Platformer as Platform: *LittleBigPlanet* and the Limits of Protocol

Brian Keilen

University of Wisconsin-Milwaukee

The platformer, a game in which the player must traverse the environment by jumping between various platforms, is one of the oldest video game genres. In fact, some of the most iconic video games of all time have been platformers, from arcade games like *Donkey Kong* (1981) to the game that has nearly become synonymous with video games themselves, *Super Mario Bros*. (1985). It is appropriate, then, that Media Molecule's *LittleBigPlanet* came in the guise of a platformer. On the surface, the game appears to be not unlike any other entry in the genre. But *LittleBigPlanet* added one major twist when it was released for Sony's PlayStation 3 console in the fall of 2008: The game came with a robust set of tools that allowed players to build their own levels.

In addition to the level-building tools, *LittleBigPlanet* also featured several other deviations from the traditional platformer format. Not could players design their own levels, they could also share these levels with gamers across the globe via Sony's PlayStation Network. Once levels were uploaded, they could be tagged with various descriptions and rated with a star system. A sequel to the game, *LittleBigPlanet 2*, was released in 2010 and gives users even more powerful tools, even allowing them to experiment in different genres. Although these changes may have been subtle on the surface, they marked a major shift in the way video games are packaged and consumed. Instead of a finite amount of gameplay on one disc, *LittleBigPlanet* promised a game that could potentially never end, as long as it was supported by the players and community, of course.

With its emphasis on player created content and the direct connections formed between users sharing levels, *LittleBigPlanet* shares many characteristics with the distributed networks described by Alexander Galloway in his *Protocol: How Control Exists After Decentralization* (2006). Unlike centralized and decentralized networks, which retain main nodes that users must

appear to, in a distributed network "each point…is neither a central hub or a satellite node—there are neither trunks nor leaves…each node in a distributed network may establish direct communication with another node, without having to appeal to a hierarchical intermediary" (p. 11-12). This model appears to remove the hierarchical boundaries between developers and game players. Instead of directing the player how to play and beat the game, the developer created levels included with *LittleBigPlanet* serve as blueprints for user-generated levels, instructing player on how to create levels of their own.

Specifically, in *LittleBigPlanet* protocol operates in two ways. First, it manifests itself on the level of the system—the sum of the interactions between software, hardware, peripherals, etc.—itself. The system as a whole dictates how players are able to interact with the game and, in the case of *LittleBigPlanet*, determines what options are available for users creating their own levels. In my conception, the system is synonymous with platform and will be discussed in more detail later on. In this case, the system allows for a relative lack of constraint on the user. As I have previously stated, *LittleBigPlanet* provides a plethora of tools that allow users to create at will. While they may seem complimentary, in this case a constant tension exists between the system and the freedom to design user-generated content and levels.

On the surface this breakdown of the distinction between creators and users may appear to be a step towards a utopic future. But Galloway also notes that protocol is dangerous "because it acts to make concrete our fundamentally contingent and immaterial desires (a process called reification), and in this sense protocol takes on authoritarian undertones" (245). In this paper, I argue that *LittleBigPlanet* does just that. In moving from platformer to platform for development, the game encourages players to make real their innermost desires and unlock their creative energy. But it is in this process of making real our desires that Sony is able to exert its control,

for once these desires are real they can become commodified. Viewed as a platform for the creation of platformers, *LittleBigPlanet* shows us protocol in full force, allowing the unprecedented freedom while tightly controlling that freedom.

