A Serious Game to Teach Modeling, Operating and Maintenance of Industrial Plants

Mauro M. Barbat, Nilo C. S. Dutra Junior, Diana F. Adamatti* and Adriano V. Werhli
Centro de Ciências Computacionais
Universidade Federal do Rio Grande
Av. Italia, s/n km 08 – Rio Grande - Brasil
*Email: dianaada {at} gmail.com

Abstract—**With the technological advancement and the level of immersion that technology represents in our lives, new teaching methodologies must be developed. The use of serious games in education gains more space among researchers and educators, because of their immersive and playful power, facilitating the cognition of basic and advanced concepts. In this context, the focus of this paper is the development of an educational serious game to be used with high school and first years undergraduate students. It deals with elementary topics on modeling, operation and maintenance of industrial plants.**

Keywords-serious games; industrial plants: modeling

I. INTRODUCTION

The Artificial Intelligence (AI) applied in electronic games is known as Games AI (Game Artificial Intelligence), and its main purpose is to create a behavior which seems intelligent due to a scenario with multiple choices [12][14] [15].

This type of game should be similar to the human-being behavior. It should make it possible to include personality, the capacity of making mistakes and provide different levels of difficulty to the employer. These characteristics add in experience and immersion in the game and improve its gameplay [13][16].

Even though in the game industry AI has been used since its beginning, when it was known as gameplay programming, its full usage is still a challenge mainly due to the following reasons: development period, learning algorithms testing and performance [3].

Nowadays, serious games have a high acceptability because they provide, among others, immersion power to players. The player is able to be inserted in a context, analyze problems and make decisions according to his/her own judgment and knowledge. The simulation is one way to test ideas and concepts with small cost and/or risks [1].

In this context, the use of serious games occurs in different research areas in order to validate its use as a support to conventional and professional teaching tool. Positive results are obtained concerning to assimilation, acceptance and acquisition of new knowledge [2].

In general, the most acceptable explanation is that serious games are more intuitive knowledge transfer, based in entertainment, where the learning tends to converge and the player learns in an enjoyable way.

In this paper, the focus of the development of serious game involves high school students and early years undergraduates of automation engineering. The idea is to learn the fundamental concepts about modeling, operation and maintenance of industrial plants, which are essential in the industrial environment. The main goal is to approach these topics to students and to motivate them to future careers in engineering.

Industrial plants can be generally described as a set of machines, equipments, sensors and actuators that together are employed for the manufacture of one or more items. The representation of processes as well as the equipment involved in the modeling of industrial plants is widely supported by international standards to standardize these representations in order to use these documents at all levels of manufacturing.

The paper is structured as following: in Section II are presented the main ideas about games and their development. Section III presents the concepts and standards about industrial plants. Section IV presents our serious game, the game design techniques, the first graphical interfaces and results. Section V presents the conclusion and future works.

II. COMPUTER GAMES

Nowadays, we are experiencing a reality filled by the presence of electronic games, which not only attract children and youth but all ages through different contexts. With the current technology, the games are present in many devices such as computers, consoles, websites, mobile phones, tablets, among others.

Analyzing this situation it is noted that this niche market is gradually gaining ground and respect, boosting the gaming industry that keeps growing rapidly, becoming one of the most profitable inside the big entertainment industry as Fig. 1.

Fig. 1 - Money spent on games in Brazil in 2011 [17].

An interesting fact is that games accompany the players throughout life, demonstrating the power to engage and keep the player interested. Due to its immersive nature, the computer games allow players to overcome the physical limits and dip into other realities, subject to experimentation with new ideas and concepts, providing gamers a environment to simulate and interact with innovative and challenging situations that drive the player to experiment and try new ideas in order to receive rewards or simply watch the unfolding story.

A. Computer Games Development

The computer games development process is a complex task due to the interdisciplinary nature involving all stages of development.

Currently, the development of a modern and commercially successful game demands a mixed team with different skills in order to supply the minimum requirements expect by the player.

The development process (Fig. 2) requires studies and considerations that should be made at all stages, from research to the deployment for the target audience and distribution, so as to provide the highest quality and acceptability level for the public.

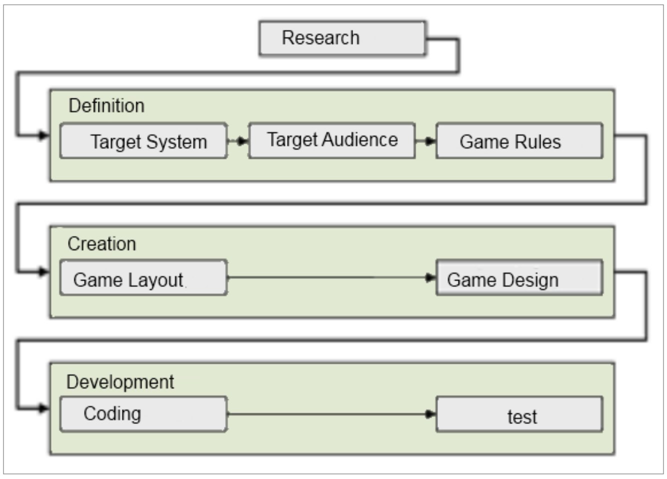

Fig. 2 – Game Development Process [4].

