Can "Gaming 2.0" Help Design "Serious Games"? A Comparative Study

Damien Djaouti, Julian Alvarez, Jean-Pierre Jessel
IRIT, University of Toulouse, France
djaouti@irit.fr, julian@ludoscience.com, jessel@irit.fr

Abstract

The "Serious Games" field raises a specific need. People without professional game design skills, such as teachers, corporate trainers, therapists and advertising professionals, request tools that could allow them to create or modify such games. This article will analyze "Gaming 2.0" examples in order to identify tools that could help fulfill this need. Indeed, "Gaming 2.0" is a way for players to create videogame content without skills from the entertainment videogame industry. Can these tools be also used to create "Serious Games"?

To answer this question, we will first define a simple theoretical model of videogames. This model outlines four "game parts" that players can create through "Gaming 2.0"-related tools, and it will be used to provide a comparative analysis of fifteen "Gaming 2.0" examples. From this analysis, insights on the relevance of "Gaming 2.0" for the "Serious Games" field will be drawn.

Keywords: Gaming 2.0, Serious Games, Game Design, Level Design, Player-Generated Content.

1 Introduction

"Serious Games" can be defined as *"Games that do not have entertainment, enjoyment or fun as their primary purpose"* [Chen & Michael 2005]. Since 2002, the field of Serious Games has been constantly growing, and these games are now used in a wide range of professional domains like Healthcare, Education, Corporate, Defense, Advertising... [Sawyer & Smith 2008]. To create such games, designers must embrace a vast amount of knowledge related to the targeted domain. However, this domain-specific knowledge is often so complex that it requires working with a specialist. It may even be simpler to let the specialist design a prototype in order to ensure the quality and relevance of the serious content featured in the game. Moreover, in some domains like education, the design of serious games is used as a teaching method to deal with serious matters. Instead of asking students to do a presentation on a given subject, the teacher asks them to design a game about it. It gives students a strong motivation to do a lot of research: they have to master the subject in order to be able to design a game about it. In a similar way, educators and teachers may want to design games they can use in a classroom as support materials for their lessons [Sauvé 2007].

Sandbox 2010, Los Angeles, California, July 28 – 29, 2010.
© 2010 ACM 978-1-4503-0097-1/10/0007 $10.00

Moreover, as identified by de Buttet [2009], a current trend in the European Serious Game market is to create "kit-based" games instead of "custom-crafted" applications. The main advantage of this new trend is to provide to Institutions and Companies who purchase Serious Games a way to modify them on their own.

To summarize, one side-effect of the rise of "Serious Games" is that some people would like to design such games, but they do not have the required game design skills and tools to do it. While there is no quantitative data yet to support this observation, there are research projects focusing on the creation of "simple game design tools" suited to teachers [Gray & Young 2007; Burgos & al. 2008; Rankin & al. 2009].

Before creating new tools, it may be beneficial to analyze those that already exist outside the serious games field. Among the directions game designers in the entertainment industry are currently exploring is the importance of player-generated content[1]. Following the wave of "Web 2.0" internet sites that allowed users to create and share their own content [O'Reilly 2005], several games let players create or modify in-game content. From the level editors featured in *Little Big Planet* (Media Molecule, 2008) and *Halo 3* (Bungie, 2007) to the object editors from *Spore* (Maxis, 2008) and *Drawn to Life* (5th Cell, 2007), some major console-based games are now incorporating player-generated content. In reference to the "Web 2.0" movement which emphasizes on user-generated content, such games involving player-generated content are labeled as "Gaming 2.0" [Le Roy 2006]. However, such approaches associated with player-generated content are far from being new practices. The computer games market shows a long history of in-game editors and player-created levels, mods and full games. The novelty of "Gaming 2.0" seems to lie in the addition of similar features inside console-based games, especially AAA-level titles[2].

On one hand, "Gaming 2.0" allows players, who can generally be considered as "people without professional game design skills", create game-related content. On the other hand, "Serious Games" are regular videogames designed to serve a serious purpose. Thus, "Gaming 2.0" may be a way for non-professional designers to create "Serious Games"?

