Gamification, Serious Games, Ludic Simulation, and other Contentious Categories

Brock Dubbels, G-Scale Game Development and Testing Laboratory, Department of Computing and Software Engineering, McMaster University, Hamilton, ON, Canada

ABSTRACT

This paper provides a conceptual framework for gamification, ludic simulations, and serious games. Central to this framework is the spectrum of design that differentiates work and play. Work and play help define software in purpose as games, productivity software, and entertainment. These categories are informed through cognitive feature analysis of narrative and game play structure. Both can be analyzed to determine the degree of work or play in an activity, as well as issues that influence sustained engagement, which is essential for avoiding game abandonment. To demonstrate the framework for the design and analysis of gamification, ludic simulations, and serious games, several case studies are presented with feature analysis to substantiate the categories.

Keywords: Engagement, Game Design, Gamification, Feature Analysis, Ludic Simulation, Narrative, Serious Games, Simulation

INTRODUCTION

Does gamifying something mean it will be fun, engaging, and compelling? Those familiar with playing and developing games understand there is an inherent problem with this assumption. It assumes that all games are fun and compelling, and that attributes of games can be easily used to increase engagement for work as gamification, ludic simulations, and serious games. This paper provides a framework for knowing how to identify, develop, and differentiate games, play, work, and become knowledgeable for the development of gamification, ludic simulations, and serious games.

This paper is presented in two parts. Part one offers a classification scheme for the software, games, and hybrid categories across a spectrum of work and play. To differentiate, the spectrum of work and play is operationalized using three dimensions for communicating purpose, process and interpretation to the user. These dimensions can help the designer or user determine whether the complexity of a task is given to the user, or handled by the software according to Tesler’s Law of the Conservation of Complexity. The framework is presented as analysis of software, includes structural elements for sustained engagement known as reward-action contingencies, social interdependence, and aesthetics.

In the second section, the framework is used for feature analysis of traditional categories known as models, simulations, and games to compare with hybrid categories known as

DOI: 10.4018/jgcms.2013040101
ludic simulations, gamification, and serious games. These hybrid categories are presented with examples with cognitive feature analysis using the dimensions of the framework. The final section provides a summary of the principles from the framework applied for design, implementation, and reasoning.

BACKGROUND

The idea that one can make an activity engaging by integrating game mechanics ignores the fact that building an engaging game is hard. According to Snow (2011), only 10% of players finished the final mission in the game of the year, Red Dead Redemption. In the article, industry experts shared that: “a blanket expectation is that 90% of players who start your game will never see the end of it unless they watch a clip on YouTube” (Snow, 2011). This fits with Pareto’s Principle, which predicts that 80% of users perform only 20% of the actions (“Pareto principle,” 2013), but it does not bode well for games designed to solve serious problems at work, data collection, in health, care, or academics. An unfinished game may be an incomplete project, partial medical intervention, faulty data set, or an inadequate training.

THE COST OF GAME ABANDONMENT

Building games can be costly. According to the article, “it can take two years for a team of 100 people to create six hours of playable story. At an average burn rate of $10,000 per man month, that’s $24 million just in developer cost” (Snow, 2011). Not all games are this expensive to develop, but it is important to consider that 90% of players did not finish the game of the year; and Red Dead Redemption took over 800 people and nearly six years to complete, with a total cost estimated at approximately $80m-$100 million, making it one of the most expensive games ever developed (“Red Dead Redemption,” 2013).

This is not an indictment of developers for making bad games; it is an acknowledgement that designing a game is difficult. Failed games are very similar to successful games. They share many of the same structural elements that make them potentially fun, but somehow, these games don’t cut it. They do not sustain engagement. When a game is meant for serious applications like work, medical treatment, and training, it is important to sustain engagement until the activity is completed and the outcomes of the game delivered. But if high profile games, struggle with this topic, how can we expect hybrid categories such as gamification, serious games, and ludic simulations to succeed?

It is important to see beyond the potential power of games, and look at their real capabilities. Games are only as good as their design. Computing, software, and the innate capacity for play are compelling and offer great potential. However, once a game is built, it is what it is. If the game is a one-trick pony, that one trick needs to be really effective, especially if it is a really expensive one-trick pony.

MAJOR ISSUES IN DESIGN FOR SOFTWARE

Well-designed objects communicate what they are for, and indicate what the user should do. They create a discourse with the user and imply or direct the user towards the intended purpose. This can be very useful. Software has enabled huge portions of the population to do things that only highly trained specialists could do in the past. Today’s hardware can be programmed to complete complex calculations on the user’s behalf for everything from email, robotic assembly lines, and turn-by-turn navigation. To do this the software developer must create a user interface that helps non-specialist users. The user interface has been essential in making complex behavior available to non-specialists. Just as hammers can direct force, software can lesson task complexity through distribution of the activity between the user and the device (Dubbels, 2011).
A Qualitative Investigation of Gamification: Motivational Factors in Online Gamified Services and Applications
www.igi-global.com/chapter/a-qualitative-investigation-of-gamification/126052?camid=4v1a