The game development process can be defined by three main stages:

1. **Definition:** At the game design process, all hardware and software requirements are defined. Moreover, the main behaviors and rules necessary are specified to direct the game to the expected audience.

2. **Creation:** definition of art and design in the manner of interaction between the player and the game (gameplay) process.

3. **Development:** Encoding of defined ideas and testing phases process.

B. Game Engines

Generically, a game engine is a game development environment that contains tools and reusable high level mechanisms to common tasks, in this way aiding the developer to improving his yield rate. In Figure 3 you can see the main components of the architecture of a GE tooling.

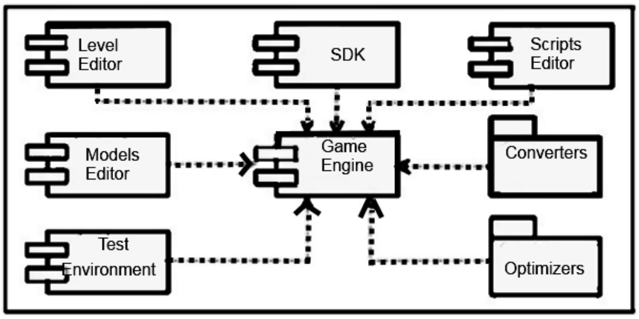

Fig. 3 - Main components of the architecture tooling of a GE [6].

The feature that a game engine can offer will depend on the type of game and focus for which GE was designed. If it was designed for 2D games, must own resources for design allowing the manipulation of layers, images and sprites animation, collision detection, techniques to overlay drawings among others .

If designed for 3D games must support positioning and handling of the camera, positioning of objects in space, physical resources, rendering and more.

The GEs are very useful and widely used in the games industry, offering more flexible ways to build games. However, the use of GEs limits the possibilities for the game construction, making them generic, difficults the possibilities to reach beyond the models offered by the game engine. As result of this a high number of similar games are created.

C. Final Considerations

Electronic games are a reality and are present in the lives of many young people and adults. They create a broad and attractive market around the games using them for various purposes. With this, the need to monitor this growing market emerges, providing developers with new mechanisms and enabling them to use advanced techniques. In this way developers can keep up with demand from players who seek new and more engaging games that surprise with new forms of gameplay, better graphics, and above all, to providing the expected entertainment.

III. Industrial Plants

The industrial Plants concept can be explained as a system composed by different mechanisms involved to produce of one or more items, the correct representation and use should be a common and essential practice in industrial environments. Their use and formal concepts to representation are supported by international standards that provide a high degree of use in all levels of production, as evidenced by the components so they can be prescribed appropriate ways of developing system, providing better strategic use of resources and possible analyzes as [5]:

I. **Technical Feasibility:** Studies about efforts, material resistance and new raw material.
II. **Economic Feasibility:** Manufacturing costs, sales and profit margin.
III. **Aesthetics:** Studies about the project shape and possibilities to show functional and nonfunctional requirements.
IV. **Ergonomics Product and Production:** Features, ease of acceptance.
V. **Environment Impact:** Able to explain the environment resources needs.

In general, every engineer and technician inserted in industrial processes should have knowledge of industrial plants and their rules, because the negligence or ignorance can reverberate flaws and defects in industrial projects. Currently, there are different tools that can assist in the development of industrial plants, as *Smart Plant*[1], PDS[2] (*Plant Design System*) and PDMS[3] (*Plant Design Management System*).

A. Industrial Plants Representation

The different processes will be represented by industrial plants are modeled independently, due to the complexity and the need for accuracy. These processes are represented by Process Flow Diagrams (PFD) (Fig. 4), which are preliminary drawings made by the designers of process.

These drawings should contain the main elements of the system and the engineering flow diagrams, consisting of process flow diagrams that add greater detail of existing elements and also detail all the control loop, equipment, machinery, pipes and primary, secondary and auxiliary resources. Figure 5 is an example of flowchart in standard engineering *Piping and Instrument Diagram* (P & ID).

1 http://www.sisgraph.com.br/smartplant/
2 http://www.intergraph.com/products/ppm/pds/
3 http://www.aveva.com/plant