In an effort to answer this question, this article will first define a formal model to outline the different "parts" that have to be crafted in order to create a game. We will then analyze a selection of current "Gaming 2.0" examples, and compare how they let players create or modify games : what "parts" can be created and how the creation process is simplified. The objective of this article is to identify aspects from "Gaming 2.0" that seem relevant to simplify the creation of "Serious Games" for non-professional designers.

2 Theoretical framework

In order to study how "Gaming 2.0" simplifies the game design process, we first need to define "game" and "game design." Through this article, a "game" will be understood as the resulting artifact from a process called "game design," as introduced by Salen & Zimmerman [2004]: "*Game design is the process by which a game designer creates a game, to be encountered by a player, from which meaningful play emerges" [p.80]*.

According to Juul [2005], a "game" can be defined as a variable state machine: *"In a literal sense, a game is a state machine: A game is a machine that can be in different states, it responds differently to the same input at different times, it contains input and output functions and definitions of what state and what input will lead to what following state. […] When you play a game, you are interacting with the state machine that is the game." [p.60]*.

Within the framework of these two definitions, several theoretical models detail the different "parts" games are made of. For example, Järvinen [2008] splits a game in nine "elements," Tajè [2007] outlines six "layers," while Elverdam & Aarseth [2007] propose to analyze games through sixteen "dimensions." More generic models examples are the "token" approach detailed in Adams & Morris [2003] and the "patterns" introduced by Björk & Holopainen [2005]. Finally, some models such as Frasca's taxonomy of rules [Frasca 2003] focus on a single "part."

In order to provide a general overview, this study doesn't require such detailed models. According to the aforementioned works, we will define a game as a variable state system composed by four "parts":

- **Initial State**: the starting state of the system, crafted through a process usually called *"level design*."
- **Input:** the means that allow players to provide information to the system.
- **Compute**: the inner mechanics that allow the system to change states, often detailed with rules and elements.
- **Output**: the way the system displays its current state to players, which usually involves the creation of visual and audio assets by artists.

These four game "parts" can be crafted by separate persons, but in the end they will always be tied together to create a "game." We will use this simple model as an analysis grid to review current "Gaming 2.0" approaches. For convenience, this theoretical framework will be referred to as the **ISICO model** *(short for: **I**nitial **S**tate, **I**nput, **C**ompute, **O**uput)*.

3 Mode of inquiry

To perform a qualitative analysis of "Gaming 2.0", we first need to define a corpus of relevant examples. In order to do so, a clarification must be made on the nature of "player-generated content." In his taxonomy of player-generated creations, Tolino [2009] makes a clear distinction between "ludic artifacts" and "game content." The latter category points to any player creation that is confined to the game, such as levels or mods, while the first category addresses any creation that goes beyond the game such as movies using game footage (machinimas), hand-made costumes representing game characters (cosplay) or in-depth game guides (faqs). Tolino's taxonomy solely focuses on such "ludic artifacts." This article will concentrate its efforts on the "game content" category. Through this study, "player-generated content" will then refer to any player creation that fits in the previous definition of a "game."

Moreover, in the wake of the "Web 2.0" movement, it's tempting to label any "Web 2.0" site related to games as "Gaming 2.0". For example, this label could be applied to machinimas, faqs and other ludic artefacts created by players and shared through "Web 2.0" platforms. However, according to our definition of a game, this should be not labeled as "Gaming 2.0" but rather as "Web 2.0 related to gaming." Indeed, most "ludic artefacts" are not "playable" by nature, unlike the game-related content created by players. Nevertheless, this supports the "ability to exchange" as a core value of the "2.0" label. In this sense, games that allow players to create content without letting them exchange it, like *Drawn to Life*, shouldn't be considered as part of "Gaming 2.0." Moreover, available definitions of the "Gaming 2.0" term [Le Roy 2006; Nations 2008] do not necessarily clarify its core characteristics. Therefore, we propose to define "Gaming 2.0" as: any game or application allowing players to create, exchange and play game content.

We then selected a corpus of 15 examples that fit this definition. This corpus gathers two kinds of items: videogames allowing for players modification, and applications that let users create full games. Both allow players to share and play their creations. Using the ISICO model, we will analyze how these examples allow players to design each game "part." We will also review their exchange abilities in order to provide a comparative analysis of current "Gaming 2.0" examples.

