IndustrySim: Finding the Fun in Industrial Simulations

Ilmari Lahti
Turku Game Lab
Turku University of Applied Sciences and University of Turku
Turku, Finland
ilmari.l@sci.fi

Tomas Rosin, Pekka Qvist, Ville Vuorela
IndustrySim Ltd.
Turku, Finland
tomas.rosin@industysim.com
pekka.qvist@industrysim.com
ville.vuorela@industrysim.com

Mika Luimula
Business, ICT and Life Sciences
Turku University of Applied Sciences
Turku, Finland
mika.luimula@turkuamk.fi

Jouni Smed
Department of Information Technology
University of Turku
Turku, Finland
jouni.smed@utu.fi

***Abstract*—This paper introduces *IndustrySim*, a game attempting to combine truly engaging gameplay with state-of-the-art industrial simulation that would suit education and casual gaming alike. We present the concept of the game and then discuss challenges and insights encountered so far regarding the asset management, programming and game design behind its implementation.**

Keywords—serious games; industrial simulation; game design

I. Introduction

Industrial simulations have traditionally been a vital part of production planning and control [1]. In production planning, we aim at establishing strategies for producing products so that the manufacturing resources are used efficiently. In production control, we want to ensure a proper implementation of the production plan despite the occurrence of random events. Shortly put, production planning is about anticipating future events by conjuring and following a plan, whereas production control is about reacting to events as they occur during the production. To realize a software system for production planning we have to establish an interaction between the computer running simulations and the human supervising the planning process [2].

In this paper, we aim at extending the range of use for industrial simulations by presenting a game, *IndustrySim*, that allows the players to design, construct, try out, test and run different industrial processes such as power plants, manufacturing plants, or water and sewage plants (see Fig. 1). The game is structured into three different levels – trainee, advanced and expert – matching the player's skills and competence. The levels also determine how demanding the game is: The trainee level provides a more entertaining gaming experience with only light simulation components, while the expert level satisfies even "hardcore" simulation gamers. Apart from entertainment, the game also suits for educational uses or training. One of the objectives in the game development is to extend the game concept so that it will have links to real-world artefacts. For example, a combination of virtual and real-world environments can open new possibilities to educate experts in the kind of scenarios that they are not able to experience in the traditional learning environments.

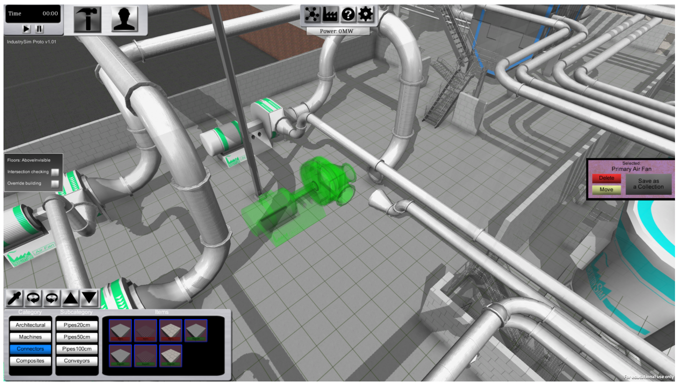

Fig. 1. Screenshot from a prototype version of *IndustrySim* showing a detail of a coal plant in "build" mode.

Game industry has not yet focused so much on game concepts that are close to the technology industry. Simulation-based games do not offer content for players who have a deeper understanding of technology industry, nor do they combine virtual and real-world activities. Even the European Union Horizon 2020 ICT programme tries to encourage the game industry to apply digital games and gamification mechanics in non-leisure contexts [3]. The technology industry is already using human-machine interfaces and 3D models for simulations, interaction systems and even virtual characters but without an efficient use of real-time graphics engines. For instance, Galambos & Baranyi [4] present the Virtual Collaboration Arena (VirCA) where engineers and researchers from different countries can collaborate and test robotics together in a semi-virtual manner. We need more such examples of industrial virtual environments efficiently utilizing digital gaming technologies and including features from both education and real-world conditions.

