Doomsday Engine - Feature #2028

libcommon uses the preprocessor excessively in order specialize for doom, heretic, and hexen

2015-05-03 09:11 - rhargrave

Status:	New	**Start date:**	2015-05-03
Priority:	Normal	**% Done:**	0%
Assignee:			
Category:	Redesign		
Target version:	Architecture		

Description

As the title states, `common` makes *extreme* use of the C/++ preprocessor in order to provide specific modifications for one game or another.

For example, in `hu_lib.h` the following is defined:

```
#if __JDOOM__ || __JHERETIC__
typedef struct {
    int slot;
    keytype_t keytypeA;
    patchid_t patchId;
#  if __JDOOM__
    keytype_t keytypeB;
    patchid_t patchId2;
#  endif
} guidata_keyslot_t;
#endif
```

If this such a type is only needed by *two* (*three* counting doom64) different plugins, then it should be defined at a plugin level, rather than putting it in the common plugin. On one hand, this will vastly reduce code fragmentation by moving single-plugin-only code in to the appropriate plugin, and on another it provides motivation to use some C++ features (as I understand, I fairly large portion of the code at hand was ported from C to C++, and as such was not written with classes, etc.. in mind).

Not knowing the nature of every usage off the top of my head, I would suggest redesigning this and similar situations as follows:

```
// in common/include/hu_lib.h
class guidata_keyslot_t {
    // common things here
}

// in doom/include/_pick a name_

class guidata_doom_keyslot : public guidata_keyslot_t {
   // common things here
}
```

Even furthermore, migrating static helper methods for these types in to such classes as instance methods would *drastically* reduce the number of symbols in the global namespace, which would help for (among other things) debugging, code clarity, and code sharing.

Related issues:

Related to Feature #1794: Mobile apps and shared client/server code (more mod...	**Closed**	**2014-05-02**

History

#1 - 2015-05-03 09:43 - rhargrave

- Tags changed from libcommon, CodeQuality to libcommon, CodeQuality, Cleanup

- Category changed from Enhancement to Redesign

Add `Cleanup` tag and change category to `Redesign`

#2 - 2015-05-03 13:54 - skyjake

Indeed, libcommon is quite a mess. Since its introduction, the long-term objective has been to get rid of the #ifdefs and arrive at a clean, "normal" shared library that provides the common functionality for the game plugins. However, due to the size and general messiness of the game plugins, this has not been achieved yet. Furthermore, it is extremely easy to break gameplay in large or subtle ways when trying to untangle the #ifdef branches into a clean set of code. Therefore we've been living with the current arrangement.

Whenever working on this topic (informed by my past experiences) I would recommend approaching it in small, easily mergeable portions so that possible issues can be weeded out as we go.

For historical context, it was my decision to go with ifdefs when I split Doomsday from jHexen and added support for the other games. It was not a wise decision for the long term, but it did allow me to quickly get rid of the most obvious overlaps between the Doom, Heretic, and Hexen code bases.

#3 - 2015-05-03 13:54 - skyjake

- Related to Feature #1794: Mobile apps and shared client/server code (more modular code structure) added

#4 - 2019-11-29 21:22 - skyjake

- Subject changed from The `common` plugin uses the preprocessor excessively in order specialize for doom, heretic, and hexen to libcommon uses the preprocessor excessively in order specialize for doom, heretic, and hexen

- Target version set to Architecture