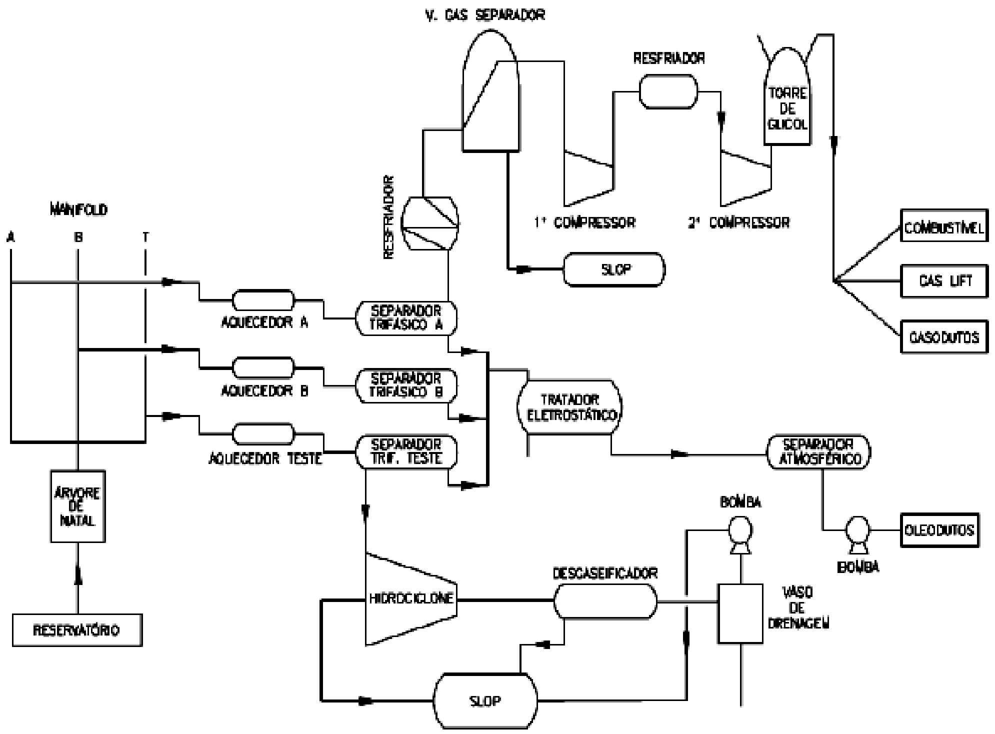

Fig. 4 - General process flow diagram of a floating production unit [7].

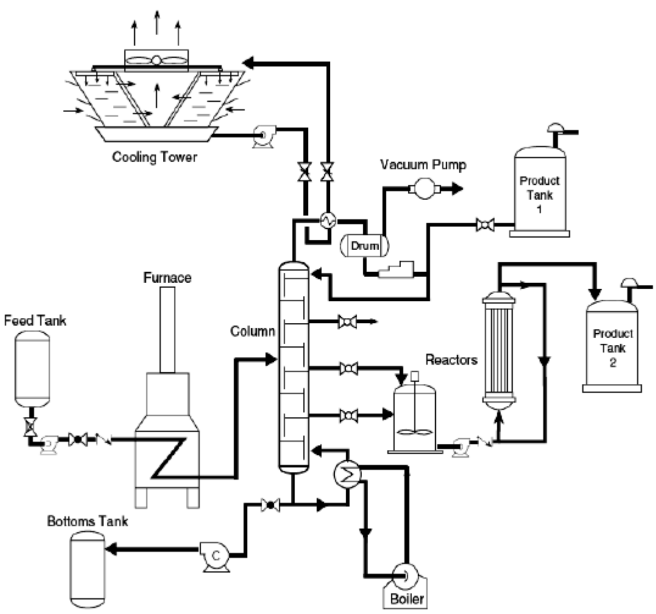

Fig. 5 - Example flowchart engineering P&ID [7].

The representation of equipment, valves and instruments, obey pre-established rules and conventions. The correct use of symbols (Fig. 6) is essential for the correct interpretation of the processes involved. All the symbolism is standardized by the regulatory agencies e.g. ISA and ABNT.

Generally, this notation is used in conjunction with the representation of the equipment and processes forming the document entitled *Process and Instrument diagram* (P&I, Fig. 7), which is used in this serious game proposed.

B. Finals Considerations

The representation and proper use of industrial plants are essential in the manufacture process. These proper representations allows the realization of preliminary analyzes during the time of plant's project, providing better utilization of resources, fault detection, overall impacts and aggregates as well as technical feasibility. Avoiding waste of resources, damage to production and possible design flaws are propagated to the subsequent stages.