4 Comparative analysis of current "Gaming 2.0" examples

Overall, the current examples presented in this article are characterized by two kinds of "tools":

- **Predefined list of choices:** present the player with a series of creative choices defined by the game designer; while easy to use, these tools restrict the player's creativity to the limits of the original game designer's own imagination.
- **Creation tools ("editors"):** a set of simple tools which allow players to create "emergent" content not anticipated by the original game designer; although more complex to use, they offer more creative freedom.

Our examples mix these two kinds of tools in different ways. Some titles intentionally limit the player's creative abilities to some parts of the ISICO model, while others allow creating or modifying all of them. The two tables on the following page detail the examples from our corpus.

Table 1: *Games allowing player modification and modification sharing*

Title	Initial State	Input	Compute	Output	Exchange
Halo 3 (Bungie, 2007) for Xbox 360	Level editor	-	Short list of parameters	-	In-game sharing platform hosting more than 20000 maps and 20000 game-variants[3].
Little Big Planet (Media Molecule, 2008) for PS3 and PSP.	Level Editor	-	-	Import art into levels	In-game sharing platform hosting more than 1 million levels[4].
Spore (Maxis, 2008) for PC.	-	-	-	3D model editor + library	In-game sharing platform hosting 129134373[5] objects (creatures, buildings, vehicles…).
Super Smash Bros. Melee (Nintendo, 2008) for Wii.	Level Editor	-	-	-	In-game sharing platform closed submissions on 06-30-09. IGN's unofficial one hosts 12607 levels[6].
Wario Ware: Do It Yourself (Nintendo, 2009) for Nintendo DS	Level Editor	In rules editor	Rules editor based on conditions & actions + "single sentence" instructions[7]	Art editor + Music Editor[8]	Yet unreleased outside Japan at the time of writing this article, but will allow sharing through WiiWare

Table 2: *Applications allowing players to create and share full games (With the exception of Kodu, games are browser-based and hosted on a website where any visitor can play them.)*

Title	Initial State	Input	Compute	Output	Exchange
Cartoon Network Game Creator (Cartoon Network, 2008-2009)[9]	Level editor	-	Short list of goals available in level editor	-	Hosting more than 6 million games[10].
Gamestar Mechanic (GameLab, 2009)	Level Editor	-	Short list of goals available in level editor	-	In public beta stage at the time of writing this article[11]
Kodu Game Lab (Microsoft, 2009) for Xbox360	Level Editor	In rules editor	Rules editor based on conditions & actions	-	Peer-sharing of games through Xbox Live.
Playcrafter (ZipZap Play, 2008)	Level Editor	-	Short list of goals available in level editor	Import art and audio	Hosting 56002 games[12]
Popfly Game Creator (Microsoft, 2008)	Level Editor	In rules editor	Rules editor based on conditions & actions + source code	Art & audio library + import	Application closed down on 06-24-09[13].
Sims Carnival Wizard (Electronic Arts, 2008)	List of levels	-	Short list of parameters	Short list of visuals	Hosting 15294 games[14] created using one of these three applications.
Sims Carnival Swapper (Electronic Arts, 2008)	-	-	-	Art import	
Sims Carnival Game Creator (Electronic Arts, 2008)	Level Editor	In rules editor	Rules editor based on conditions & actions	Art & audio import	
Sharendipity (5D Research, 2009)	Level Editor	In code editor	Code editor based on conditions & actions + source code	Art & audio import	In public beta stage at the time of writing this article[15]
Sploder (Geoff Gaudreault, 2007)	Level Editor	-	-	-	Hosting 863861 games[16]
Ugengames (MobiTween, 2007)	-	-	-	Art library + import	Hosting 3316 games[17]
WhoseGame (Orange, 2007)	Level Editor	-	Short list of goals available in level editor	Art editor[18] + import	Hosting 1125 games[19]

