# CeRC Story-Game Engine: An open source technology to power story based investigation games

David Farrell, Patty Kostkova, Lisa Lazareck, Dasun Weerasinghe
City eHealth Research Centre, City University London, London, UK

*Abstract:* **Playing computer games is widely popular among children and teenagers as an entertainment activity; however computer games can also be easily transformed into tools for education. City University London's City eHealth Research Centre (CeRC) - has developed such educational computer games to improve young people's understanding of the importance of hand and respiratory hygiene and responsible antibiotic use; and to teach school syllabus concepts such as microbes, the spread and prevention of infection, antibiotic use and antibiotic resistance. We have developed a game platform as an open-source framework to promote game development for education and entertainment. The platform enables the efficient development of new games with new learning objectives along with the ability to support translation into any language. For example, the current CeRC games have been translated into 11 European languages (English, Belgian-French, Flemish, Czech, Danish, French, Greek, Italian, Polish, Portuguese, and Spanish). The existing CeRC games are targeted towards school children; however, we investigate the usability of utilizing such games for adult education as well as examining the effectiveness of these games to convey messages to particular academic, industrial (or otherwise) communities.**

## Introduction

The entertainment value of videogames is long established and videogames are a popular past time activity amongst young people in the developed world [1]. Consequently, there is an interest in using games in an educational environment for this demographic [2]; however there is a lack of successful games based learning (GBL) products available [3]. One of the difficulties facing a GBL designer is the lack of established, cost effective platforms upon which to build and expand their game. As part of the EC-funded e-Bug Project, the City eHealth Research Centre (CeRC) at the City University London (London, UK) developed an open source platform, or Story-Game Engine, which provides a suitable middleware framework from which new games can be developed. This paper describes the "e-Bug Detective Game" that was developed using the Story-Game

Engine, the technology behind the Story-Game Engine, and details of how the framework can be used to develop multilingual games with new learning outcomes.

## The e-Bug Detective Game

The e-Bug Detective Game is a story-based adventure in which the player is charged with solving microbial mysteries by visiting locations, speaking to non-player characters (NPCs) and by searching for clues and evidence. The game-play was designed following focus groups with 15 boys and girls between the ages of 13-15 years (target demographic). The game uses a Problem Based Learning (PBL) approach "in which the problem drives the learning. That is, before students learn some knowledge they are given a problem" [4]. PBL is a pedagogy that is largely used in medical training, and has been considered by Annetta *et al.* as a suitable pedagogy for GBL products [5]. Also, PBL has been shown to be an effective approach to GBL, whereby Chen *et al.* found a statistically significant superiority for medical interns educated (using PBL) in such areas as professional knowledge and clinical techniques [6].

PBL is implemented through game-play via a story scenario that is presented to the player. The scenario assumes that the player does not yet have knowledge of the target Learning Outcome (LO). In the story scenario, there is a problem which is established, but not fully understood or explained by the NPCs. The player must speak to these NPCs to get the appropriate clues in order to find the desired evidence to advance within the story puzzle. For example, to learn about the importance of post-toilet hand hygiene, the designer may first introduce the player to bacteria discovered on a NPC's plate of food whereby said NPC is sick in the toilet, having already visited several locations (such as a barbeque, kitchen, etc.) The player can seek potential sources of contamination, and through a process of elimination, discover that the microbes on the food plate match those microbes found in the ill NPC's bathroom and on the hand of another NPC. By building a mental model of the problem scenario, the player identifies how the gathered evidence can be used to solve the puzzle and, in the process, learns about the desired LOs.

The e-Bug Detective Game has four scenarios or missions, and includes such LOs as "*microbes are found in our food and can transfer to humans*," and "*if your doctor prescribes you antibiotics, it is important to finish the course of medicine*." The e-Bug Detective Game is currently online (http://www.e-bug.eu).

## The Technology Behind the Story-Game Engine

In order to ensure that the e-Bug Detective Game is accessible to as wide an audience as possible, the game was developed using Flash ActionScript CS2 targeting the Flash 8 web browser plug-in, which has a wide penetration (95% of internet enabled machines [7]) and, as such, does not typically require the target user to have administrative rights to install new software to their machine.

