A game engine plug-in for efficient development of investigation mechanics in serious games

CrossMark

Ivan Carmosino, Francesco Bellotti *, Riccardo Berta, Alessandro De Gloria, Nicola Secco

Dept. of Electrical, Electronics and Telecommunication Engineering and Naval Architecture DITEN, University of Genoa, Via Opera Pia 11a, 16145 Genova, Italy

ABSTRACT

Despite their potential, the market for educational serious games (SGs) is still limited, partly because of the high design and production costs. In this paper, we propose a game-engine plug-in for efficient development of a format of investigation in SGs that we designed in order to support discovery-learning in information-rich virtual environments. After describing the framework, the paper presents its application to the development of "Can's Crime", a SG teaching industrial production processes. Data from a preliminary pilot have shown that using the new module allows a significant reduction in development time. Moreover, creation of an abstraction level over the game engine allows improving structured content organization, which is in turn expected to enhance maintainability and modularity. We argue that the same approach could be employed to other game formats, especially in the emerging context of developing games as services.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Educational serious games (SGs) are being used ever more frequently in education [1], training [2] and domains such as sport and health care [3,4]. In spite of the growing interest towards SGs [5–9], their spread has been hindered by several obstacles, including the difficulty of combining by design educational effectiveness and entertainment, the high amount of resources required by the development, also involving different disciplines beside game design and software development, such as pedagogy and domain-specific expertise.

Such complexity is one of the main reasons why margins are little and the SG market still limited. In order to create economies of scale, one idea could consist in identifying some common and fundamental functionalities, and make them available as services in the cloud, according to the Service Oriented Architecture paradigm. These services could be related to a format, which is a model that can be instantiated with different contents in different contexts [10]. Formats create limitations, but, if they are well designed, also provide a good opportunity for combining cost and development efficiency with fun and pedagogical effectiveness.

As an example, which has represented a strong motivation for the present work, open world SGs (OWSGs) [11] are a particular class of SGs in which the players have to explore a 3D virtual environment where information to be learnt is disseminated. The environment could represent human bodies [12], hospitals [13] and geography [14,15], etc. In such games, players are typically asked to perform some tasks in relevant points of the environment, in order to get (by doing) or test knowledge about some specific subjects, according to the Task Based Learning (TBL) theory [16]. Of course, accurate pedagogy is needed, not only in the SG design, but also in its use, as suggested by the latest findings of enhanced discovery learning [17].

In this paper, we propose a development framework for supporting investigation in OWSGs, but also in other types of games. This tool provides a description template that allows developers to define the items to be investigated in a structured and straightforward way. The descriptor is processed at runtime by a game engine plug-in, which is responsible for setting up the virtual environment and managing the gameplay accordingly. The goal is to make the game development easy also for knowledge domain experts with no programming experience, and to separate the content from the code, thus providing a modular work environment for the production, management and maintenance of games.

The remainder of the paper is organized as follows. The next section provides the background and discusses the state of the art. Section 3 describes the supported game model, while Section 4 presents the descriptor format. Section 5 shows the application to the development of the "Can's Crime" SG. Sections 6 and 7 present,

* Corresponding author.
E-mail addresses: ivan.carmosino@gmail.com (I. Carmosino), franz@elios.unige.it (F. Bellotti), berta@elios.unige.it (R. Berta), adg@elios.unige.it (A. De Gloria).

respectively, a preliminary assessment of the benefits and the conclusions of our work.

2. Background and related work

Creating a videogame involves a series of development steps performed by different experts, or teams. The steps include:

- Game design, consisting in the ideation of the mechanics, structures and rules of a game.
- Level design, design of a game map where all the interactive and static elements are positioned in the game world (scene).
- Game programming, including display of the scene (which is usually automatically performed by game engines such as Unity 3D, Unreal Engine 4, Crytek, Lumberyard, etc.), user interaction management, event specification and management, and logic of the game.
- Game art design, which is the artistic process where all the 2D and 3D elements of the game are developed, through modeling tools such as 3ds Max, Maya, Blender, Softimage, Cinema 4D, etc., and photo-editing tools, such as Photoshop, GIMP, etc.
- Audio production, where music and sound effects are created.
- The game under development is also continuously and incrementally tested, in order to find bugs and defects as early as possible in the development process.

New methodologies are needed in order to support design of games able to effectively combine fun and learning [18], covering all the above mentioned points.

