# The Game is Not the Medium

... or “How To Ignore the Shiny Box” - Julian Oliver 2006

*The following paper can be read as a continuation of my 2003 DAC paper, 'Developers In Exile'.*

If there is such a thing as *artistic game development,* then it is natural to assume the medium of the artist must be videogames themselves. When people talk about mediums, they usually refer to a single output format, the medium of film follows this case. In other examples, the medium also describes the process, as in painting. With game development, however, it is never as clear, so, when people talk about the *medium of the game,* what are they referring to? What is this medium comprised of?

Unlike a film, a game cannot be easily considered a singular and discrete object, and this applies to any artistic mod based upon it. Artists are manipulating or creating data of a great many types: sound files, textures, models, scripts for game logic and even video, in the course of making their games. Once gathered together in the right formats and places on a machine it then needs to be run by software to even provide the experience of a game at all.

This software, usually known as the game-engine, is complex from an engineering perspective; it is comprised of many separate parts: event managers, sound server, renderer, input server etc, and, as such, it can take a very long time to write one alone. For this reason, the modder is in an uncommon artistic position at the outset; to work with games is to use other people's work, other peoples material.

**Modding: Art on the Shoulders of Giants**

From the perspective of the fine artist, this relationship with the medium through the work of others is a defining one, often shaping how they engage with it altogether. The artist finds a game or game-engine that interests them for its technological or aesthetic offerings, learns how it works and then starts to pick it apart. The more they learn, the more they can pick, and so, at the very base of artistic modding, is a strong hacking culture, where *to hack, is to make*. This makes it difficult to determine where the art (as an object) begins and ends, given that no artist is ever wholly responsible for every component of the delivery of their work; from the player, through software and down to system hardware. For reasons I'll explain later, this puts artists in a vulnerable position regarding distribution of their work and also shapes the kind of work they actually make.

Are artistic mods then akin to Graffiti Art, where the artist uses existing property to 'host' their work, or does it go deeper? Maybe they could be likened to altering the electronics of a video camera so that it records the world in a way it was never intended to?

Whatever it is, game modification is an unusual technical practice. It is to learn how something is made enough to transform it into something else, but a something that still references and needs it's original. No artistic modification can be considered out of the context of the original - the resulting work points both to its current and parent forms simultaneously. At the same time, however, art-mods innately challenge a mass-market driven design paradigm where consumers are gathered into large interest groups or demographics and then marketed games on that basis; every mod can be seen as a way of personalising the original and, in so doing, they affirm that the medium is larger than the form in which it was given, that it is larger than the game.

For this reason *artistic* modding is especially historically important; it steps completely out of the market of intended use, yet becomes intimate with the product at a *material* level. It peels

back the layers of awe, techno-prowess and glitz to reveal it as a system of working parts that can be understood and repurposed.

While doing this, art-modding affirms secondary ownership of the game itself, where each art-mod is as an *expression of the right to mis use*. It is here we have a glimpse of our medium, describing with every hack how it is fundamentally separate from the game that uses it.

In this way artistic modding shares something in common with 'Found Art', a chapter of fine art that sought to re-invent our relationship with ubiquitous objects, to challenge the rarification of objects in the context of Art. As with Found Art, modifying an existing game by changing its presentation context, changing parts, removing parts, or even adding new parts, is innately a commentary - a critique of the original through a strategic misappropriation of intended use. By exploring new ways of seeing the original, often lifiting an element out of context, exploiting a graphical glitch[1] or emphasising one element over the next, artistic mods affirm the game as a medium beyond mere *product* in an economic climate that monopolises both distribution channels and the means of production, whose unerring drive toward increasing fiscal turnover pushes game design toward the ambitions of the American motion picture industry (of animated features in particular) with increased emphasis on graphic realism as a primary selling point.

Several companies have actively supported game modification as a way of ensuring their products stay on shelves longer; users create and share new content that will only run on their game. The first known occurrence of this was 'The Bard's Tale Construction Set', released in 1991, which enabled users to create new adventures for the game. Since this time, game modification has become a commercially endorsed phenomenon in a large number of game products, the most famous of which was Valve Software's release of 'Half-Life' which resulted in the commercialisation of a user created[2] mod called 'Counter-Strike', now played by millions of people the world over. Now it is not uncommon for companies to ship tools and software development kits (SDK's) with their games to assist and encourage users in the creation of content for their existing product, some of which may manifest as a saleable product.

