL_InitRef.

   re.RenderScene is called in cgame vm via the CG_R_RENDERSCENE CL_CgameSystemCalls

```
     re.RenderScene
      RE_RenderScene   (In order to deal with mirrors, this may be called several times)
       R_RenderView
       {
          tr.viewCount++;

          R_RotateForViewer ();
          R_SetupFrustum ();

          R_GenerateDrawSurfs();

          R_SortDrawSurfs( tr.refdef.drawSurfs + firstDrawSurf, tr.refdef.numDrawSurfs -
firstDrawSurf );

          // draw main system development information (surface outlines, etc)
          R_DebugGraphics();

       }
```

TODO: Trace this weird contruct:

```
                static int (QDECL *syscall)( int arg, ... ) = (int (QDECL *)( int, ...))-1;

                void dllEntry( int (QDECL  *syscallptr)( int arg,... ) ) {
                     syscall = syscallptr;
                }
```

SPLINES LIBRARY:
================

Implementing Scripted Cameras in Vanilla Quake 3: http://rfactory.org/camerascript.html

BOTS:
=====

```
Seems to have five difficulty levels
    Q: What are the differences between the levels ?

Bot module function pointers are initialized in SV_BotInitBotLib:

     botlib_export = (botlib_export_t *)GetBotLibAPI( BOTLIB_API_VERSION, &botlib_import );

     Q: Does the same thing happen with botlib (aka: triggered in the kernel, passed to the vm and
sent back to
     the kernel module ?
     A: Yep it works exactly like the renderer crazy loop.



MULTIPLAYER:
    Q: Where is the game connecting for multiplayer, where does it find the list of servers
available ?
    A: Server list are requested from the master server.

              AUTHORIZE_SERVER_NAME    authorize.quake3arena.com
              MASTER_SERVER_NAME       master.quake3arena.com
              UPDATE_SERVER_NAME       update.quake3arena.com

              master.quake3arena.com resolves to 192.246.40.56 (Dallas,TX)

              authorize communication is done with OUT OF BAND datagram: "leading 0xff0xff0xff0xff"

              Note:
              =====
              A client will be accepted if a valid cdkey was sent by that ip (only) in the last 15
minutes.
              If no response is received from the authorize server after two tries, the client will
be let
              in anyway.

              ====> This mean that a server can be modified in order to accept invalid CD Key.
              Q: Does the authorize server check for the server integrity before it allows it to
join the q3 server list ?


          Client operation for connection in CL_CheckForResend..
           CA_CONNECTING, upon challengeResponse received, move to CA_CHALLENGING state
           CA_CHALLENGING

              Q: How autorize a Client ? The Server or the Client itself ?
              A: It seems the client autorize itself by calling the Autority server alone.

              CA_CONNECTING ->CL_RequestAuthorization:

Client................................................................Authorize Server
                          ...........> ....getKeyAuthorize(challenge,cdkey) > .............



           Quake3 seems to have used PunkBuster at some point but PunkBuster is closed source
software
           and all the hooks for it were removed from Quake 3 before the source was released to the
public.

           Note: I was still able to connect to Q3 servers without punkbuster....maybe I was only
able to access "non-pure servers".
                 According to Punkbuster wikipedia entry, quake3 arena support has ended...maybe it
is open for all now.

           Network communication are hard to understand with code only. Luckily a few people have
done some reverse engineering:
           http://www.tilion.org.uk/Games/Quake_3/Network_Protocol
           http://ra.is/unlagged/solution.html
           http://ilxm.blogspot.com/2011/01/zt-bookofhook-quake3-networking-model.html

           client sends   0xffffffff getchallenge
           server replies 0xffffffff challengeResponse <ID> this id is used for all subsequent
encryption between client and server
           client sends   0xffffffff connect "<CS>"  CS is huffman compressed details about client
           server replies 0xffffffff connectResponse

           NOTE: The huffman tree is precomputed for a text language. It is note transmitted with
every datagram.

           How is NAT bypassed ? Are they using UDP punching ?


VIRTUAL MACHINE :
=================
    .plan (Aug 16, 1999) I decided to go ahead and try a dynamic
code generator to speed up the game interpreters. I was uneasy about it,
but the current performance was far enough off of my targets that I didn欽檛
see any other way.


    The generated code is pretty grim if you look at it, in part due to the security measures (mask
and add for each load/store), and in part due to the fact that it is a straight bytecode
translation:


CLIENT-SERVER :
===============

The demo servers have general purpose
铿俓od-protection that has caused some confusion. (Nov 16, 1999)
Clients are only allowed to make one command a second of any kind.






shader system (more here: http://www.gamers.org/dEngine/quake3/UQ3S)
                Q3 uses procecural textures defined by "shader scripts".
                See the Renderman API, as well as:
                  "A Shading Language on Graphics Hardware:
                   The PixelFlow Shading System",
                   Olano, Marc and Anselmo Lastra, UNC Chapel Hill,
                   Proceedings of SIGGRAPH 98.
                (available on the web, as is an earlier 1995 paper).

                Q3Map2 Shader Manual (http://q3map2.everyonelookbusy.net/shader_manual/ch1.htm#what)


Monolitic approach: No need for modularity anymore since there is no single player experience and
only renderer is OpenGL.
        - QVM

           From: http://www.gamers.org/dEngine/quake3/UQ3S ([4-1] VM/DLL Handling)
                        However, certain libc and other external C functions (networking
                etc.) available to a DLL will not be available on the VM. You can't
                link against any libraries, so every function must be resolved.
                Functions like strcmp(..), memcpy(..), rand(), etc. must all be
                implemented directly. Q3A's VM and CGame source will provide code
                and hooks for all the ones id uses, but mod coders may have to
                modify their coding styles or provide standalone implementations
                for missing functions.

IA Bots
Memory system
Inverse square root
Lagometer ?
```

```
THIS ENTRY IS GOLD:
http://www.gamers.org/dEngine/quake3/UQ3S


README.txt: Not all code is GPL, libs for dealing with PCM,jpeg,md4 are different licences


A short summary of the file layout:

code/                                   Quake III Arena source code ( renderer, game code, OS layer
etc. )
code/bspc                               bot routes compiler source code
lcc/                                    the retargetable C compiler ( produces assembly to be turned
into qvm bytecode by q3asm )
q3asm/                                  assembly to qvm bytecode compiler
q3map/                                  map compiler ( .map -> .bsp ) - this is the version that
comes with Q3Radiant 200f
q3radiant/                              Q3Radiant map editor build 200f ( common/ and libs/ are
support dirs for radiant )

code projects
For cgame: c:\<quake3 install dir>\baseq3\cgamex86.dll
For game: c:\<quake3 install dir>\baseq3\qagamex86.dll
For q3_ui: c:\<quake3 install dir>\baseq3\uix86.dll


Code statistic for entire source:
=================================
                        SanglardFa@ond2c00558095 /cygdrive/c/opt/quake3-1.32b-source
                        $ cloc quake3-1.32b/
                            1356 text files.
                            1183 unique files.
                             378 files ignored.

                        1 error:
                        Unable to read:  quake3-1.32b/code/quake3.ncb

                        http://cloc.sourceforge.net v 1.53  T=9.0 s (103.1 files/s, 56369.7 lines/s)
                        -----------------------------------------------------------------------------
---
                        Language                     files          blank        comment
code
                        -----------------------------------------------------------------------------
---
                        C                              428          43887          65338
238723
                        C++                            129          11203          12920
52686
                        C/C++ Header                   308          10563          13269
39019
                        Objective C                     12            886            744
3128
                        Perl                             4           1278           2954
2798
                        make                             6            279            230
2198
                        HTML                             3            176              0
1657
                        Assembly                         7            219            282
1150
                        DOS Batch                       14             39             10
478
                        Bourne Shell                    10             67             50
352
                        CSS                              1              0              0
207
                        XSLT                             1             28              2
202
```

```
                    yacc                        1             16             1
185
                    XML                         1              0             0
95
                    Teamcenter def              3              0             0
8
                    -------------------------------------------------------------------------------
---
                    SUM:                      928          68641         95800
342886
                    -------------------------------------------------------------------------------
---
```