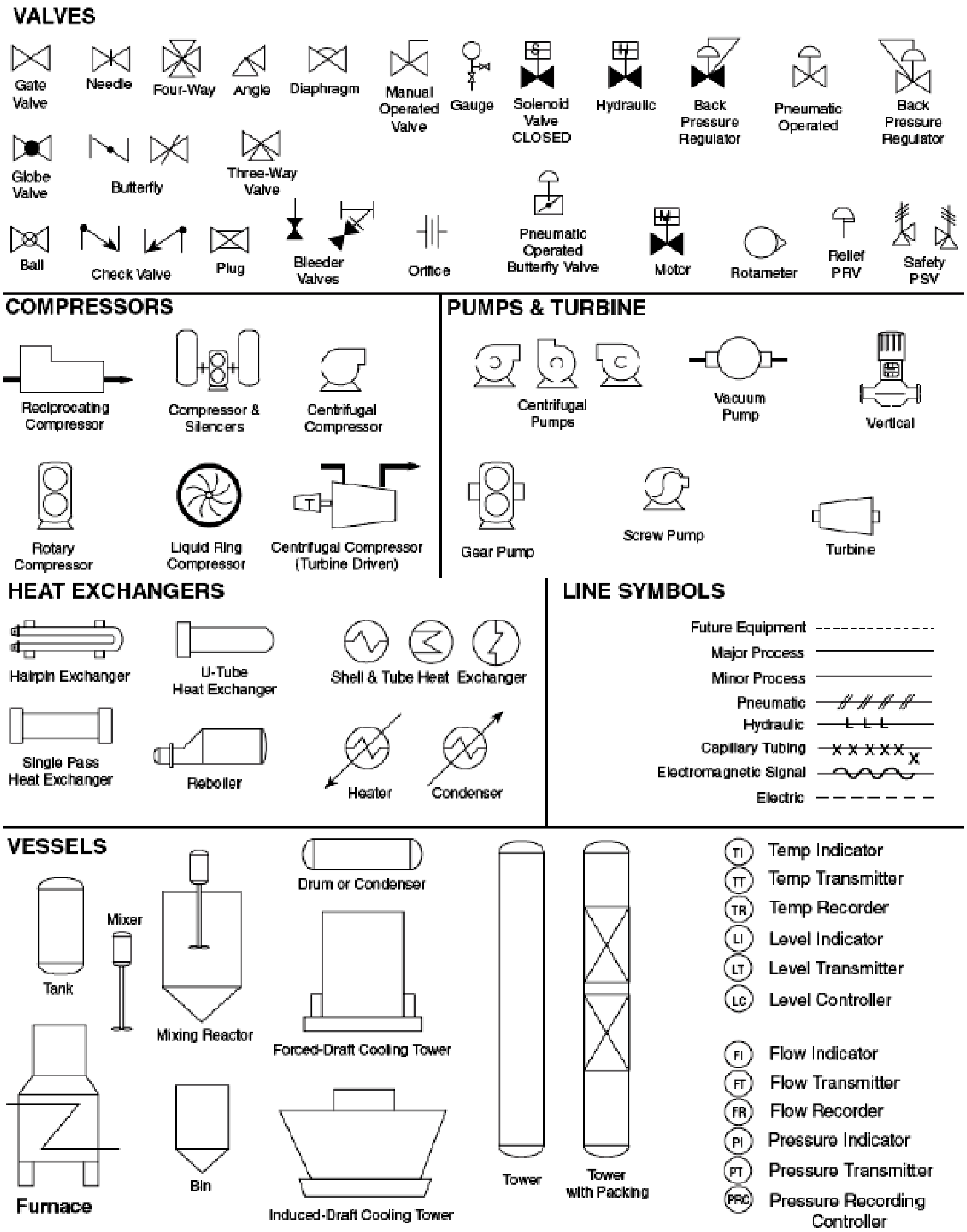

Fig. 6 – Basic instruments symbology [8].

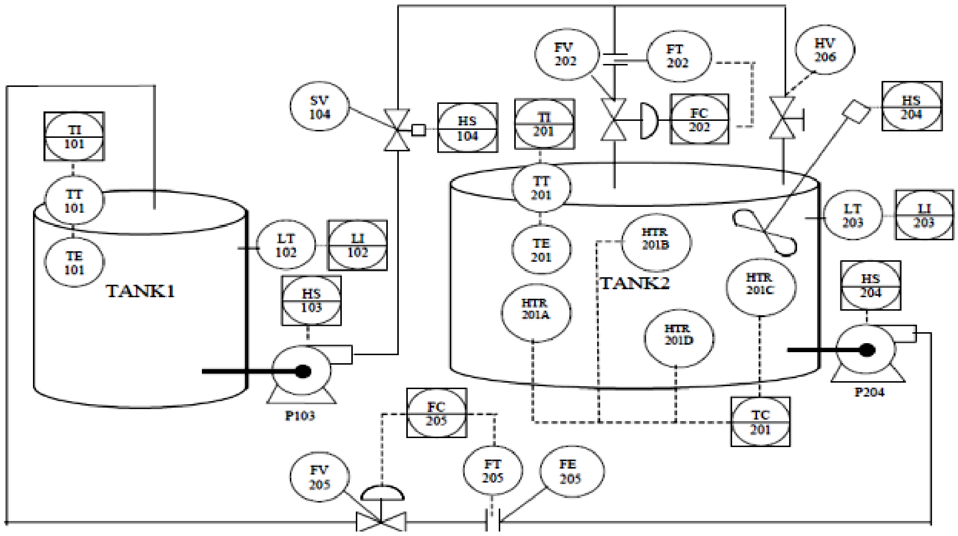

Fig. 7 – Example of a P&I diagram [7].

IV. SERIOUS GAME PROPOSED

The process of development presented in this paper has been organized based on software engineering and game development techniques. Thus, the coding process is structured and the basic platform designed to contain game engine basic features, focusing on maintenance and software updates so that the game can receive new components over time without the need to restructure main code.

A. Software Engineering Application

According to [9], software engineering is a discipline whose focus is on all aspects of software production from the early stages of system specification to maintenance when the system is already in use.

Software engineering works to integrate knowledge about computer science and the needs of the customer. According to [10], the role of software engineering in relation to computer science, is not to investigate hardware designs or prove theorems about the functionality and efficiency of algorithms, but use them as tools for solving problems (Fig. 8).