It is here that we see a limit of protocol. In his *Convergence Culture* (2006), Henry Jenkins notes that we have moved away from passive media consumption to a "participatory culture" where "rather than talking about media producers and consumers as occupying separate roles, we might now see them as participants who interact with each other according to a new set of rules…" (p. 3). Cleary *LittleBigPlanet* falls into this classification. The system it puts in place re-positions users not only in relation to one another, but also to Sony and the developers of the game. However, Jenkins also notes that "Not all participants are created equal. Corporations—and even individuals within corporate media—still exert greater power than any individual consumer or even the aggregate of consumers" (p. 3). In the case of *LittleBigPlanet*, I argue that the distributed network is a guise for corporate control of the system. In this system, the users have become free laborers, performing the developer's work for them. Even though they may have been granted the power to create and share their own levels with anyone else, the ultimate goal of *LittleBigPlanet* is not to foster creativity and unhindered creation, but to create perpetual consumers.

Studying the interactions that take place within *LittleBigPlanet*, Dave Jones (2012)writes "these networks form cultural spaces in which people and technologies assemble to perform coordinated, collaborative knowledge work. Users are now crucial *participants* in the production of information and knowledge in the social web" (p. 243)(emphasis original). This may be the case, but Jones does not recognize the factors that allow—or potentially restrict—this form of participation. *LittleBigPlanet* may signal a shift to increased participation and collaboration by

users. The addition of this social layer of gameplay stands in striking contrast to the conflict found in typical video game fare. But under closer examination, these utopian visions are tempered by severe limitations placed on the users by the system. It is necessary to explore these limitations and their implications for users.

Similarly, most of the scholarship examining *LittleBigPlanet* has largely ignored protocol. Ross, Holmes and Tomlinson (2012), for example, explore the social creation of genre within *LittleBigPlanet 2* without considering the effects and constraints the platform places on the user. Much of their work focuses on how well various user-created levels in *LittleBigPlanet 2* conform to genre expectations, "how user-generated game design exists within a game's semiotic domain." In their conception, users are "performing" a game's genre to various degrees when they design their own levels with the platform provided by the *LittleBigPlanet* platform. To them, players participate in the genre by imitating conventions already established by game developers. In this sense, as the "Main Story" mode of the *LittleBigPlanet* games are presented as platformers, then the majority of user-created levels are platformers. Additionally, some user-created levels may break out of this pattern and explore other genres, Ross, Holmes and Tomlinson assert that these levels, too, rely on creator's and users' prior knowledge of those genres in order to be understood and properly played. While they admit that "the influence of the encompassing game in particular is likely in part due to the affordances of the game's level editor" they do not explore the affect the system and its limitations may have on user-generated level design.

Before moving forward, some historical perspective is needed. The main ideas behind *LittleBigPlanet*—or at least their building blocks—are not especially new. In video game culture, several practice resembling what take place in *LittleBigPlanet* have existed for decades. PC

gamers have been creating "mods"—modifications to video games that affect the base game in some way—since at least the early 1990s. Mods are constructed by altering a game's source code in some way and can range from minor graphical upgrades to completely changing the dynamic of the game with new rules, items or weapons. In some cases, mods create an entirely new game.

In fact, in the heyday of PC gaming, modding actually encouraged by game developers. It was common practice for developers to include the source code for their games as part of the commercial release. The first-person shooter *Doom* (1993), for example, included files that allowed users to modify character designs, sound effects and other aspects of the game. Some mods have become so popular that they have been appropriated by the developers of the original game, one of the earliest and most prominent examples being Valve Corporation's *Counterstrike* (1999), which was originally developed as a mod of the company's *Half-Life*.

Critics have noted that modding is often seen by video game developers as a form of outsourcing in which modders work on – and often improve – a game without receiving any compensation for their efforts (Arakji and Lang, 2007). What's more, modding is not often viewed as a form of labor within the community itself. Dubbing this phenomenon "playbour" Julian Kucklich argues that "the precarious status of modding as a form of unpaid labor is veiled by the perception of modding as a leisure activity, or simply as an extension of play" (2005). Following this line of thinking, others have argued that neoliberal business discourses have become embedded in the modding community (Hong 2013) and that new forms of digital entertainment have continued to disrupt traditional definitions of "work" and "play" (Goggin, 2011).