In order to ease the creation process, designers of these "Gaming 2.0" applications often choose to use a "list of choices" tool instead of an "editor." Most examples seem to combine a level editor to create the "Initial State," a common feature in videogaming culture[20], with lists of choices for the other "parts." However, *The Sims Carnival Wizard* allows players to create full games using only predefined lists; it presents a series of choices to players in order to create a game from scratch. Players first select a game genre (e.g., *Racing*), then a sub-genre (e.g., *Top-Down Racing*) and a visual theme. Another series of questions will allow players to "fine-tune" the game (e.g., pick a goal between *win the race* or *last man standing*; set the *number of laps* and the value of *physics variables*...)[21]. Players can even modify the look of each car, and select a *race track* (i.e., an "Initial State") from a predefined list. Overall, this method is very easy to use, but its creative potential is limited to the "choices" imagined by designers of the application. Indeed, it can even automatically generate games, which are created by letting the computer pick random choices from each list.

Besides level editors, an art editor can be found in some examples. The simplest way is to allow players to import external images files from their computers. For examples, the *Sims Carnival Swapper* allows players to pick any game on the site and replace its art assets with their own images. The resulting games are published as new titles on the site. *Ugengames* is based on the same concept, but unlike *Sims Carnival* it does not offer additional editors to modify other "parts."

Other games feature a "real" art editor. For example, in *WhoseGame*, a 2D drawing program allows players to draw within predefined zones, which will then be animated as parts of characters (e.g., arms, head, body). *Spore* also features an art editor, in which players can create 3D models for any object in the game (e.g., creatures, buildings, vehicles). The *Spore* editor is a very interesting example of how to democratize game content creation without limiting it too much. Professional 3D creation software tools are usually very complex to use, and were obviously not designed to appeal for a large player audience. The actual editor built in this game works like some kind of puzzle: players can create quite complex 3D models by assembling pre-built forms and tuning their size, orientation, and color. As an indicator of its efficiency, *Spore*'s sharing platform hosts nearly 130 million objects designed by players.

Four of our examples also offers a "library" that allow players to select pre-existing art assets to use in their creation. While *Sims Carnival Wizard* is only offering this feature, the three other example complete it with an art editor, thus allowing both people with and without artistic skills to modify or create the "Output part."

The case of "Input" editors is simple: most computer games allow players to configure their inputs; therefore, this feature is hardly provided as a "design tool" in our examples. However, the few "Gaming 2.0" examples who offer a "Compute" editor include the ability to write "rules" associating input commands to actions in the game.

Finally, the case of "Compute" editors is indeed interesting. Only five out of fifteen examples featured such an editor; the others rely on a predefined list of goals that can be selected in the level editor. These five compute editors borrow heavily from tools like *Game Maker* (Mark Overmars, 1999-2008) or *The Games Factory 2* (Clickteam, 2006), which are used by amateur game design communities. Indeed, these tools introduced two easy methods to create "rules":

- The "step-by-step" way starts by asking the player to apply some basic rule templates on elements (such as *race car*, *platform* or *top-down*). The user can then test the game, and whenever a "probable event likely to become a rule" occurs (such as *collision*, *key pressed*), the game stops and asks whether the user would like to create a rule. The creation of this rule is made through the second method described below.
- Replacement of traditional programming languages by a "point & click" approach. The user can manipulate "actions" and "conditions" to create "rules" organized into a giant table, easier to read than a script for beginning game designers. First introduced in *Klik & Play* (Clickteam 1994), this feature was so interesting that several others tools like *Sims Carnival Game Creator, Popfly*, *Game Maker*, *Construct* (Scirra 2008-2009) and *Game Develop* (CompilGames 2009) included similar ones. *Kodu Game Lab* improved this idea for gamepad-controlled interfaces, where players can create rules made of "conditions" and "actions" represented by cartoon icons.

On a side note, *Cartoon Network Game Creator*, though not allowing players to modify the "compute" part, asks players to play and successfully complete their levels before accepting them in the exchange platform. Therefore, some kind of "beta-testing" is done by players before sharing their newly created single-level game.

5 Result

With regard to the comparative analysis presented in this article, our "Gaming 2.0" examples might seem quite limited to fulfil all the needs outlined in the introduction. The design process of "Serious Games" consists in combining a "game" structure with "serious objectives" [Alvarez 2008]. In order to achieve such a result, the designer must be able to create or modify all the four "parts" of the ISICO model. For example, if a teacher wants to modify a retail racing game to teach the Highway Code, the teacher must be able to modify the "compute part." If the only way to modify this part is through a predefined list of choices, the teacher may not be able to include the preferred rules. Thus, to design such a simple example of "Serious Game," non-professional game designers must be provided with tools that allow them to create the "compute" part of a game in a simple way. Among the fifteen examples studied, only four applications presented such a feature.