Great link to understand Quake3 virtual machine:

FORMAT: http://icculus.org/~phaethon/q3mc/q3vm_specs.html
BUILD : http://www.btinternet.com/~AnthonyJ/tutorials/QVM.html
HOW IT WORKS: http://www.gamedeception.net/threads/19198-Runtime-QVM-Modification

Poking around: MUCH MORE commentaries all over the place: Yummy !

Q: where do we start ?
Partial-A: Find the main.
A: WinMain is in win_main.c in the quake3 project

Engine detect pentium, mmx,3dNow and KNI
Background stream file loading does NOTHING (void methods).

New C notation convention: leading bracket is on the same line as the "if". Closing bracket is indented

NETWORKING :
============

Introduction article by Brian Hook:
http://trac.bookofhook.com/bookofhook/trac.cgi/wiki/Quake3Networking
Quake 3 Networking Primer: http://www.ra.is/unlagged/network.html
Excellent paper about time synchronization for game mirroring:
http://warriors.eecs.umich.edu/games/papers/netgames02-tss.pdf

Since prediction is in a virtual machime, modders were able to write their own lag compensation mecanisms:
http://unlagged.com/
http://www.ra.is/unlagged/faq.html

It seems only the OOB (Out of Band) Connect packets are huffman compressed. The huffman compression tree is preprocessed according to what
will likely be send.
http://www.tilion.org.uk/2011/11/quake-3-network-format/
http://aluigi.altervista.org/papers/q3info.txt
http://caia.swin.edu.au/reports/070730A/CAIA-TR-070730A.pdf
http://www.flipcode.com/archives/Network_Game_Programming-Issue_07_I_bent_my_Wookie.shtml

Checking out the renderer stages:

Disable entities: r_drawentities 0

```
 Disable HUD     : bind j "toggle cg_draw2d"
 View Lightmaps  : r_lightmap 1
 View diffuse texture only:
      1. Edit tr_shade.c
      2. Add GL_Bind( tr.whiteImage ); after // lightmap/secondary pass R_BindAnimatedImage( &pStage-
 >bundle[1] ); in DrawMultitextured
 Screeshot       : /bind F11 screenshotjpeg
```