In their *Games of Empire*, Dyer-Witheford and de Peuter chart the turn to "immaterial labor," labor that is not necessarily concerned with the creation of actual commodities. Rather,

"Immaterial labor is less about the production of things and more about the production of subjectivity, or better, about the way the production of subjectivity and things are in contemporary capitalism deeply intertwined" (Dyer Witheford and de Peuter 4). In the case of modding, then, the immaterial labor of the modders themselves is not just the alterations to the games themselves, but also the idea that such activities are "fun" and therefore do not deserve compensation. The packaged software itself goes a long way to building these subjectivities (Burger-Helmchen and Cohendet, 2011). Many PC games come with toolkits or simply allow users access to the game files and source code, leaving distribution up to the modders themselves. *LittleBigPlanet*, on the other hand, while not as robust as some game editors, rolls everything, creation, distribution and promotion into one package.

While it may seems there is little difference between these mods and what is going on in *LittleBigPlanet*, some important distinctions need to be made. First, the vast majority of mods came on the PC, where it is much easier to work with and manipulate code than with consoles such as the PlayStation 3. Furthermore, even though the source code is available and manipulatable, modding has mostly remained in the background, only of interest to a dedicated subculture. *LittleBigPlanet*, on the other hand, foregrounds the modification process, as the game's tagline "Play. Create. Share" attests. The creation, modification and sharing of different levels is the point of the game.

As opposed to most games that allow for modification, *LittleBigPlanet* is significant because it appears in a rare setting: a home console. The move to the console has implications of its own, the foremost being the limiting of the protocological endeavor that stands at the heart of *LittleBigPlanet*. Users may have a great amount of control over the creation of new levels, but the game and its source code remain the property of Sony and Media Molecule, preventing users

from making modifications outside of those allowed by the game. Any attempt to do so would be a copyright violation. In order to fully examine what this means in terms of protocol and how the game actually operates, we must first take a look at what makes *LittleBigPlanet* a platform.

In their *Racing the Beam* (2009), Nick Montfort and Ian Bogost advocate an approach to studying digital media that will "investigate the relationships between platforms—the hardware and software design of standardized computing systems—and influential creative works that have been produced on those platforms" (p. 2). To them, in order to better understand the use of computers and digital media as an avenue of creative expression, we must consider everything that allows users to participate in their culture through these platforms, "the chips, boards, peripherals, controllers, and other components that make up the hardware of a physical computer system" (p. 2). This definition encompasses hardware such as the PlayStation 3 system itself. But Montfort and Bogost also include operating systems, "whatever the programmer takes for granted when developing, and whatever, from another side, the user is required to have working in order to use particular software" (p. 2) in their definition of a platform. It is from this definition that we can begin discussing *LittleBigPlanet*.

The Nintendo Entertainment System, for example, is a good example of a platform. This platform consists of the hardware of the system itself, its controller (and the limitations put on it by the number and arrangement of its buttons) and the cartridges that contain the actual game data. This is the traditional platform model: In order to play more games, players must buy new cartridges for their system. Similarly, operating systems such as Microsoft Windows can also be considered platforms. Platforms such as Windows may or not include features such as a graphical user interface and are limited by factors such as software compatibility issues and the

types of disc drives equipped on their local PCs. With this in mind, it is time to turn to *LittleBigPlanet* itself.

In this regard, *LittleBigPlanet* is wholly unique from other platforms. In one sense, the game is purely a piece of software that runs on the PlayStation 3 hardware platform. But within the software itself—the code—the game also serves as the vehicle for constructing user-generated content. One of the most significant implications of this is that users are limited to using the PlayStation 3's DualShock 3 controller while designing their own levels. More importantly, as a game exclusive to the PlayStation 3 and a property of Sony, *LittleBigPlanet* content and levels are only available to users connected to the PlayStation Network. When considering *LittleBigPlanet* as a platform, then, we must consider not only the physical hardware of the console itself, but its extension into cyberspace and the limits it places on users there.