This type of game design and development requires multidisciplinary research and development activities comprising areas such as instrumentation, environmental engineering, energy engineering, computer science, graphical design, computer-aided design, and game programming. In our case, the development team comprises professionals coming from different fields of expertise but sharing an interest to create a game portraying industrial plants. The team members share an extensive experience in working with information technology, industry and industry automation and automated measurements, and on the other hand in creating educational games, simulation-like learning environments and working with schools, universities and the academia. Modelling a whole industrial plant is a big undertaking, but we have been bringing into our team the best professionals of all the relevant areas, industrial IT and engineering, game development and educational understanding.

The plan of the paper is as follows: We begin with an introduction of the game idea behind *IndustrySim*. After that we will look at the development process from four perspectives: simulation, assets, programming, and game design.

II. The Game Idea of IndustrySim

The *IndustrySim* game draws inspiration from successful games that have featured a "tycoon-like" approach into building your own economic empire. However, our focus is on building an actual industrial plant, or several ones, that will work and behave very much like their real-world counterparts. The deep sandbox of the underlying simulation might be satisfying for some, but the game will also include a progressive challenge-based "career mode", where the players are presented a campaign of scenarios that will be increasingly more difficult the further the players advances on their journey.

The game is developed to work on multiple platforms, such as mobile smartphones, tablets and computers, which all have their distinctive advantages when it comes to styles of playing and enjoying the game. Multiplaying also plays an important part, since teamwork is essential in many educational exercises. The game offers both co-operative and competitive playing styles for multiple players. Utilizing social multiplayer options helps us to link together the achievements and standings of the players and to form virtual industries and corporations of multiple players.

A virtual economy supplements the game and simulation. The players can work together, buying and selling resources, getting earnings from their successful industrial operation based on the efficiency of the plants running in the virtual world (e.g., emissions will cost as well) – and eventually become the top tycoon of the game world.

IndustrySim will not only be a fun game to play, but the fundamental core of the game logic will run on detailed computational simulations that model the operations and processes of a real-world industrial plant. Depending on the player's skill level, parts of the simulation can be switched off, allowing the player to learn the game on easier levels, or just enjoy the entertaining aspects of the game without having to partake in a full-blown simulation.

The simulation will help the beginner level users to design and construct the plant and to run the different functions (e.g., mills, fans, pumps) of the plant. The goal is to prepare, by picking up skills, the player to step up onto a higher level. The career game will start on this level of the simulation, allowing the player to go through several tutorial levels and progress with their skills naturally, and be prepared for more challenging missions and challenges on the advanced level.

Players of the advanced level will make advanced technical designs of plants and also need to operate them well to progress in the campaign. The advanced level players can also compete against other players, in the multiplayer portion of the game, trying to beat each other in areas such as plant design and operation.

The expert level players are equal with the advanced level players when it comes to the features of the technical design and operation of the plants. The main difference is that the expert level players also have economic issues involved. This together with the full, unfettered simulation model in the background will make the game a true and interesting challenge.

III. Simulation

When simulating industrial processes for a computer game such as *IndustrySim*, totally new challenges will emerge compared to simulations done in the industry so far. A computer game is traditionally built to host different scenarios and events that the players need to go through. These game scenarios and events create numerous situations in which the traditional simulation techniques provide poor results. Hence, simulating industrial processes in the game need to be built with another approach. One of the challenges for the process simulation of the game is that the process simulation needs to go, frequently, outside the normal boundaries, and it needs to cope with unusual events as well. This fact, in combination with inherent instabilities involved in iterative algorithms and the need to minimize processing overhead, requires relatively much optimization and improvements to the calculation algorithms used. It should also be stressed that the simulation results must nevertheless always stay inside the scope of accuracy as well.

We have chosen to run the simulation programs on cloud servers, and the data exchange between the simulation programs and the Unity3D part will take place through TCP/IP sockets. The simulation software is written in C/C++, which enables efficient execution of the relatively heavy calculations, many of which are of iterative character. Other benefits with keeping the process simulations on cloud servers is that it allows smooth updating of the software, as well as easy integration of third party software (e.g., an industrial equipment manufacturer) to the simulation. The synergy between the process simulations used by different clients can be optimized when they run inside the same sphere of resources. For example, the result of some heavy simulation (e.g., flow calculation) can be cached in a database and easily utilized by other clients, which would save a significant amount of valuable processor time. The process simulation needs to give fast responses to the player. This challenge requires adaptive elements in the mathematical models as well.