Ushaw et al. [19] describes how developers with extensive experience in the entertainment video game industry approach the creation of rehabilitative games. They present a number of design and implementation practices that bring a heightened sense of engagement and replay value to games for patients.

Callele et al. [20] argues that most video games development failures are due to improper requirements engineering, especially because of the associated costs [21]. They conclude that new methods are needed to support the transition from game design document through formal requirements and specifications. Salazar et al. [22] proposes enhancing a game design document from the software engineering requirements perspective. Furuichi et al. [23] proposes a SG design process consisting of eight phases: user needs definition, planning, user needs analysis, system architecture, software design, coding and test, integration and qualification test, deployment and field test, evaluation.

Ibáñez et al. [24] proposes a development methodology based on the concept of Knowledge Centered Approach. The methodology features seven steps: creation of a knowledge base, knowledge characterization, connection of pedagogical and game scenario, creation of the cognitive model, deduction of learning activities, game genre definition, integration of knowledge into the game. Preston [25] presents a scalable and repeatable process to focus participant creativity through constrained design to rapidly prototype games that have the potential for behavioral change intent and knowledge retention among players. SeGAE is a SG authoring environment providing authors a set of editors to adapt a SG to their specific pedagogical needs [26].

Service Oriented Architecture approaches promise to make SG development faster and easier, also enhancing modularity and flexibility. Carvalho et al. [27] presents a case study of a simple SG exploiting a web-service for player's formative assessment. The activity theory-based model for serious games analysis and conceptual design (ATMSG) method has been defined in order to support a systematic and detailed representation of educational SGs, depicting the ways that game elements are connected to each

other throughout the game, and how these elements contribute to the achievement of the desired pedagogical goals [28]. Carvalho et al. [29] studied the application of the ATMSG method for identifying the most suited SG components that could be implemented as services in the cloud.

In recent years, development of video games has been hugely facilitated and enhanced by the use of game engines. Such tools eliminate all the programming effort needed for the graphic and the physics, leaving to the game programmer "just" the implementation of the specific game rules and level design. In the specific field of SGs, IGER is built upon the Panda3D game engine, and provides features tailored to rehabilitation such as a virtual therapist easily pluggable in the game [30]. Tong et al. [31] have shown that game engines can be used to develop serious games for cognitive assessment, cost effectively and without loss of predictive validity with respect to a purpose-built game. The next consequential step is represented by plug-ins that provide higher-level abstractions, for instance for dialogue/conversation management (e.g. [32,33]).

The framework proposed in this paper is positioned in this point of the game development workflow, as it allows an automatic management of the events and of the player choices. The module is suited only to investigation games, where players have to search for clues, select the most promising ones, annotate facts, face trials (typically implemented as mini-games) and finally identify the final solution. Thus, the framework is not a universal solution, but the concept itself is general, and similar modules may be designed for other formats/typologies of games and SGs as well.

3. The supported game model

Based on the requirements coming from various types of customers or stakeholders, we have developed games where the player has to explore a 3D virtual environment representing a city, region, natural area, factory, etc. In the environment, the player has to perform a quest/investigation, either hunting for damages/failures or facing trials and performing tasks such as minigames [11,34–36].

The OWSGs class [35] represents a model that can fit in several different contexts (e.g. exploration of a city's cultural heritage, of an industrial plant and its machinery, of human bodies [12] or of the Oceans [14]). The most didactically-relevant places in the quest environment are called points of interest (POIs). The player has to visit the POIs, play the corresponding tasks and/or collect clues. The tasks are usually implemented as mini games. POIs can be represented by a large variety of things ranging from monuments or historical buildings to pieces of machinery of a factory.

Based on our experience and state of the art analysis (particularly [36–39]), we identified a set of requirements for the OWSG category of educational SGs. The main requirements are the following:

1. Support to exploration of a virtual environment where the player can perform meaningful educational activities.
2. Possibility of inserting in the scene point of interests (POIs) containing objects to be investigated or tasks to be performed.
3. Possibility of describing the context (e.g., story and rules) in which the player acts and of the possible issues to face.
4. Possibility of inserting in the scene explanations about actions a player may take.
5. Possibility of inserting in the scene dialogues based on a tree hierarchy scheme through the interaction with non player characters.
6. Possibility of inserting in the scene help texts for players to familiarize with the game.
7. Possibility of exploiting audio contents.
8. Separation between data and code.