This extension includes several important factors, such as the system put in place to allow the sharing of levels and the actual tools supplied for creating the levels. As for the former, Sony and Media Molecule integrated the network into *LittleBigPlanet*'s menu system. One of the options from the game's main title screen includes a "community" option that links the player to all of the user generated content. In this we see one of the major aspects of protocol: The direct connection of users to one another without an appeal to an intermediary. However, this is somewhat misleading. While users are able to share levels with one another, they still must operate within the system put in place for them by Sony. There is no other way to share *LittleBigPlanet* levels. This is just one example of Sony using protocol—or at least something closely resembling it—to its own advantage.

In addition to the system that allows users to share their own levels, we must also consider the methods in which levels are created in *LittleBigPlanet* as part of their platform.

With mods of PC games the platform is much more open. Access to the game's source code, coupled with a multitude of options for distribution over the internet, allows greater freedom for their creation and dissemination. Unlike mods, however, at no point is the user of *LittleBigPlanet* tinkering with the games actual code. Instead, they are simply acting out the range of activities made available to them by the developer. The game cannot, then, be considered a true mod. This distinction is crucial. In the case of actual mods, users are free to create whatever they want, their only limitations being those of the game's engine. On the other hand, in *LittleBigPlanet*, a game that is based upon the promise of unleashing one's dreams and imagination, players are limited to using objects provided for them by the system. While these objects are basic (they range from buttons, switches and string to decorative stickers) and can lead to the creation of any variety of complex mechanisms, users are still limited to the basic game elements and aesthetic. As we will see, this seemingly subtle difference is of the utmost importance when discussing Sony's exploitations of protocol.

In order to explore this further, we must drive at the heart of what the *LittleBigPlanet* is as a game. Jesper Juul's (2005)distinction between progression games and emergence games is of particular use here. In progression games, players complete a series of tasks in order to complete the game (p. 69-72). Emergence games, on the other hand, are games that "exhibit a *basic asymmetry* between the relative simplicity of the game rules and the relative complexity of the actual playing of the game" (p. 73-75). At its core, *LittleBigPlanet* provides a dialectical tension between these two extremes that ultimately expresses itself in the finished, user-created levels. The main content of the game, the platformer levels, tend to be more progressive with some emergent elements. They emphasize progressing in a linear fashion to some ultimate goal.

Players are not necessarily required to make the same exact moves to accomplish this goal, but are limited in almost every other sense.

Indeed, the Story mode in *LittleBigPlanet* (this is how the game terms it) follows the pattern of a progression game. Put in the shoes of a personally customized Sackperson, the player makes their way through themed worlds—ranging from the streets of New York City to the wilds of Africa—while trying to stop "the Collector" an evil being that is stealing other creators inventions and keeping them for himself. In *LittleBigPlanet 2*, the player's Sackperson struggles against the "Negativitron," a vacuum clear that wants to suck up all of "Craft World." The plots of the games are not important and little time will be spent on them here. What is important is that they both point away from disciplinary video game design and to a seemingly protocological, emergent one.

Despite the progressive nature of the main story modes in the *LittleBigPlanet* games, the actual act of creating levels is highly emergent. From a few simple design tools, a wide variety of different levels and games can be produced. Perhaps, then it is more appropriate to describe the game as a platform for emergent design rather than describing its gameplay as emergent. In any case, this is a key distinction to be made. In one regard, *LittleBigPlanet* resists protocol and in the other, it wholly embraces it. Keeping this split in mind, we must examine *LittleBigPlanet* as a system.

Through their gameplay, the story modes of both *LittleBigPlanet* games enact this disciplinary game design while at the same time the stories themselves seem to rail against this very design. This inherent contradiction works in this instance because *LittleBigPlanet* deemphasizes its story in favor while encouraging the act of creation. In this way, the story modes become a method for users to orient themselves in the *LittleBigPlanet* universe before

they strike out on their own. This is not the case with other video games. In most games, the rules and story are closely intertwined, resulting in player behavior that was dictated by the system. In this sense, the rules of the game coerce the player into a certain type of behavior. Therefore, it can be said that, under this system of game design, that almost all games produced are not protocological.