While much of the work coming out of the modding scene is interesting, a separation needs to be made between artistic modification and community made and distributed mod addons, not just on the level of creative intent, but also in process. At the base of artistic modding is a strong hacker antic and so the 'curated' or 'intended use' modification kits supplied by game producers are often not satisfactory for an artist engaged in the process of media reclamation or seeking unanticipated, uncommon consequences from the hacking process.

Retroyou's 'Nostal(g)' work is a good example of this distinction in practice. Retroyou has gone through the data directory of an existing flight simulator, found files modifiable in a text editor and explored the consequences. The result is a graphic abstraction of the mathematical and geometric universe at work in a flight simulator. It is completely unrecogniseable as a flight-sim, yet the same calculations are being made: somewhere amidst the shearing lines and numeric output a plane flies in a pure Euclidean space.

Nostal(g) expresses the way chance plays an important role in the modding process, where development is a process of *re-discovering the game*. The idea of re-discovery is critical to artistic modding, and often affirms a clear distinction between the commercially ordained or supported modding practices and the work of the game hacker. Here the game is rediscovered through art but the medium itself is exposed through an explication of internal working parts, and simultaneously knowledge is transferred to the artist.

Is artistic modding an abuse of the game? Yes of course. But it is not an abuse of the medium. To abuse the *medium* of the game is to merely play it.

---

1 See S.Honegger's pioneering work 'Margin Walker'. While not a mod, this work beautifully expresses the edge between a game and the medium that supports it.

2 Counter-Strike was created by Minh Le in 1999, following the release of Valve's Half-Life the year before.

## Mapping: Editing the Entry-Level

Repurposing games is not exclusive to modding of course. The art of 'level-editing' or 'mapping' (often confused with modding) involves the production of art to be run by the game-engine and is made using image editors and level-editing tools either shipped with the game or otherwise made externally available. Level editing is at an entry-level accessible enough for visual artists with existing abilities in 3D-modeling, digital sound and image creation. This has resulted in a great number of artistic 'levels' that explore architectural, political and situational applications for first person computer games most particularly.

A primary trend in artistic level editing is the practice of making places, or finding new ways of being in an already known site. In this way level editing has truly domesticated the previously rarefied practice of simulation and, since it's inception, such artists have rigorously explored the making and distribution of digital places; creating experiences otherwise not available to us in the corporeal world.

These range from: astute recreations of a known site but with new scenarios[3], modifications to the appearance of an already known site, entirely fictitous sites[4], virtual 'knowledge architectures'[5], through to digital architecture 'played' as a musical instrument[6]. These levels are not uploaded as death-match arenas to be used as the site of virtual combat, instead they appear in galleries, public spaces, architecture studios and university design classes. None of these works seek to *entertain* in the strict sense. They reveal that through merely changing or substituting the art-assets alone (using 3D modelers, sound and image editors) an entirely new application for the game, as a medium, can result.

Most have grown so used to the video game as an object in a shiny box on a shelf in a store, and so to consider it as a rich intersection of mediatypes - images, sounds, polygons and human performance - seems unnatural. In this way game artists are doing something that no-one else is. They are affirming cultural (and sometimes even practical) value in the medium of the game outside of a mass market and it's intended uses. The artist recognises, that when they buy a game, it is theirs to do whatever they like, that it is ultimately a collection of files on disk whose relationships, when newly described, provide great creative possibility.

It is a critical time to be thinking in these ways, to learn to separate the *medium* from the *game*.

## Second Hand: Art After the Market

At first it would seem that artists don't suffer from a lack of innovation in the game industry, it is of course their self-assigned quest to innovate regardless, but this is not the case. As the market becomes larger it increasingly generalises its sales targets, genres become established and so do modes of gameplay. [7]

When you talk about a genre of game, you're also talking about a certain kind of game-engine. There is no such thing as a generic game-engine, they all have innate 'breeds' of intended use, some more so than others. What we're seeing however is an *increasing* homogenisation of technology, whose primary development rationale centres around 'out-aweing' competitor's offerings, on both a rarefied hardware architecture and in many cases, privileging a single operating system[8]. Fewer and fewer people, year on year, are directly responsible for the design directions of gaming technology at an architectural level. The medium of the game is increasingly and soley engineered to serve mass capital interest and this directly affects how it is, and can be, used later.