The Story-Game Engine is built such that it can power any story without having to make changes to the underlying technology; i.e., the story-game is data driven. This means that when the story-game is accessed, it loads text files which define the characters, rooms, objects, graphics (or art assets), and lines of dialogue used in the playing scenario. All of the puzzle logic is also loaded from these text files; i.e., if a change is required to the story-game logic, it is not necessary to edit the coded software, only the contents of the text files (and a subsequent re-compilation of the story-game).

The text files are created via an Excel spreadsheet (.xls). This spreadsheet has a format that guides the designer - including several tabs (or new spreadsheet pages) for each element of the design. For example, if the designer wants to write a new NPC, s/he accesses the ‘Character’ tab and fills out the fields in a new row. Similarly, if the designer wants to edit or add dialogue, s/he accesses the ‘Conversations’ tab, etc. Finally, each spreadsheet is saved as its own Comma Separated Value (.csv) file for the story-game to load during compilation. Lastly, the web page that embeds the story-game can pass variables to the game to define which scenario is loaded. In this way, the same code can be used in multiple web pages and can show a different scenario each time.

## Facilitating Multilingual Games and New Learning Outcomes

Because the story-game is data driven, it is possible to replace the text files with their translated equivalents that will be accessed by the story-game during compilation. To facilitate this process for the e-Bug Detective Game, at first, the English language version of each scenario was written and tested. Next, the finalised spreadsheet was sent to each partner country for translation. The puzzle logic remains as originally designed; therefore, although the text changes language, it is trivial to switch language files (.xls) based on user preference. For example, the e-Bug Detective Game is implemented in 11 European languages (English, Belgian-French, Flemish, Czech, Danish, French, Greek, Italian, Polish, Portuguese, and Spanish) - with the European Centre of Disease Prevention and Control (ECDC) already in agreement to fund further translations into the other remaining major European languages.

In order to deliver a new Game LO (in medical or other contexts – such as teaching  about gender issues, STDs, sustainability development and agriculture, infection control for hospital staff, etc.), a designer must create a story-based scenario which has a flaw or problem to solve, whilst simultaneously considering the clues, evidence, misinformation (or red herrings), and hints provided.  Similarly to the example provided for the e-Bug Detective Game (harmful microbes on the food plate), any LO is possible to implement within the story-game as long as a logic puzzle can be formulated accordingly.  For instance, several LOs about providing fair trade principles to propose a sustainable development solution for a small town can be incorporated into a story-game investigating the impact of under-priced supermarket products on local farming communities.

## Conclusion

There is great interest in using Games Based Learning (GBL) products in the classroom but one of the difficulties facing GBL practitioners is the cost of developing bespoke software solutions according to each new subject or project.  City eHealth Research Centre (CeRC) has developed a low-cost and re-usable framework for building story-based educational games.  The resulting Story-Game Engine is open-sourced and provides an appropriate technological platform on which to design new games with ease in English or in any other language.

## References

[1] Bengoechea, J I I. “Videogames and Education: a First Empirical Research in the Basqu Country,” Proceedings of the 3rd European Conference on Games Based Learning, pp 195 – 201, 2009.

[2] Williamsin, B. “Computer games, schools, and young people: A report for educators on using games for learning,” Futurelab. 2009, in press.

[3] Kirriemuir, J. McFarlane, A. “Literature Review in Games and Learning,” Futurelab. 2009, in press.

[4] “McMaster Problem Based Learning Program”. Website accessed in January 2010, http://chemeng.mcmaster.ca/pbl/PBL.HTM

[5] Annetta, L A. Cook, M, Schultz, M. “Video Games: A Vehicle for Problem Based Learning.” Technology-Enhanced Education Part II. Springer. Netherlands. 2009.

[6] Chen, Y-C. Fang, J-T.  Lin, J-D. Cherng, W-J. “Evaluation of Problem-Based Learning Education after Clerkship at the Chang Gung University School of Medicine,” Chang Gung Medical Journal, Vol. 25 No. 11. 2007.

[7] 2007 Millward Brown survey carried out for Adobe.com. Website accessed in Jan 2010, http://www.adobe.com/products/player_census/flashplayer/version_penetration.html.