For its part, *LittleBigPlanet* seems to have its feet in both worlds. On the one hand, by turning the platformer into a platform for creation, the games break down the relationship between player and designer. With this centralized model severed, the game begins to resemble Galloway's distributed network, where any user and connect directly with any other user to share levels and games. On the other hand, while users can connect directly with one another, most levels adhere to the same disciplinary design style as other games. Other users are not allowed to modify levels created by other users, for example. They must only play them. So while the powerful design tools offered by *LittleBigPlanet 2* allow users to create levels well beyond the control of Media Molecule, each is saddled with the burden of authority.

To be clearer, the *LittleBigPlanet* games reveal an important characteristic of video games. At the level of coding and rules, video games are inherently anti-protocological. No matter what the situation players find themselves in, they are bound by certain actions available to them. These mainly come "under the hood," at the level of the game's code. *LittleBigPlanet* does not allow users creating their own levels to alter these actions. The actions may change slightly, but creators cannot create their own, nor can they map them to specific buttons on the controller. They must build their levels around them. Users only have a great amount of control over the objects that they place in the environment. This means that in order to create levels in

various genres, creators must rely on these external objects. These objects, then, are where protocol takes its hold in *LittleBigPlanet*.

Returning to Galloway, we have learned that through the process of reification, protocol takes on "authoritarian undertones" (p. 245). Understanding what we do about *LittleBigPlanet* as a platform for the creation of games, we can begin to see this taking place. Running through the PlayStation 3 console, the platform combines both physical and digital elements. The main game itself operates much like a traditional platform. After beating the main game, the player must look to other games if they want to continue playing. However, *LittleBigPlanet* adds a wrinkle to this formula by extending its platform into the online space. It becomes a platform for both playing new levels and a platform on which these levels are built. This may seem innovative, but it is also where we begin to see these "authoritarian undertones" come through.

I have already discussed how *LittleBigPlanet*'s online platform falls under the control of Sony and Media Molecule. But thus far I have not discussed in detail the tools and objects made available to users to create their own levels. For all intents and purposes, *LittleBigPlanet* is based upon the act of creation. The game's opening sequence establishes this idea by implying the game itself is a fantasy world where humans can express their creative energy. The game's playful art style further encourages this idea. Instead of building seamless game worlds that attempt to be indistinguishable from our own, everything in *LittleBigPlanet* looks as though it was made in an arts and crafts class: objects appear to be made of cardboard and construction paper and look as though they are held together by string and glue. Even the game's avatars echo this aesthetic and encourage customization. Dubbed "Sackpeople," they appear as small, burlap sack dolls that can be covered with several different kinds of fabric and further customized with various clothing options, including shirts, pants, and hats.

The endless customization is charming, but it is also where we begin to see Sony influencing protocol. As I have already established, *LittleBigPlanet* presents users with a limited set of tools and items to work with. Instead of having access to the raw materials for creation—the game's code—users can only use building blocks created by the game's programmers. This model allows Sony and Media Molecule to continually create new blocks to sell to their customers. For example, since its initial release, both *LittleBigPlanet* and *LittleBigPlanet 2* have seen the release of several purchasable downloadable content (DLC) packs. These packs give the player new objects, palettes and more to use in building and customizing their levels. These packs are based on Sony—and other's—intellectual property. For example, to date DLC packs have been released based on franchises ranging from Konami's *Metal Gear Solid* to Disney Pixar's *Toy Story*. Because of the limited design tools, users are unable to create these objects on their own.