---

3 See B.Condon's 'Chinatown'
4 See the unique work of artist Scott Swearingen
5 See Fuchs-Eckermann's 'Expositur'
6 See J.Oliver's 'QTHOTH'
7 Needless to say the console market is more diverse than the PC gaming market where innovation is concerned, in particular the Japanese console market.
8 You guessed right..

Where mass-markets are concerned, a comparison between the medium of the videogame and that of film produces immediate parallels but, at the fringe, things are quite different. Low budget films, short and experimental films, are already widely recognised as a culturally important form. Since the advent of consumer grade digital video cameras, it's arguable that the cost of producing and distributing films has actually dropped, whereas for game development, as a whole, it has vastly increased[9].

What was once a lively 'grass-roots' industry, an early culture of development that gave us video games altogether, is now a multi-billion dollar industry that doesn't believe it can afford to innovate. Riddled with software patents[10], IP lawyers and product distribution monopolies, games are more expensive to take to market year on year.

Because of the entry level cost of making a game that can compete in the market is so high, developers become dependent on external investors. However, to have any financial backing at all you need a Publisher, and these people have very clear ideas about what they think will do well in the market[11]. It's here, inadvertently, we have our real game designer, operating as a kind of filter for game designs by deciding which will and which won't go to market. In the 21st century, the Game Designer is ultimately the Game Publisher.

**Haunted Code**

With each AAA title sold in stores there will be dozens that will seek to emulate, or ride-upon, it's success. The first person shooter is just one of many possible genres that could have manifested from the idea pool of possible games, yet it's influence on the market as a whole has produced a specialist breed of technology that we all now consider a near default for the term 'computer gaming engine'. Engines of this kind are QuakeIII, Unreal Tournament 2004, Halo, and Doom3.

These software engines, not the games themselves, are built around a very particular kind of digital subject with a very particular sort of representative body. This digital body is provided a sphere of possible events comprised of picking up items, messaging other players and suffering damage. These components of the first person engine, along with game features from the specific games themselves, are so deeply coded into the technology itself that developing a project beyond these constraints is very difficult. In this way the artist, using an existing game as the basis for their own work, is inadvertently steered by these limitations, guiding their creative process through what they can and can't do. To work with these engines is therefore, in part, to work 'within' the industry; art created using this technology is simultaneously an expression of the industry, its ambitions, its broader project.

Should the artist decide to choose another game that is more appropriate to their final outcome, they are still guided by what is available. The market has chosen for them. This is like the canvas artist having their paint colours chosen by someone else before they begin painting: how much of the *medium,* as a raw field of possibilities, is really available to the artist?

As Greg Costikyan clearly puts it:

"We're only 30 years into the gaming revolution. Additionally, games are an enormously flexible form: They've been created with every technology from the Neolithic to the modern. And software is an enormously flexible medium, too; if you can specify it, you can implement it. We've gone from three genres to dozens in a few short decades, but we've charted only the merest coastline of a vast, virgin continent."

So as the market homogenises, so do the tools, and this means more 'creative undoing' for the artist. This is all very well if the artist is interested in modification as a directed critique of the

9 USD 200,000 to make a game in 1992. 5 to 10M in 2005. G Costikyan, 'Death to the Game Industry' The Escapist, Issue 8

10 See my article 'Patenting Play in the European Parliament': http://www.selectparks.net/modules.php?name=News&file=article&sid=256

11 See my paper, 'Developers in Exile' 2003: http://hypertext.rmit.edu.au/dac/papers/Oliver.pdf

original form but, much of the time, it is the raw potential of the medium that draws the artist to work with games, and so they must look elsewhere for their blank canvas.

Where game development is concerned, the equivalent of this 'raw medium' is *source-code*, or the set of instructions describing the interoperation of parts in a game, and how it accesses the hardware system itself. Source-code is generally *compiled,* meaning processed in such a way as to become the kind of file you can't open up in a text editor, a *binary* file. Compiled code generally runs faster, and for the kind of computer languages that most games are programmed in, it's often necessary to compile it anyway.[12]

The moment parts of the medium, in this case the game-engine[13], are no longer accessible in a text editor, they cannot be modified. This is known as *closed-source software* and it generally costs a lot of money to have the right to read and modify the source-code that produces it.

