An authoring tool for educators to make virtual labs

No Institute Given

Abstract. This paper focuses on the design and implementation of a tool that allows educators to author 3D virtual labs. The methodology is based on web 3D frameworks such as three.js and WordPress that allow to make simplified interfaces for modifying Unity3D templates. Two types of templates namely one for Chemistry and one for Wind Energy labs were developed that allow to test the generalization, user-friendliness and usefulness of such an approach. Results have shown that educators are much interested on the general concept, but several improvements should be made towards the user-friendliness and the intuitiveness of the interfaces in order to allow the inexperienced educators in 3D gaming to make such an attempt.

1 Introduction

Educational organizations often use electronic games in order to facilitate learning. These games simulate real-life situations, and allow the learner to be trained in a controlled environment. 3D games have been significantly improved in the last years with the advent of Virtual Reality (VR) technologies. Their maturity has raised the interest of several organizations in order to be exploited in education and training [9],[12]. However, several barriers such as the high cost and the luck of proper design prevent their expansion. In our work, We seek into surpassing these obstacles with a proper designed authoring tool that allows to make virtual labs with a low cost.

Games in general cost a lot to develop [13]. The process of making a game starts with the scenario that defines the principles on which the game relies upon, proceeds to the the game-play that defines its characteristics for playing, next the artistic content is prepared, then the programming takes place, and finally the promotion is responsible for disseminating the game. Educational organizations can not achieve these steps, as they require expert knowledge and experience. Therefore, there is a need to automate this process by providing templates that allow several of the aforementioned steps to be auto-filled. The whole procedure is based on game project templates that incorporate high level organization to allow object behavior inheritance, i.e. by selecting a category for each item, its behavior in the virtual lab is concretely defined. Thus, unnecessary details are hidden from the teachers by allowing items to inherit a pre-programmed behavior, e.g., all items on a table should have a collider so that they stay on the table and do not pass through it. Our methodology relies on the development of a user-friendly platform that is used to create and design virtual labs by using high quality game engines and web interfaces. More specifically, we employ WordPress [7] web content management system in order to develop an editor for Unity3D game engine [14]. This editor is actually a web portal for educators that allows them to build educational game projects. The game projects are compiled by the Unity3D game engine and the game output is therefore a high quality lab.

Analytics are an essential part of improving software products. In the developed templates and the authoring tool, analytics and visualizations were implemented, enabling educators to receive a feedback about the effectiveness of their labs and make changes accordingly. To achieve this, we have borrowed technology from game analytics [11], i.e., tracking infrastructure and game data analysis, and have apply it to the educational context of the authoring tool.

The outline of this paper is organized as follows. In Section 2, the existing tools for making Virtual labs are outlined and discussed. The methodology for making our virtual labs authoring tool is provided in Section 3. The user evaluation of the resulted implementation is provided in Section 4. Finally, conclusions and future work are discussed in Section 5.

2 Existing tools

Authoring tools for making Virtual Labs are few, due to the fact that it is not financially viable to make tools with the strict limitation to create only games for learning. Thus, in order to create educational games, developers and designers resort to one of two solutions: use desktop based game design engines (desktop game-makers) or cloud based game design engines (web game-makers).

The advantages of desktop game-makers are a) the realistic high-quality graphics, b) the large user community that develops new components and functionalities (Leap, Kinect, Oculus Rift, etc.) and c) the support of multiple output formats, e.g., Android, Windows, iOS, Playstation, Xbox, WebGL and others. On the other hand, they require programming skills, and they are computational heavy requiring local installation in high-end PCs. A list of desktop game-makers can be found in Table 1. Unity3D and Unreal 3D engines are the most popular solutions for research projects as they are open source and can be distributed for free assuming non-commercial use. Another option is the Torque3D, being a completely open software with MIT license and quite mature. However, it has a very small community and it does not export the games into mobiles and consoles. Open source solutions such as Blender, Godot, and Copperlicht are face similar problems. For our developments we have exploited Unity3D as it is the game engine that exports to most of the operating systems in desktops and mobile devices.