This type of limitation is not directly related to the design of "Serious Games" but to the design of all types of videogames. "Gaming 2.0" tools were created to ease the design process of entertainment videogames, but even for this goal they seem to suffer from some limitations. As discussed in this article, their ability to create mechanics, story or interface relevant to entertainment games is limited to keep the tools easy to use. Of course, these limitations are also present when someone tries to use a "Gaming 2.0" tool to create a "Serious Game." An additional study on the limitations of "Gaming 2.0" tools to create

entertainment games should then be conducted to provide more data on this topic.

Moreover, these applications borrow heavily from tools used by amateur game designers. Hence, their level of required skills of is often higher than what can be expected from non-professional people who want to create "Serious Games."

Finally, the core tools of these applications were imagined about 15 years ago, when "Serious Games" were not a major concern. They are targeted at the creation of entertainment games only and rely on a videogame culture that some members of the market interested in "Serious Games" do not share. Indeed, the audience targeted by "Gaming 2.0" consists of players rather than professionals with "serious" needs. While this may not be a problem in many cases, we can imagine that some situations will require tools fully addressing the needs of "Serious" game designers. For example, one major concern in education is the ability to evaluate the performance of players, since teachers want to be able to assess the progress of their students. None of our examples propose such a feature, though a skilled designer may be able to create a custom-based evaluation routine with tools featuring a "Compute Editor." However, if the goal is to allow any teacher to create "Serious Games," such a feature should be included in the design tool. The same would go for prerequisites related to other fields, such as the compliance with standard protocols, but the ability to assess the progress of players seems like a cross-domain feature relevant for all "Serious Games."

These observations stress the limits of our current study; the examples we analyzed should be tested with "real-world" situations to evaluate their relevance for the Serious Games field. While further scientific field studies should be conducted that focus on this question, within the scope of this paper we can provide an empirical insight. It comes from a discussion with Julien Llanas, member of the Pedagame research group[22]. This high-school teacher is conducting experiments on the use of videogames in the classroom, and he tested *Little Big Planet* for approximately twelve months. Llanas' current conclusions are that the level editing tool was simple enough to use for both students and teachers. However, his field experimentation highlights some needs related to the "Serious" dimension. For example, students lacked guidance on how to create a "good" level, along with project-planning methodology. In the end, few of the levels created by students were actually "enjoyable." Pedagame's experimentation is still ongoing, and he will concentrate on *Kodu* in 2010 before publishing results. For now, these preliminary observations related to *Little Big Planet* inspire two empirical thoughts: First, tools alone are not enough when used in the classroom, and they should be accompanied by some kind of methodology; this seems to be a logical requirement for targeting any other "non-game design skilled" audience. Furthermore, if *Little Big Planet* featured information on the level design process, it would have been more relevant to be used as a teaching tool; this appears to be a good example of the bias toward "entertainment culture" lying at the core of "Gaming 2.0."

These empirical observations lead us to propose the following hypothesis: "Gaming 2.0" offers interesting ideas on how to simplify the game creation process, but these ideas would be more efficient in software explicitly targeted toward a "Serious Game" related audience. Further steps in our research project will focus on the validation of this hypothesis.

Another major question that hasn't been addressed is how we can train the aforementioned "non-professional game designers" to use such tools. How do we explain the basics of game design to teachers, corporate trainers, advertising professionals and therapists? How can we connect their knowledge to the use of "simple game design tools"? Such questions are starting to be addressed in the education field [Crosbie 2005; McMahon 2009; Westera & al. 2008]. If tools suited to the Serious Game field are designed, could game design skills be taught more easily to non-designers? Additional study on this topic would be required to address this question, but a promising solution seems to be the combination of both: to create a simple game design tool, and to teach people how to use it.