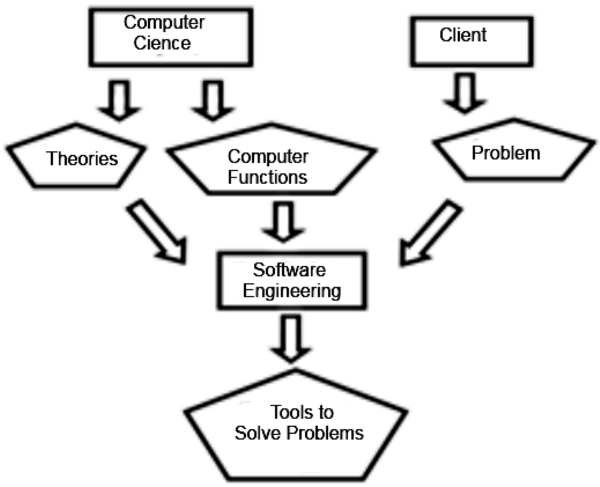

Fig. 8 - Relationship between Software Engineering and Computer Science. Adapted from [10].

The development process used in this paper, was elaborated according to five key layers of software engineering: requirements engineering, systems engineering, project engineering, agile methods and testing techniques [11].

Each layer is applied in different levels and distinct stages, as can be seen in Figure 9.

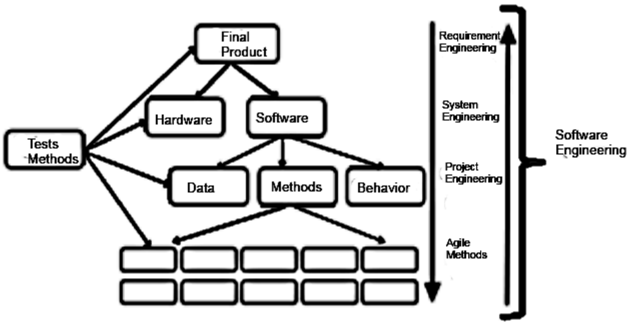

Fig. 9 - Software Engineering Layers. Adapted from [11].

The software engineering layers can be freely defined as:

- **Requirements Engineering :** Survey and understanding of customer needs , is a delicate step because based on the problem that the best solution will be chosen .

- **Systems Engineering :** To be submitted to development process a system must be understood and related to the environment in which it will reside and operate, as technical needs, relationship with users , connectivity with the environment, among others .

- **Project Engineering :** Is started as soon as the first interactions between requirements engineering and systems engineering are completed . The overall objective is to model and establish a set of principles that will lead to the development of quality software in structural level , addressing key concepts that will provide the basic structure necessary for the correct construction of the product.

- **Agile Methods :** The agile development methods are techniques that seek develop dynamic concepts to execute software projects, focusing on the client and the product itself .

- **Testing Techniques :** Technologies and techniques used to provide a higher system reliability degree, these techniques can and should be used at all stages of development.

B. Game Design

As previously stated, the primary purpose of the serious game proposed and developed in this work is to encourage students from high school and college to develop knowledge about industrial plants in a playful manner.

Therefore, it was necessary to design a way to keep the game process enjoyable, exposing a context based on the reality found in industrial environments, so that the player feel encouraged to develop the creativity needed to solve recurrent problems of these means.

The game was designed (Fig. 10) to be a Puzzle, where the player to solve a specific problem, can use Drag and Drop components to create a industrial plant.

The game play consists of three distinct stages in which the player / student will be inserted:

1. Initially the player comes into contact with a proposed problem getting information about the problem and the requirements needed to satisfy it.

2. Following, the player will have time to develop an industrial plant reliant on the types of components and the amount of available components, so that satisfies the problem in the best way possible.

3. Finally, the player must run the simulation and analyze if the created project meets what was requested, according the score, the player may have to go back to step preparation or follow to a next level.

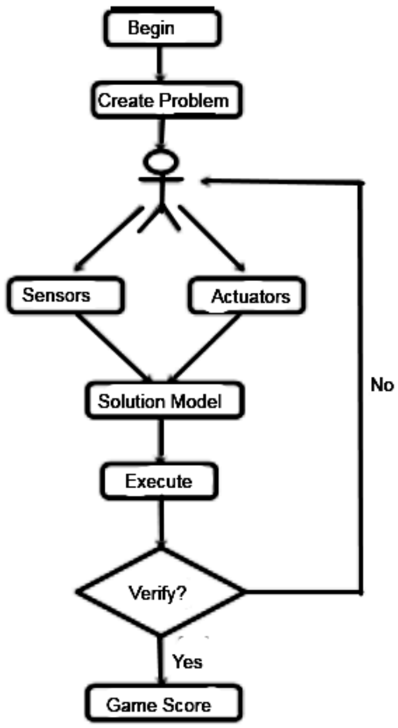

Fig. 10 – Gameplay flowchart.

Aiming to serve a large number of students, educators and schools, the need for requirements engineering process has become essential due to the lack of any well-defined computing infrastructure in schools.

Thus, the development for specific platforms like Windows or Linux became not viable. Ir is necessary the development to multiplatform and also to make possible future migration to the web, making the Java programming language a good choice to meet these needs.

With that, the game should run properly on any system that the Java virtual machine can be installed. In the structure of the project we considered techniques that seek the performance gain for both the modeling stage as for execution.

The player must be able to interact, relate and configure the components offered generating a functional dynamic system. All system components must provide relevant information about the system status that is being developed at run time (Fig. 11).