LittleBigPlanet, then, is essentially a toy box, not the toy factory or toy development studio. The game encourages us to make our dreams reality, but then tells us what those dreams are, puts a price tag on them and tries to sell them back to us. This creates a world in which are dreams are separate from us, created by the culture industry instead of being generated from our minds. This process is given a physical manifestation within *LittleBigPlanet*'s main game in the form of the bubbles that contain objects. Strewn about the levels that come packed with the game, these bubbles exist outside of the manifestation of the player within the game, their sackperson, and must be collected in order for the user/player to have access to the objects they contain while creating their own levels. When collected, these bubbles burst with a popping sound, signifying the successful collection of that object, but also, perhaps, the transition of the object from its ideal state in the imagination to its real state of existence in the world.

This variation in the use and placement of external objects allows users to create a great variety of levels, from recognizable genres to the bizarre. This variety of user-generated content shows the potential of protocol within a game like *LittleBigPlanet*, but also reveals where Sony takes its hold and intervenes in the system. The level *Shoot! Jump! Die?* by user jackofcourse, for example, is a rather traditional platforming level. Players (up to four can play each level at a time attempting to work cooperatively) navigate a series of complex jumping puzzles that also require precision shooting with the paintball gun, an in-game item that can be used for various purposes. In this case, players use the gun to activate various platforms placed throughout the level.

Despite the addition of the paintball gun, *Shoot! Jump! Die?* is clearly recognizable as a platform game with its complex series of jumps and avoidance of various obstacles, including pits in the floor. But jackofcourse was not able to produce a level from scratch. They were not able to choose to use the paintball gun, but instead had to choose from the options presented to them. On the other hand, because the game is highly emergent and allows for a great amount of customization, levels have popped up from a wide range of popular cultural contexts, including those that would normally be forbidden by copyright laws. Users have attempted to recreate various levels from other game franchises, from 80s classics like *Contra* to more contemporary games such as *Dead Space*. Nintendo mainstay Mario even makes an appearance on the Sony-owned property thanks to user-generated content. One such example, *Resident Sack* by user phantommordicai, combines story elements from Capcom's *Resident Evil 4* (2005) with first person shooter mechanics. In the level, the player's Sack person is modeled after Leon Kennedy from that game. The gameplay itself resembles a shooting gallery, with the player looking down the sights of a pistol while blasting zombies. *Resident Sack* exemplifies something only possible

through *LittleBigPlanet* and protocol: A mixing of genres and stories from across a range of stories.

Other user created levels don't resemble traditional levels in any sense. Instead, these levels serve a variety of functions, from telling stories to simply giving away user-created items for others to use in their own levels. These levels do the most to highlight the link between *LittleBigPlanet*, genre and protocol. They highlight the fact that the games do not restrict user created levels to one genre. Furthermore, the tagging and rating system adds to the social element, meaning that the community itself comes to a consensus on how to describe and evaluate each level.

Near the end of his *Protocol*, Galloway makes a distinction about protocol. Comparing protocol to traffic control, he argues that protocol "always operates at the level of desire, at the level of 'what we want,'" adding that anything else is simply "coerced behavior" (p. 241). Viewing video games from this angle, then, it is possible to see the traditional game design model as this "coerced behavior." A path is laid out that the player must not deviate from in order to finish the game. Furthermore, each game comes preloaded with a set of rules dictated to the player that they must abide by. This model represents what Galloway calls "disciplinary societies," a term that he borrows from Michel Foucault.

LittleBigPlanet, on the other hand, taps into our desires. Instead of putting the user in the shoes of an already constructed character, the game allows us to project a personality onto a character and make them our own. In the process of doing this, though, the personalities we choose to give our Sackpeople are reflected only in the clothes we put them in. In doing so, we fall under the sway of the corporations that want us to perpetually consume new objects. These (im?)material objects are the price we pay in order to gain access to this world of (limited)

freedom and play. Early in his book, Galloway describes protocol as a "*contradiction* between two opposing machines: One machine radically distributes control to autonomous locales, the other machine focuses control into rigidly defined hierarchies. The tension between these two machines—a dialectical tension—creates a hospitable environment for protocological control" (8). This perfectly encapsulates the contradictions at the heart of *LittleBigPlanet*. On the surface (and in the game's marketing campaign), it is billed as a space of unhindered creation, where users can share everything with one another free from control by a higher authority. In reality, however, users are subjected to will of Sony and Media Molecule. If they wish to continue creating, they must continue buying.