Every significant piece of process equipment (module) is simulated in detail. The different modules are connected together, with the help of connection matrixes, to form a simulation of a whole process. The modules can also be run as single tasks and for many modules, the mathematical models have been extended to also work outside the area of normal operation of a real world industrial plant.

IV. ASSETS

In our previous studies [5], we have explored the conversion of a technical 3D model designed using SolidWorks, a parametric 3D mechanics design software, into an optimized format that suits better for real-time presentation in an interactive game application developed in the Unity3D game engine. We have introduced and discussed 3D modelling techniques, modelling programs, and the tools and features of those programs that can be used for building a simplified version of an extremely detailed technical model. In addition, we have explored factors that are important for real-time 3D graphics presentation, and how models made with computer aided design (CAD) need to be altered in order for them to meet the required standards. See Fig. 2 for an outline of the 3D modelling process.

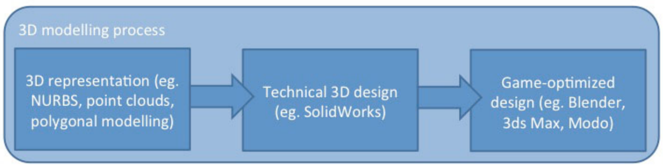

Fig. 2. Phases of the 3D modelling process.

Based on these studies, it seems that we can achieve an optimized model with 20 times reduced detail compared to the original technical model, using standard modelling and texturing tools in the Luxology Modo 3D modelling application. The performance of the optimized model is compared to the original using the profiling tools included in the Unity3D game development kit. The conclusion, based on the results and the efficiency of the method, is that a manual modelling method is more suitable for this purpose than the use of automated optimization tools. In addition, some of the technical 3D models need to be generated from laser scanned point clouds, which is described in more detail in [6].

V. PROGRAMMING

As mentioned in Section III, there are several potentially conflicting factors affecting the architectural design of a simulator game like this. On one hand, the simulation should be as accurate and realistic as possible in order to preserve the educational viability of the game. On the other hand, the game should run smoothly and responsively from the point-of-view of the player, as is expected from any state-of-the-art game. Other considerations include (but are not limited to) preventing cheating by overly industrious students or seriously competitive casual players, allowing a teacher or another supervisor to monitor and control a particular game through a separate device, and ensuring that game instances will be stable and reliable as well as easily accessible despite their cloud-based nature.

To keep these goals attainable and the whole project practically manageable, all the constituent components of the system will be made independent and decoupled from any particular instance of a component. These components are, for example, a simulation server, a game server, and the server running the virtual economy. The simulation server will only run the actual background simulation, and it will only communicate directly with the game server, not the clients. The game server will handle the upkeep of the more concrete game world, as well as the communication with the clients. It will present the clients with the results of the simulation and, after a validation process, relay their responses back to it. The server will also make sure that the state of the game instance is always saved into a database with sufficient frequency.

A "server of servers" is responsible for keeping track of all the other component instances available. When an administrator (e.g., a teacher) or a player requests an instance, the head server will handle the necessary matchmaking between instances and the potential initialization of new ones. Because of the independent nature of the components, and also to enable easy integration with external services and data, all communication between these separate server components consists of JSON data. JSON, or JavaScript Object Notation, is a simple, lightweight data serialization format. JSON libraries are readily available for most programming languages and frameworks that might be utilized, such as .NET/Mono and Unity in the case of *IndustrySim*. Using JSON as the serialization standard makes it straightforward to add entirely new components into the whole system; for example, a web server could easily be made to display simulation data in the form of a web page without any further changes in the system.

VI. GAME DESIGN

The game design of *IndustrySim* faces the old conundrum regarding educational games: how to present real-world information and challenges in an entertaining manner. The solution can be found right next to the problem, in the reality. The player is tasked with the construction, running and improving of an industrial plant. Just like in the real world, the board and shareholders expect a certain level of performance and incentivize the plant manager with bonuses if those expectations are exceeded. In addition, the player may set himself his own goals, like achieving a higher rating in eco-friendliness, becoming a forerunner in the implementation of the latest technology and so on. Some of these additional goals are incentivized by game mechanics and built-in achievement systems. Others are mainly for personalization, enabling you to run your factory the way you want it to be run.