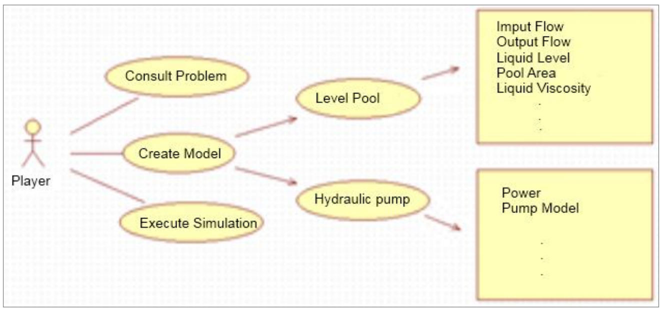

Fig. 11 - Interaction between players and components.

During the modeling process, P&ID representation is utilized due to the fact that it is an international standard and is widespread in industrial modeling representation. Thus, it offers to players, in a playful manner, knowledge about this representation. Therefore, it is expected that the student will be able to recognize the main components and interactions from a real industrial plant.

The system was designed giving attention to modularity and functional independence (Fig. 12). All components should be responsible for controlling their own internal states, but the system as a whole depends on the interaction of different components.

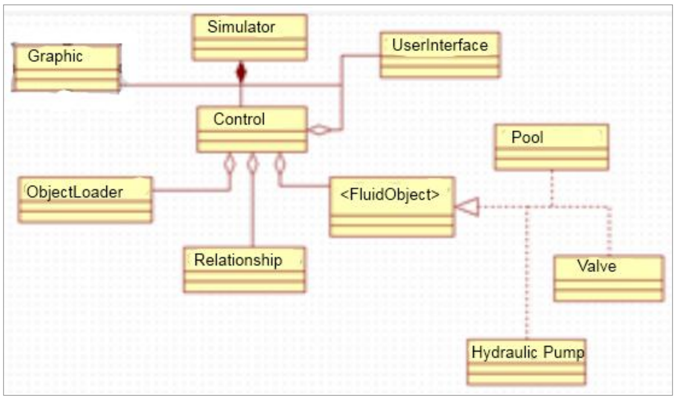

Fig. 12 – Control Class.

Because of this, there is a control class responsible to linking communication between different objects, without their knowledge about these connections.

Thus, it is capable to control the transfer of information between different modules in the system. This class also plays others relevant functions to flow control system, this model is based on MVC (Model View Controller) design pattern, which focuses on the separation between structures responsible for functional models, stages of visualization and control.

These and other abstractions were defined during the project in order to enhance features such as modularity, functional independence and extensibility. Thus, it is expected that over time new features to the game can be inserted without the need to know the entire system.

C. Simulation

In order to ensure maximum fidelity between dynamic processes and visual simulation, it was necessary to establish a set of environmental rules aimed at keeping the metric relations of the components presented.

The player can determine the basic characteristics of its components such as height, width , diameter and volume. And those characteristics should be reflected not only visually and conceptually.

During the execution phase, the time unit used is second (s) , which is obtained through the control class that holds possession of the Thread object . Thus, the delay can be set at run time, if the player wishes to speed up the process.

Dynamic models (Fig. 13) used must be abstracted , aiming not to overload the player / student, enabling it to have a learning experience simple and fluid gameplay, observing the phenomenon passively by observing visual simulations.

However, the player will have the option to meet the employees models also follow the numerical simulation, if he desires.

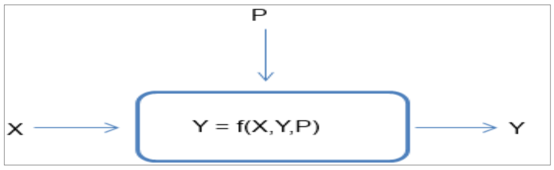

Fig. 13 - Simplified Math Model of a Process [5].

Where:

X = Set of input variables

Y = Set of output variables

P = System parameters including boundary conditions

The whole dynamic process is encapsulated in the object itself. The object only needs to know what kind of object it is connected, for example, if a tank level is connected to another tank level, a hydraulic pump or a valve. This information is obtained transparently at runtime through connectors objects such as pipes, which are responsible for transmitting this information to both objects connected to it.

Thus, the response of each object depends on the type of connection. For example, if the system is configured to interconnect two tanks of similar structure level (Fig. 14) the result is a mass balance between them.

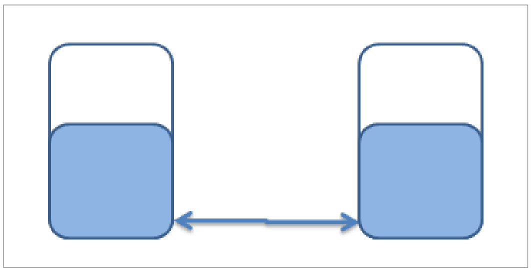

Fig. 14 - Example of coupling two tanks level.

What happens in this case is a relative flow described as:

The total flow in each tank (Qt) is given by the sum of the negative flow (Qn), other words, leaving less the sum of positive flow (Qp) coming.

In this case:

$$Qt = \sum Qn - \sum Qp \quad (1)$$

However, in this case, the negative flow is described in relation with the pressure given by the tank level height and the size of outlet port that is given by the cross sectional area of the fluid connector pipe area.

