

## Chapter 9

# Development of an Educational Video Game That Can Be Modified by End-Users

**Noah L. Schroeder**

*Wright State University, USA*

**Kenneth Deffet**

*Wright State University, USA*

**Alexandrea Oliver**

*Wright State University, USA*

**James Morgan**

*Wright State University, USA*

### ABSTRACT

*The development of educational video games can be a challenging process. Typically, games are created for one content area and are designed to specifically facilitate learning in regards to that content. However, this approach inherently limits the utility of the game to specific content. In this chapter, the authors describe the development and systematic iterative playtesting of an educational game designed to be modified by end-users (e.g., teachers, researchers, students). Quantitative and qualitative data were collected through three phases of beta-testing, and the results from each test informed the subsequent version of the software. Overall, the results indicated that, aside from fixing software glitches, the addition of aesthetically pleasing graphical user interfaces and the integration of sound effects appear to have made the biggest contributions to players' perceptions of the game.*

### INTRODUCTION

Educational games can be operationalized as rule-based, interactive learning experiences in which the player receives feedback while progressing towards a defined goal, and educational games can be played individually, collaboratively, or in competition with others (Wouters & van Oostendorp, 2013, p. 413). Meta-analytical evidence has shown that educational games can facilitate learning (Wouters, van Nimwegen, van Oostendorp, & van der Spek, 2013), and innovations in programming software and game development engines has made creating educational games more accessible as each year passes. However, good game design does not automatically guarantee good design for learning. For example, Kiili (2005) noted that games “should be balanced so that the main determining factor for the success

DOI: 10.4018/978-1-5225-2639-1.ch009

of a player is the player's skill level" (p. 20), but Theodosiou and Karasavvidis (2015) found that new game designers "experienced major difficulties in integrating learning content into the game context and using appropriate mechanics to support learning" (p. 145). These are salient issues, as research has shown that the influence of an educational game on learning can vary depending on the game's design features, such as the learning outcome, the knowledge domain, the age of the target population, and the types of instructional support (Wouters & van Oostendorp, 2013).

Although game development engines have made massive strides towards increasing the accessibility of their use, researchers and practitioners face a variety of development and implementation challenges when trying to create their own educational video games. Most notably, educational games have a tendency to be expensive and time consuming to develop (Moreno-Ger, Burgos, Martínez-Ortiz, Sierra, & Fernández-Manjón, 2008). The development of educational games can require numerous skill sets such as graphic artists, animators, programmers, and instructional designers. To use each of these unique skillsets effectively may require dedicated software programs and high-end computing resources, both of which tend to be expensive. These costs become even more of a barrier to game development when one considers that many educational games have been developed for use in specific knowledge domains, which may limit their reuse and transferability. For example, *Murder on Grimm Isle* is designed to facilitate students' argumentation skills (Dickey, 2011) and *Crystal Island: Uncharted Discovery* is designed to help students understand how to use maps, navigate, and identify different landforms (Lester et al., 2014).

From a practitioner's perspective, educational computer games may be unfeasible because many cannot readily create or modify their own games. The alternative is to rely on games that have been professionally developed for their specific content area, yet there are many fields in which computer games are not currently available. Even if such a game is made available to an instructor, it may not be easily modifiable, which could limit its utility to the instructor. For example, content coverage may be too superficial or too comprehensive, resulting in a game that is not useful unless instructors can tailor the content to their specific needs.

In an effort to bridge the gap between financial investments and practitioner customizability, some researchers have created software that is modifiable by the instructor. Bindoff et al.'s (2014) pharmacy simulation is an example of this type of software. They created a virtual community pharmacy to help train pharmacy students, and instructors can write their own scenarios which are then integrated into the game. Bindoff et al. describe a high level of customizability within the software, including the ability to modify patient details, all dialog options, and the feedback associated with different responses (among other customizable scenario details). However, a limitation of such customizability is the time associated with creating complex scenarios. Bindoff et al. reported that "it takes our clinical pharmacists about 4 hours to write a reasonably complex scenario" (p. 2). It should be noted however, that once a scenario was created it was shareable, and the authors stated that they aimed to create enough scenarios where the software can supplement other teaching techniques in the field across institutions worldwide (Bindoff et al., 2014).

While Bindoff et al.'s (2014) software may be best classified as a simulation, researchers have also begun creating games that are customizable by instructors. For example, Neuneier and Lochmann (2015) developed a smartphone-based game where the game content could be edited by teachers. The content of the game was mathematics, and the game provided an online interface where the teachers could create their own mathematics problems for the students to solve. The game had a two-dimensional interface and worked in a similar fashion as multiple choice questions, with correctly answered questions resulting in

19 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/development-of-an-educational-video-game-that-can-be-modified-by-end-users/183019](http://www.igi-global.com/chapter/development-of-an-educational-video-game-that-can-be-modified-by-end-users/183019)

## Related Content

---

### Modelling Human Activity in People-Oriented Programming with Metamodels

Steve Goschnick, Leon Sterling and Liz Sonenberg (2015). *International Journal of People-Oriented Programming* (pp. 1-24).

[www.irma-international.org/article/modelling-human-activity-in-people-oriented-programming-with-metamodels/163932](http://www.irma-international.org/article/modelling-human-activity-in-people-oriented-programming-with-metamodels/163932)

### End User Types: An Instrument to Clarify Users Based on the User Cube

Chittibabu Govindarajulu and Bay Arinze (2008). *Journal of Organizational and End User Computing* (pp. 61-81).

[www.irma-international.org/article/end-user-types/3841](http://www.irma-international.org/article/end-user-types/3841)

### Educating Our Students in Computer Application Concepts: A Case for Problem-Based Learning

Peter P. Mykytyn (2008). *End-User Computing: Concepts, Methodologies, Tools, and Applications* (pp. 1322-1329).

[www.irma-international.org/chapter/educating-our-students-computer-application/18256](http://www.irma-international.org/chapter/educating-our-students-computer-application/18256)

### VDT Health Hazards: A Guide for End Users and Managers

Carol Clark (2002). *Advanced Topics in End User Computing, Volume 1* (pp. 83-92).

[www.irma-international.org/chapter/vdt-health-hazards/4426](http://www.irma-international.org/chapter/vdt-health-hazards/4426)

### The Technology Acceptance Model: A Meta-Analysis of Empirical Findings

Qingxiong Ma and Liping Liu (2008). *End-User Computing: Concepts, Methodologies, Tools, and Applications* (pp. 1088-1100).

[www.irma-international.org/chapter/technology-acceptance-model/18243](http://www.irma-international.org/chapter/technology-acceptance-model/18243)