Here is an excerpt of the licensing terms for Epic Game's 'Unreal2' Engine. It outlines the cost of access to engine source-code and revenue return.[14]

"A non-refundable, non-recoupable license fee is due on execution of the agreement. The cost is US $350,000 for one of the available Unreal Engine 2 platforms, plus US $50,000 for each additional platform. A royalty of 3% is due on all revenue from the game, calculated on the wholesale price of the product minus (for console SKUs) console manufacturer fees. In the case of massive-multiplayer online games, the royalty is also due on the additional forms of revenue including subscriptions and advertisements."

Closed-source software presents a real problem if we want to consider videogame technology as a medium in it's own right; the reliance of so many artworks on this expensive intellectual property greatly restricts its distribution and rights of sale. An artistic modification of a commercial engine cannot be legally sold to a museum for instance, to do so would mean the artist needs the rights to sell modified game-code and artwork. Mods cannot also be distributed *with the game* to an interested gallery for redistribution. Work made does not, in a very clearly outlined legal sense, belong to the artist. Most importantly though, and this is something not mentioned enough, is that with this maldistribution of code comes a maldistribution of knowledge. *Those that can afford to see the code, and work with it, are learning the most*. It's here that we see that the artist hits a glass-ceiling; a hard delimiter between the technical potential - and therefore knowledge capital - of Artists and Industry.

Were these restrictions also applying to the medium of film, we would not have the field of video art or perhaps even a short-film industry today. If digital-video file containers like AVI, MOV, OGG cost vast amounts of money to license for distribution, the digital film industry would be at a standstill; there would be no YouTube, no Google Videos and very boring and small short-film festivals. A career as a game-based artist, as opposed to that of a film-maker or traditional artist, requires either accepting this impediment – and scaling down ambitions accordingly - or investing development efforts elsewhere.

Put simply, if games are ever to have a scene as prolific and critical as that of short-films, the cost of material needs to drop to that of a handycam, rights must reside with the artist, and the presentation medium needs to be as generic as any domestic computer [15]

How do we, as artists, get to this state of a *truly* Independent Game Development practice? Is it possible to engage our medium without these innate restrictions and without genres and modes of game-play hard-coded into our work?

---

12 C,C++ are typical formal languages of this kind, known as 'compiled languages' unlike Python, which is an 'interpreted' computer language and so is innately more open.

13 Game-code needs to be considered separately from the source-code of the game engine itself. Game-code provides an interface to aspects of the game relating to game logic, but not the 'lower-level' components of the game-engine itself. While game-code is often shipped with an SDK, source-code for the engine itself, is not.

14 Taken from the Unreal Technology licensing page at the date of writing this paper: http://www.unrealtechnology.com/html/licensing/terms.shtml

15 Cited from 'Developers in Exile', Oliver 2003

## Art Beyond the Box: Toward a Sustainable Artistic Development Culture

Artistic modifications act as an important critique of specific games themselves and the culture of play that surrounds them. They also expose the assumptions game developers make about us as players, by providing alternative play-models, aesthetics and game-logics. At some point however - if game development as an artistic practice is to progress - we need to work at a lower level. We need to take things into our own hands and educate ourselves about tools and technologies that give us greater freedom as artists; freedom to build whole new kinds of games, share, sell and distribute them as we like. We need to pass the knowledge and techniques we aquire amongst ourselves to ensure the medium of the videogame is kept alive and open for the purpose of rich and diverse invention.

It's precisely here that *open-source* software and its related development models can play an key role in the sustainability of videogame technology as a *freely accessible medium* for artists. As long as we work with code that is open, and share that code as a result, the medium of the videogame can develop independently from a marketplace that otherwise seeks to contain it.

Let the game industry do what the hell it likes, but let us artists do what we like with what we make, and take our ideas as far as we need to. Only by using (and writing) code and tools that are free[16] and open will we continue to educate ourselves while defining the medium for our own purposes, in our own time and in our own way.

**References:**


M. Friedler, ***The Video Game Business is Broken,*** Game Daily,
http://biz.gamedaily.com/industry/feature/?id=13236

G. Costikyan, ***Death to the Videogame Industry***, The Escapist, Issue 8,
http://www.escapistmagazine.com/issue/8/3

Anonymous Game Developers, ***The ScratchWare Manifesto,***
http://209.120.136.195/scratch.php

R.Stallman, ***Patent Absurdity***, The Guardian, June 20 2005,
http://technology.guardian.co.uk/online/comment/story/0,12449,1510566,00.html


16 For a definition of 'Free' in the context of software, see: http://www.gnu.org/philosophy/free-sw.html