Therefore, according to [5], the following relationship can be obtained by an adaptation of the Bernoulli equation for the case of laminar outflows and incompressible fluid.

$$Qn = srqt((2 * g * h) / 1 - (d/D)^4) \quad (2)$$

Where:

g = Gravity Acceleration;

h = Liquid Level;

d = Output section diameter;

D = Tank diameter.

Thus, it is possible to see that Positive flow in both tanks is given by the summation of the flows of all tanks connected to it.

If the tank is informed by the connector that is part of a pump-tank system (Fig. 15), the flow is not obtained in a manner analogous to that shown previously. In this configuration, the flow in and out of the tanks depend on the properties of the hydraulic pump.

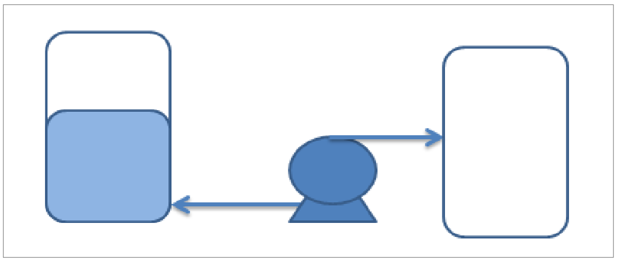

Fig. 15 - Example of System-Tank-Pump-Tank.

It can be observed that the dynamic processes belonging to objects used are encapsulated in the own objects, without external interference, the objects only require basic information to provide a context for painting and timing.

First Results

In Fig. 16 are presented the first step of the game: the description of the problem to solve, as well as the requirements that must be used to obtain the solution of the problem. This step is inserted in the context of each phase. In this way, each phase can present a different problem and a specific graphical interface.

V. CONCLUSIONS AND FURTHER WORKS

The development of tools that encourage young people to learn and understand about the world phenomena is vital. These tools can provide more effective and stimulating the cognition process, taking advantage of technologies that are present in the our daily lives.

The application of software engineering techniques is required to the development of new software, because it provides mechanisms to meet a higher level of efficiency and quality in the development process and the final product. In this way, we will have quality software and reliability. For this reason, in this paper, we have a great attention on the techniques of software development, in order to generate the final result a platform that is easily extensible. The initial tests of the game are being done as well as to define the steps of the game.

As further works, we will finalize the game with 3 steps, where each one will have incremental level of difficulty and test it with high school students, to better understand the ideas and problems to join to careers in engineering.

ACKNOWLEDGMENT

This project is supported by CNPq (Brazilian Council Research), grant number 455400/2012-1.

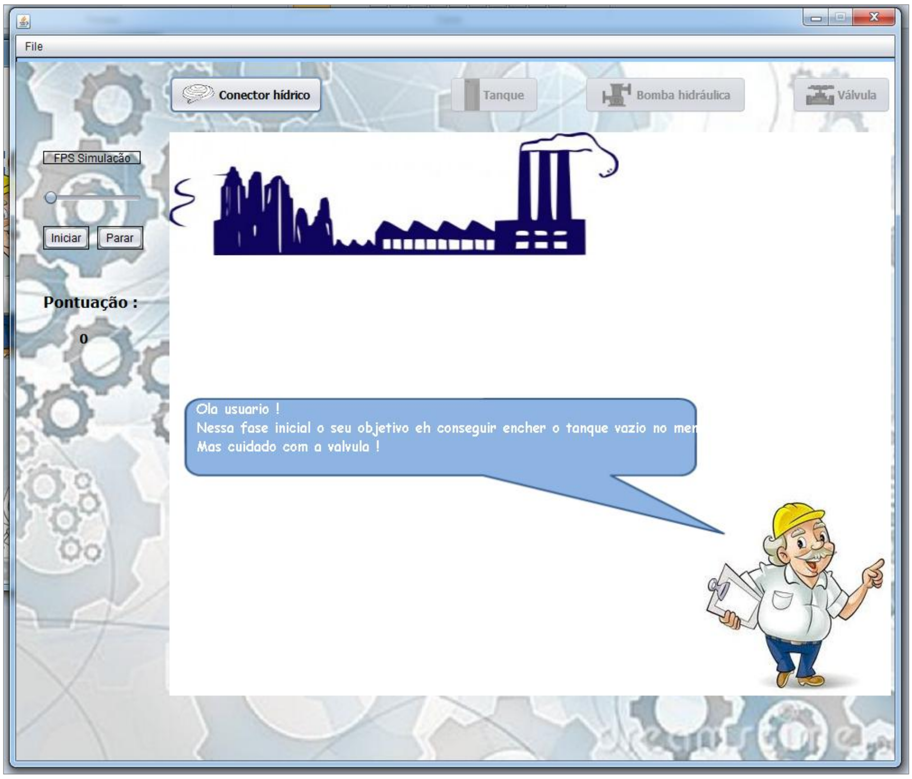

Fig. 16 – Description Problem to Step 1.

Fig. 17 presents the execution of the industrial plant proposed to solve the problem, using different equipment and configurations.

