

# Law of the Level

Posted at 5:12 AM on September 23, 2009 by Sheppard Mullin

## Who Owns The Tools Of The Trade?

Let's start with the basics. Under a typical game development contract, a publisher furnishes the funds required to develop a game to a game developer. Understandably, access to these funds is subject to certain restrictions, expectations and other contractual obligations. Invariably, there is a provision dedicated to the handling of the rights to intellectual property created pursuant to the terms of the agreement.

Now this is where it gets tricky. How are these rights handled? If the funds are coming from a publisher, the developer can expect to see the phrase "Work Made for Hire" or some other language allocating the intellectual property rights for works created pursuant to the contract to appear somewhere in the agreement. This provision typically specifies that the intellectual property created by the developer will belong to the publisher if it falls within certain parameters (often anything "related to" the project or falling under a list of defined terms).

Certain traditions are observed in video game contracts. The game (which normally includes the finished product, the source code, and all of the rights to exploit the underlying intellectual property) will be spoils of war for the publisher. The developer may rejoice and pour numerous libations in celebration of the fact that it has probably retained rights to the tools of the trade such as the game engine and internally created development tools.

This is where the potential for disaster begins.

Even contracts that contain excruciatingly long descriptions of the handling of the intellectual property can run into problems if the developer-publisher relationship breaks down. As a game moves from the drawing board to an executed product, the odds are that the line between "Tools" and "Game" may difficult to distinguish. As specified by the language of the contract, the publisher will typically have rights to the source code and the developer will have rights to the tools and game engine. The problem is that may not be easy to determine where one begins and the other ends.

This is particularly true if the developer and publisher part ways during development. A publisher may argue that the source code necessarily includes certain aspects of the engine and other developer tools since a workable build cannot be compiled without that framework. Conversely, a developer may argue that the viability of the code is irrelevant where a contract details the parties' rights to a code without reference to the value or usefulness of the intellectual

property.

The best solution will of course depend upon context, but one can imagine possible solutions falling into one of three categories: technical, legal, and business. An exhaustive treatment of these solutions would take too long and put me out of a job, so I'll give an example of each:

**Technical:** Throughout the development process, tag modules of code as the publisher's or the developer's when they are programmed. Obvious drawback: Time, patience, loss of sanity. Obvious benefit: There is a mutually agreed upon separation that is hard to dispute if everything goes up in flames.

**Legal:** Create a term in the developer-publisher agreement that will specifically handle that possibility of a termination that results in unfinished source code that requires the developer's engine to proceed. For example, grant the publisher a non-exclusive engine license to the publisher for the creation of the game or for five years, whichever occurs first. Obvious drawback: Contracts were often born to be litigated, and one can imagine arguments regarding whether the source code is "unusable" without an engine. Obvious benefit: Much cheaper than the technical solution above (unless you have very expensive attorneys), particularly if you pre-negotiate the license.

**Business:** Realize that the industry is a pretty congenial place, assume everything will be fine and hit the pub. Obvious drawback: Does not plan for the possibility that everything will not be fine. Obvious benefit: Camaraderie and Irish drinking songs. Or, in the alternative, you could have designated staff review the handling of the tools and the source code for each development milestone to determine whether adjustments in operating procedure need to be made.

In determining the best solution there are many questions that should be answered, but here are a few biggies:

- What intellectual property will be created or otherwise involved in the development process?
- Who will own that intellectual property?
- Will there be any assignments/licenses regarding that intellectual property?
- What are these assignments/licenses going to look like?
- How will you keep the intellectual property separate?
- What happens if the project falls apart a month in? Two milestones in? The day after completion?
- More epic cheat: Street Fighter II Same Character Code or Contra 30 Lives Code?

The goal is clarity. Everyone wants to make a game, and hashing out a contingency plan for a rainy day is never an undertaking anyone relishes; however, securing certainly during contract negotiations is significantly cheaper than paying for it once a dispute arises. Besides, litigation never makes anyone new friends.

Authored By:

[Shawn Foust](#) (GamerTag: OMG SRSLY)  
(310) 228-3724  
[sfoust@sheppardmullin.com](mailto:sfoust@sheppardmullin.com)