6 Conclusion

In conclusion, we can observe that "Gaming 2.0" is heterogeneous. While all our "Gaming 2.0" examples are defined by the ability to create and share game content, the way such content is created can vary greatly. The comparative analysis provided in this paper is intended to help people who want to create "Serious Games" with simple tools. A selection of examples is detailed in order to highlight respective levels of creative freedom. If needed, the same method of analysis can be applied to any other "Gaming 2.0" software to understand what can or cannot be created with a given tool. However, our examples do not appear to suit all the needs of non-professional game designers who want to address "Serious" topics. More specifically, "Gaming 2.0" offers some interesting ideas related to the simplification of the game design process, but seems to lack awareness of the "Serious" potential of videogames.

The objective of our mid-term research project is to identify or create tools matching the needs and skills of a "Serious Game" related audience. Our next steps will focus on the analysis of other game design tools, such as the professional applications used to create "Serious Games." We will also evaluate some tools presented in this paper through "real-world" Serious Game situations. By combining the ideas found in "Gaming 2.0" examples with applications suited to the creation of "Serious Games," we hope to build a tool that allows non-professional game designers to create such games.

From a general perspective, this topic allows videogames to be used by "serious" domains like education, industry or healthcare. Although several Serious Games already exist in these domains, they are only designed by professional game designers. Offering the ability to create prototypical Serious Games with simple tools to a market studying an area of knowledge but lacking game design skills should allow more diversity in "Serious Games" and bring more attention to them.

References

ADAMS, E., MORRIS, D., 2003. Game Architecture and Design: A New Edition, New Riders.

ALVAREZ, J., DJAOUTI, D., JESSEL, J.P., METHEL G., 2008. Evaluer la cohérence d'un Serious Game. Fifth EDEN Research Workshop, France, Paris, October.

BJÖRK, S., HOLOPAINEN, J., 2005. Patterns in Game Design, Charles River Media.

BURGOS, D., MORENO-GER, P., SIERRA, J.L., FERNÁNDEZ-MANJÓN, B., SPECHT, M., KOPER, ROB., 2008. Building Adaptive Game-Based Learning Resources: The Marriage of IMS Learning Design and <e-Adventure>. In *Simulation & Gaming, 39, pp. 414-431.*

DE BUTTET, S., 2009. Serious Games: modèles économiques. Presentation at the *Serious Games : un nouveau domaine d'application de la recherche ?* conference, France, Toulouse University, October.

CHEN, S., MICHAEL, D., 2005. Serious Games: Games that Educate, Train and Inform. USA, Thomson Course Technology.

CROSBIE, W., 2005. Instructional Design does not equal Game Design - Lessons learned in delivering a course in game design and education. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA) 2005,* Canada, Montreal.

ELVERDAM, C., AARSETH, E., 2007. Game Classification and Game Design: Construction through critical analysis. Games and Culture, 2(1), 3-22.

FRASCA, G., 2003. Simulation versus Narrative: Introduction to Ludology, in *"The Videogame Theory Reader"*, Routledge.

GRAY, O., YOUNG, M., 2007. Video Games: A New Interface for Non-Professional Game Developers. In *ACM International Conference on Computer-Human Interaction (CHI 2007)*, USA : San Jose.

JÄRVINEN, A., 2008. *Games without Frontiers: Theories and Methods for Game Studies and Design*, Phd Thesis, Finland, University of Tampere.

TAJÈ, P., 2007. Gameplay Desconstruction : Elements and Layers, GameCareerGuide. Retrieved 01-03-10 from http://www.gamecareerguide.com/features/355/gameplay_deconstruction_elements_.php

JUUL, J., 2005. Half-Real: Videogames between real rules and fictional worlds, MIT Press.

LE ROY, B., 2006. Gaming 2.0. Retrieved 03-16-10 from http://weblogs.asp.net/bleroy/archive/2006/04/01/Gaming-2.0.aspx

MCMAHON, M., 2009. Using the DODDEL model to teach serious game design to novice designers. In *ASCILITE 2009 conference*, New Zealand : Auckland.