The game presents both low- and high-intensity challenges to the player and determines his performance from his responses to these challenges. All the challenges are taken from real-life events, or they are at the very least possible in the operation of such a plant from an engineering point of view. Obviously, the challenges facing an oil refinery would be different from the challenges facing a modern coal plant and this is where the co-operation and expertise of industry partners becomes invaluable. In the game mode, the challenges are determined more or less randomly, with some additional mechanics to ensure variety and sufficient recovery times from losses and damages. When *IndustrySim* is used as an educational tool, game administrators and teachers can design, browse, modify and trigger various challenges on their users and then monitor their actions as they try to resolve them.

Low-intensity challenges are things like market price and demand category fluctuations, which necessitate adjustments to the processes and provide education opportunities on how altering the different variables in the system affect other parts of the system. Ideally the game is fluid, always benefiting from some adjustments, and while the efficiency changes can be small, they accumulate over time. Wear and tear on the components and general maintenance is also part of the low-intensity challenge system.

VII. Concluding Remarks

In this paper, we presented the idea and realization of the *IndustrySim* game, which aims at extending the use of industrial simulations outside the conventional production planning and control. The development comprises four areas: the actual industrial simulation, the visual assets based on real-world industrial machines, programming that weaves simulation and assets together, and game design providing game mechanics and interaction models.

The central challenge with a project such as this has turned out to be that traditional methods and workflows for making simulations and games are not directly applicable, and similarly the people working in these fields tend to come from very different professional cultures. Especially the understanding of the digital product process and an efficient use of technical 3D models is one of the main challenges when developing these types of games. Graphical assets need to be realistic and accurate, but lightweight enough to run on consumer hardware. The exact same thing applies to the simulation code – in order not to sacrifice either realism or smooth running, a distributed architecture is employed. The greatest of the challenges lies in making the game itself engaging and approachable yet potentially educating and informative. The prototype version of *IndustrySim* (see Fig. 3) has enabled us to take the first steps towards reaching these goals.

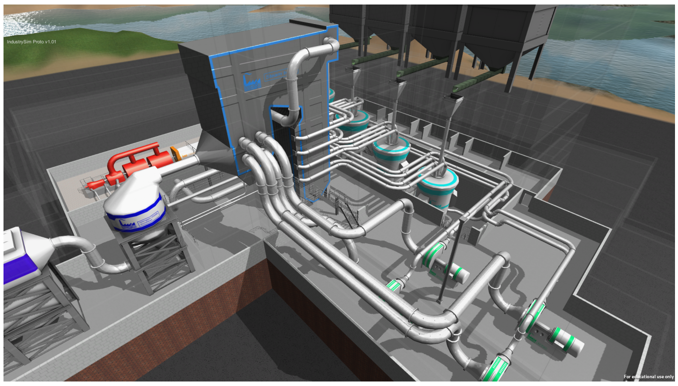

Fig. 3. Screenshot (with the GUI disabled) of an example coal-fired power plant built inside the *IndustrySim* prototype.

Acknowledgement

The authors wish to thank the folks at the Turku Game Lab who have provided us with enthusiastic feedback, observations and comments to help us develop the system further.

References

[1] W. Bolton, Production Planning & Control, Essex, UK: Longman Scientific & Technical, 1994.

[2] J. Smed, Production Planning in Printed Circuit Board Assembly, PhD thesis, University of Turku, Turku, Finland, Jan. 2002.

[3] European Commission, Horizon 2020 – Work Programme 2014-2015, LEIT – Information and Communication Technologies, European Commission Decision C (2013)8631 of 10 December 2013.

[4] P. Galambos, and P. Baranyi. "VirCA as virtual intelligent space for RT-middleware", in Proceedings of the International Conference on Advanced Intelligent Mechatronics, 2011, pp. 140–145

[5] J. Susi, Optimization of a Technical 3D-model for Real-time Graphics Engines, BSc thesis, Turku University of Applied Sciences, Turku, Finland, 2013.

[6] J. Jämsä, M. Luimula, S. Pieskä, V. Brax, O. Saukko, and P. Verronen. "Indoor positioning with laser scanned models in the metal industry", in Proceedings of the International IEEE Conference on Ubiquitous Positioning, Indoor Navigation and Location-Based Service, 2010. DOI: 10.1109/UPINLBS.2010.5653563