It would appear that *LittleBigPlanet* has had a major impact on the video gaming landscape. Since its release, games focusing on user-generated content that make use of protocological systems have begun appearing at a rapid pace across a range of genres. *ModNation Racers*, for example, applies *LittleBigPlanet*'s concept to the kart racing genre of games, allowing players to design their own custom karts, tracks and avatars (appropriately, *LittleBigPlanet* entered this arena in November 2012 with the release of *LittleBigPlanet Karting*). Additionally, games such as *Minecraft* and *Terraria* take things a step further in that they make the act of creation the game itself. This shift in focus also represents a shift in the concept of play. Instead of playing the game, players now play with the game. But we should not unquestioningly accept this new form of play as a positive step. As Galloway tells us, protocol can be dangerous. In the case of many of these games, protocol is used as a tool to fuel corporate profits, putting users to work as free labor. This may be just one limit of protocol, but if left unquestioned, it might give us a glimpse of the future.

References

Arakji, Reina and Karl R. Lang. (2007). Digital consumer networks and producer-consumer collaboration: Innovation and product development in the video game industry. *Journal of Management Information Systems, 24* (2), 195-219.

Burger-Helmchen, Thierry and Patrick Cohendet. (2011). User communities and social software in the video game industry. *Long Range Planning, 44*(5-6), 317-343.

Dyer-Witheford, Nick and Greig de Peuter. (2009). *Games of Empire: Global Capitalism and Video Games*. Minneapolis: University of Minnesota Press.

Galloway, Alexander R. (2004). *Protocol: How Control Exists After Decentralization*. Cambridge, MA: The MIT Press.

Goggin, Joyce. (2011). Playbour, farming and leisure. *Ephemera: Theory & Politics in organization 11*(4), 357-368.

Hong, Renyi. (2013). Game modding, prosumerism, and neoliberal labor practices. *International Journal of Communication, 7,* 984-1002.

Jenkins, Henry. (2008). *Convergence Culture: Where Old and New Media Collide*. New York: NYU Press.

Jones, Dave. (2012). Tracing the User Experience of Participation: What is it and Why Does it Matter? *SIGDOC '12*. Paper presented at the 30th ACM International Conference on Design of Communication, 3-5 October 2012, Seattle (243-50). New York: Association for Computing Machinery.

Juul, Jesper. (2005). *Half-Real: Video Games between Real Rules and Fictional Worlds.* Cambridge, MA: The MIT Press.

Kucklich, Julian. (2005). Precarious playbour: Modders and the digital games industry. *The Fibreculture Journal, 5*. http://five.fibreculturejournal.org/fcj-025-precarious-playbour-modders-and-the-digital-games-industry/

Montfort, Nick and Ian Bogost. (2009). *Racing the Beam: The Atari Video Computer System*. Cambridge, MA: The MIT Press.

Sony Computer Entertainment. (2008). *LittleBigPlanet*. PlayStation 3 video game. Guildford, Surrey, UK: Media Molecule.

Sony Computer Entertainment Europe. (2011). *LittleBigPlanet 2*. PlayStation 3 video game. Guildford, Surrey, UK: Media Molecule.

Ross, Joel, Oliver Holmes and Bill Tomlinson. (2012). Playing With Genre: User-Generated Game Design in *LittleBigPlanet 2*. *Technical Report LUCI-2012-003*. Retrieved from <http://luci.ics.uci.edu>