Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 186 (2017) 76 – 83

Procedia Engineering

www.elsevier.com/locate/procedia

XVIII International Conference on Water Distribution Systems Analysis, WDSA2016

Serious Game Approach to Water Distribution System Design and Rehabilitation Problems

Mark S. Morley, Mehdi Khoury, Dragan A. Savić*

Centre for Water Systems, University of Exeter, Devon, United Kingdom

Abstract

This paper presents an online, web-based Serious Game developed to investigate end-user behaviour when faced with complex WDS design and rehabilitation problems. SeGWADE (Serious Game for WDS Analysis, Design & Evaluation) couples an innovative and visually attractive interactive front-end with a server-side modelling engine handling real-time hydraulic simulation.

A multiplayer online game infrastructure is implemented allowing interactions with a model to be instantly broadcast to both collaborating and competing users. The interactions are recorded and can be subsequently be replayed or analysed through the interface to explore the decision making process in detail. The Serious Game engine is designed to be extensible and can be reconfigured dynamically through an interface allowing the upload of EPANET[1] input files and the parameterisation of different game scenarios permitting a range of games to be developed independently. In addition, the front-end employs adaptive graphics that can switch between HTML5 canvas or WebGL rendering technologies, depending on the client hardware capabilities, in order to deliver the best user experience.

The serious game has been evaluated through a classroom-based exercise in which 20 students competed to obtain the best WDS design solution. The results obtained demonstrate a high degree of player engagement with the game.

Peer-review under responsibility of the organizing committee of the XVIII International Conference on Water Distribution Systems

Keywords: Water Distribution System; Serious Game

* Corresponding author. Tel.: +44-1392-723637; fax: +44-1392-217965.
E-mail address: D.Savic@exeter.ac.uk

Peer-review under responsibility of the organizing committee of the XVIII International Conference on Water Distribution Systems
doi:10.1016/j.proeng.2017.03.213

1. Introduction

Serious games are an emerging type of cognitive tools designed to train and teach in an entertaining and interactive manner [2,3,4]. Although serious games have already been used in numerous domains, only a handful of computerised game simulations have been applied to the water sector [5]. These games tend to focus on Water resource management, with a strong emphasis on the development of awareness and understanding of problems and trade-offs. Present games such as Aqua Republica [6], CauxOperation [7], the Climate Game [8], EMOVER [9], Marine Spatial Planning Challenge [10], Shariva [11], and the UVA Bay Game [12] usually involve multiple stakeholders cooperating to resolve conflicting priorities in the context of resource scarcity. This paper presents a basic game that allows a number of users to take part in a competitive game to optimize the performance of a water distribution system. WDS optimization can be applied to many facets of the topology or operation of the network. The topological design and layout of the system can be optimized with a view to minimizing the cost of a layout [13] whilst meeting some minimum design constraint – typically the provision of sufficient pressure to each demand node in the network. The costs of pumping water through a system are a key consideration in network design and pumping arrangements [14] and schedules [15] can be optimized to minimize energy consumption and to ensure that off-peak electricity tariffs are exploited accordingly.

1. Methodology

The SeGWADE (Serious Game for WDS Analysis, Design & Evaluation) game structure owes much to other multiplayer online games. As such, it has a modular architecture where clients (the players using a web browser from their machine) visit a web site (the server), log in, and play the game collaboratively.

The first application of this game architecture is to the rehabilitation of an existing WDS by the optional installation of duplicate pipes in order to alleviate a pressure deficiency. The principal activity of the game is, therefore, to select the diameters of the duplicate pipes from a predetermined list of commercially available pipe sizes, the cost of which increases as a function of diameter. The user can select diameters for each available pipe interactively and then submit the resulting model for hydraulic simulation at which point the resulting pressures are evaluated to determine whether the model meets the minimum pressure requirements. When a new “best” (i.e. lowest cost) solution is identified in the session of online gamers, the game banner on each of the currently connected clients is updated to show the identity of the player and the score that they have achieved – adding a competitive element to the exercise. At the end of the game, it is possible to summon a high-score table on each of the players’ browsers to display their relative performance.

2. Architecture

The game engine is designed using a well-known MVC (Model-View-Controller) Client/Server paradigm with a few modifications so as to make it slightly more efficient for the purposes of gaming over the internet, as seen in Figure 1. As in a standard Model-View-Controller pattern, the *model* organises the application from the point of view of the data, logic and rules, independently from the user interface. The *view* has the user interface in which the information is visually represented. The *Controller*, acts as an intermediary that takes input from the user and converts it to commands for the *model* or *view*. In order to reduce the communications overheads and load on the web server, the MVC loop is short-circuited by using the *angular.js* [16] library so that relatively trivial operations, such as input data validation, are performed on the client-side and not sent to the server-side *model*.

The game engine part of the application is written entirely in JavaScript on both server and client side. This provides an unmatched flexibility from a development point of view. The *Node.js* [17] and *Socket.io* [18] libraries are used for connectivity to handle instant messaging between clients and server during game time. *Sails* [19] being the most popular MVC (Model–view–controller) framework for *Node.js,* is being used for deployment. The code is hosted as open source on GITHub at *https://github.com/gentr1/DWS_serious_game*.