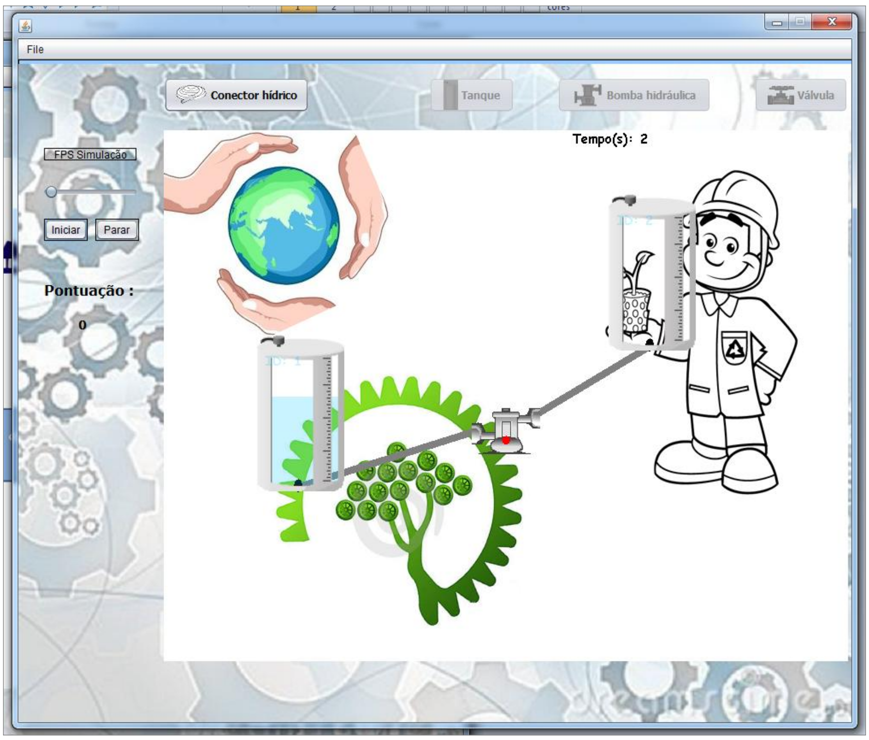

Fig. 17 – The execution of Problem to Step 1.

REFERENCES

[1] D. P. Martinelli. *A utilização dos jogos de empresas no ensino de administração*. Dissertação (Mestrado em Administração) – Departamento de Administração da FEA/USP. São Paulo: Universidade de São Paulo, 1987 (In Portuguese).

[2] J. Alvarez; O. Rampnouxo. *Serious Game: Just a question of posture?, in Artificial & Ambient Intelligence*, AISB'07, Newcastle, UK, April 2007, p.420 to 423

[3] D. Bourg, G. Seemann. AI for Game Developers, OReilly ISBN 0596005555 (2004)

[4] S. D. Amazonas. *Desenvolvimento de Jogos 3D em Java com a Utilização do Motor Gráfico Irrlicht.* Trabalho de Conclusão de Curso, 2007 (In Portuguese).

[5] C. Garcia. Modelagem e Simulação de Processos Industriais e de Sistemas Eletromecânicos. 2ª ed. Ver. E ampl., 1. Reimp. – São Paulo: Editora da Universidade de São Paulo, 2009 (In Portuguese).

[6] B. Feijo, P. Pagliosa, E. Clua. *Visualização, Simulação e Games*. Anais da XXIV Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, pp. 1313-1356, Campo Grande, Brasil, 2006 (In Portuguese).

[7] T.R. Kuphald. *Lessons In Industrial Instrumentation*, Version 0.2 – Realesed September 29, 2008.

[8] R. Mulley. Control System Documentation – Applying Symbols and Identification, 1993.

[9] I. Sommerville. *Engenharia de Software*, 9ª ed. São Paulo: Pearson Prentice Hall, 2011 (In Portuguese).

[10] S. L. Pfleeger. *Engenharia de Software: Teoria e Prática*. 2ª ed. São Paulo: Prentice Hall, 2004 (In Portuguese).

[11] R. Pressman. *Engenharia de software*. 6ª ed. Porto Alegre: Bookman, 2006 (In Portuguese).

[12] C. A. B. C. W. Madsen, D. F. Adamatti., G. Lucca, G. Daniel. *FURG Smart Games: a Proposal for an Environment to Game Development With Software Reuse and Artificial Intelligence*. The Fourth International Conference on Networked Digital technologies. p. 369-381, 2012.

[13] K. Chellapilla, B. D. Fogel, *Evolution, neural networks, games, and intelligence*. Proceedings of the IEEE, vol. 87, nr. 9, p. 1471-1496, ISSN 0018-9219, 1999.

[14] R. T. Santana. *IA Em Jogos a Busca Competitiva entre Homem e a Maquina*, Faculdade de Tecnologia de Praia Grande.,Praia Grande, Brasil, 2006 (In Portuguese).

[15] J. D. Funge, *Artificial Intelligence for Computer Games: An Introduction*., AK Peters/CRC Press, ISBN 9781568812083 (2004)

[16] B. Schwab, AI Game Engine Programming., Hingham: Charles River Media, ISBN 1584503440, 2004 (In Portuguese).

VI. NEWZOO. GAMES MARKET RESEARCH. AVAILABLE IN: HTTP://WWW.NEWZOO.COM/ (AUGUST 2014)