

Doomsday Engine - Feature #1794

Mobile apps and shared client/server code (more modular code structure)

2014-05-02 15:23 - skyjake

Status: Closed	Start date: 2014-05-02
Priority: Normal	% Done: 40%
Assignee: Deng Team	
Category: Redesign	
Target version:	
Description	
<p>Currently, the client and the server are separate applications but share some of the source code files. This is not an ideal situation, because code is duplicated, preprocessor segments <code>__CLIENT__</code> and <code>__SERVER__</code> can get cumbersome/confusing, and the basic structure lacks flexibility.</p> <p>In the future, for instance an iOS app needs to contain both the client and the server components wrapped under an iOS-specific UI.</p> <p>The common code shared by the client and server (basic map data, resource handling), needs to be separated into its own (shared or static) library, so that both applications can use it. (The client would extend some of the classes with GL, etc. functionality.)</p> <p>The naming of the libraries could be improved:</p> <ul style="list-style-type: none">• libdeng2 => libcore• libdeng1 => liblegacy (support for legacy code) <p>New libraries:</p> <ul style="list-style-type: none">• libserver => static library with server-specific, platform-independent functionality• libclient => static library with client-specific, platform-independent functionality• libdoomsday => library containing code shared by the client and server (GPL)	
Related issues:	
Related to Feature #1887: Migrate project build system to CMake 3	Closed 2014-10-20
Related to Feature #1991: Qt 5.5 as minimum requirement for GUI, Qt 5.x for n...	Closed 2015-03-10
Related to Feature #1675: Switch to new Qt 5.4+ OpenGL APIs (from QGLWidget)	Closed 2013-10-24
Related to Feature #2028: libcommon uses the preprocessor excessively in orde...	New 2015-05-03

History

#1 - 2014-05-04 08:13 - skyjake

- % Done changed from 0 to 10

#2 - 2014-11-23 20:56 - skyjake

- Assignee changed from skyjake to Deng Team

#3 - 2015-02-24 19:00 - skyjake

- Description updated

#4 - 2015-03-21 22:49 - skyjake

- % Done changed from 10 to 30

#5 - 2015-03-21 22:53 - skyjake

- % Done changed from 30 to 20

The work in the cmake-build branch brings some improvements for modularity. Also, Qt 5 has better support for mobile OSes overall.

#6 - 2015-03-21 22:54 - skyjake

- Related to Feature #1637: Switch to Qt 5 added

#7 - 2015-03-21 22:54 - skyjake

- Related to Feature #1887: Migrate project build system to CMake 3 added

#8 - 2015-03-21 22:54 - skyjake

- Related to Feature #1991: Qt 5.5 as minimum requirement for GUI, Qt 5.x for non-GUI build added

#9 - 2015-03-21 22:54 - skyjake

- Related to deleted (Feature #1637: Switch to Qt 5)

#10 - 2015-03-30 21:11 - skyjake

- Related to Feature #1675: Switch to new Qt 5.4+ OpenGL APIs (from QGLWidget) added

#11 - 2015-03-30 21:14 - skyjake

- Subject changed from Modularized project structure to enable mobile apps, shared client/server code to Mobile apps and shared client/server code (more modular code structure)

#12 - 2015-03-30 21:14 - skyjake

- Category changed from Enhancement to Redesign

#13 - 2015-05-03 13:54 - skyjake

- Related to Feature #2028: libcommon uses the preprocessor excessively in order specialize for doom, heretic, and hexen added

#14 - 2015-05-03 14:11 - skyjake

- Tags set to CodeQuality

#15 - 2016-07-05 23:57 - skyjake

- Status changed from In Progress to Progressed

#16 - 2016-08-07 10:41 - skyjake

- % Done changed from 20 to 30

Progress in resource management.

#17 - 2016-08-07 10:42 - skyjake

- Status changed from Progressed to In Progress

#18 - 2017-10-30 07:19 - skyjake

- Status changed from In Progress to Progressed

- % Done changed from 30 to 40

A mobile build for iOS has been successfully made, although via non-modular means: compiling everything statically into a single binary.

#19 - 2019-11-29 16:31 - skyjake

- Status changed from Progressed to Closed

Closing as obsolete.