NATIONS, D., 2008. Web 2.0 + Gaming = Spore?. Retrieved 03-16-10 from http://webtrends.about.com/b/2008/12/01/web-20-gaming-spore.htm

O'REILLY, T., 2005. What is Web 2.0. Retrieved 12-05-09 from http://oreilly.com/web2/archive/what-is-web-20.html

RANKIN, J., VARGAS, S., TAYLOR, P., 2009. Testing Metaphorical Educational FPS Games. In *International Journal of Computer Games Technology*, Volume 2009.

SALEN, K., ZIMMERMAN, E., 2004. The Rules of Play, MIT Press.

SAUVÉ, L., 2007. Designing Multiplayer Educational Games with Online Generic Shells. *In The European Conference on Games Based Learning (ECGBL)*, Scotland, Paisley, October

SAWYER, B., SMITH, P., 2008. Serious Game Taxonomy. Presentation at *the Serious Game Summit 2008*, USA, San Francisco, February.

TOLINO, A., 2009. Beyond Play: Analyzing Player-Generated Creations", Gamasutra. Retrieved 01-03-10 from http://www.gamasutra.com/view/feature/4008/beyond_play_analyzing_.php

WESTERA, W., NADOLSKI, R.J., HUMMEL, H.G.K., WOPEREIS, I.G.J. H., 2008. Serious Games for Higher Education: A Framework for Reducing Design Complexity. In *Journal of Computer Assisted Learning*, v24 n5 p420-432.

Notes

[1] http://en.wikipedia.org/wiki/User-generated_content#Player_generated_content

[2] AAA videogames usually refers to high production budgets and major retail sales in the videogame industry.

[3] Retrieved 05-12-09 from http://www.bungie.net/online/ It appears this official exchange platform limits the hosted files to 20000, as the oldest available files are only two months old while the game was released two years ago.

[4] Retrieved 12-05-09 from http://blog.us.playstation.com/2009/07/littlebigplanet-community-reaches-one-million-creations/

[5] Retrieved 12-02-09 from http://www.spore.com/sporepedia

[6] Retrieved 12-05-09 from http://supersmashbros.ign.com/

[7] The core challenge in Wario Ware series is to figure out how to play a collection of "single-movement" games in a few seconds.

[8] These two editors are somewhat reminiscent of *Mario Paint* (Nintendo, 1992).

[9] An online game creator released in three variations inspired by popular cartoons: Ben 10: Alien Force Game Creator (2008), Batman: The Brave and The Bold Game Creator (2009) and Star Wars: The Clone Wars Game Creator (2009). All are browser-based (Flash).

[10] Each game creator featured on http://www.cartoonnetwork.com/games/gamecreators/index.html

restricts play to the 200 “best games”, but according to an official press release 6 million games were created in total : http://news.turner.com/article_display.cfm?article_id=4712 (Retrieved 12-05-09)

[11] Available from http://scholasticawards.gamestarmechanic.com/GSM/web/home.html

[12] Retrieved 12-05-09 from http://www.playcrafter.com/

[13] Retrieved 12-05-09 from http://popflyteam.spaces.live.com/blog/cns!51018025071FD37F!336.entry

[14] Retrieved 12-05-09 from http://www.simscarnival.com/

[15] Available from http://beta.sharendipity.com

[16] Retrieved 12-05-09 from http://www.sploder.com/

[17] Retrieved 12-05-09 from http://www.ugengames.com/ Although all games are modifiable through the built-in output editor, completely original games must be created in Flash by skilled developers and then uploaded to the site.

[18] This editor lets players draw the objects of their game. For a character, they are asked to draw each part separately (arm, body, head…) in front and back views. The game then automatically animates the pieces, much like in *Drawn to Life*

[19] Retrieved 12-07-09 from http://www.whosegame.com/

[20] The first game widely recognized for its built-in level editor was *Lode Runner* (1983). Douglas Smith, who designed *Lode Runner*, developed a level editor in order to create 150 levels for the retail version of his game. At first, he didn’t plan to include this level editor in the retail version of the game, as he only intended to use it as development tool. However, during the course of development, he asked kids in his neighborhood to create some of the 150 levels for him. Noticing the way kids enjoyed to use this simple level editor, he decided to include it with the retail game. This feature greatly contributed to the success of this game.

[21] Observations made on 12-01-09 from the tool available at : http://www.simscarnival.com/wizardtool

[22] http://www.pedagame.com/