Web game-makers is an emerging area of the game industry due to the relatively new WebGL feature of HTML5. The benefit of web game-makers is that they are easily accessible through web and easier to learn. However, as a new software area, web game-makers have strong weaknesses. More specifically, they lack of several features of desktop game-makers such as exporting into several

Table 1. Desktop based runtime technologies for making games

Desktop game-makers				
Name	**Compiler - GUI**	**License**	**Exported for**	**Features**
Unity3D	Unity - Unity	Open or Proprietary	Desktop, Mobile, Web, Consoles	Leap, Kinect, Oculus++
Unreal Engine	Unreal - Unreal	Open or Proprietary	Desktop, Mobile, Web, Consoles	Leap, Kinect, Oculus++
Godot	Godot - Godot	MIT	Desktop, Mobile, Web	Lightweight
Torque3D	GFX - Torque3D	MIT	Desktop, Web	Leap, Oculus, RazerHydra
Blender	Blender - Blender	GPL	Desktop	Incomplete
Blend4Web	Blender - Blender	Proprietary	Web	Plug-in for blender
Copperlicht	Copperlicht - Copperlicht	GPL	Web	No plug-ins for third parties

formats, supporting various hardware peripherals, and having a robust physics engine. Physijs was tested and found inadequate for games as regards speed and robustness[1]. Although WebGL is exploiting the graphics card, it is not as fast as a Unity3D game. A list of the existing web game-makers can be found in Table 2. Three.js is by far the most popular open web framework for 3D games. However, it hasn't any reliable GUI. Babylon, Superpowers, XeoEngine, PlayCanvas, Goo, and Cyberix3D have a GUI but it is targeting for programmers as it shows many generic details. In our approach, we want to make a GUI that hides programming details and it leads the educator into making a game with certain steps under certain templates, e.g. a Chemistry template and a Wind Energy template. From the aforementioned review, we have concluded that only desktop game-makers can really offer high quality games. Web technologies can be used though for 3D scene editing only.

3 Methodology

The developed tool supports the authoring of Virtual Labs through the configuration of certain templates, namely the Chemistry and the Wind-Energy templates. In order to make a template several steps are followed. First, the game is designed and it is implemented into Unity3D. Analytics tracking functions are also embedded during game implementation. In this way, any generated game has already the tracking functionality incorporated. Next, the YAML code of the game is split into pieces of code that are inserted into WordPress data structures. Then, the web front-end interface is built according to the requirements for modifications on the split code. The game template can be modified by an educator using the front-end and compiled on the server side. A link of the game

[1] https://github.com/chandlerprall/Physijs

Table 2. Web based technologies for making games

Web-gamemakers					
Name	**Compiler**	**GUI**	**License**	**Export game**	**Features**
Three.js	Three.js	-	MIT	Web	-
Babylon	Babylon	Babylon Editor	ASL 2	Web	-
Superpowers	Three.js	Superpowers HTML5	ISC (GPL like)	Web, Desktop	Real-time collab.
XeoEngine	XeoEngine (SceneJS)	-	MIT	Web	-
Turbulenz	Turbulenz	-	MIT	Web	-
PlayCanvas	PlayCanvas	PlayCanvas	Proprietary	Web, iOS	Real-time collab.
Goo	Goo	Goo Create	Proprietary	Web	-
Cyberix3D	Cyberix3D	Cyberix3D	Proprietary	Web, Android	-

is provided for downloading or accessing the game for WebGL versions. After a game (Virtual Lab) has been deployed, any game data are send to an analytics server that performs all the analysis required. The game analytics are led back in the authoring tool through proper visualizations in order to be inspected by the educator and to make edits in the lab accordingly.

The overall architecture of the authoring tool is shown in detail in Figure 1. The backbone of the system is the Master Server, which contains the creation, editing and design functionalities for authoring virtual labs. The Analytics Server, collects, stores, and process raw game data from each deployed game in order to provide meaningful analytics visualizations back to the Master Server.

The usage scenario is as follows. The Educator access the platform front-end via a web browser. The front-end has also a 3D editor that allows to position the 3D objects correctly. The platform has already uploaded 3D Assets of certain behavioral categories so that the Educator can easily use them. The Educator makes the necessary configurations and saves them into a new Game Project. After configuring the Game Project, and setting up the Scenes with the required Assets, the Educator defines the export format such as Windows, Mac, or WebGL in order for the game to be compiled and waits to receive the binary (Windows or Mac), or the link for WebGL compiled games. The Educator provides the game to the learners for playing. When the game is played it sends game data and the game id to the Analytics server through an API. The data are aggregated and augmented with time statistics, and various features extracted with machine learning methodologies which can be found in a technical reports [4], [5], [6]. The analytics are served back to the Master server through an API in order to be visualized and thus provide a feedback to the Educator. In the following, implementation details are provided.

The software to make an instance of the Master Server is open, namely it consists of WordPress, the developed plug-in [8], and Unity3D game engine. The aforementioned technologies are open and free to install. As regards Analytics

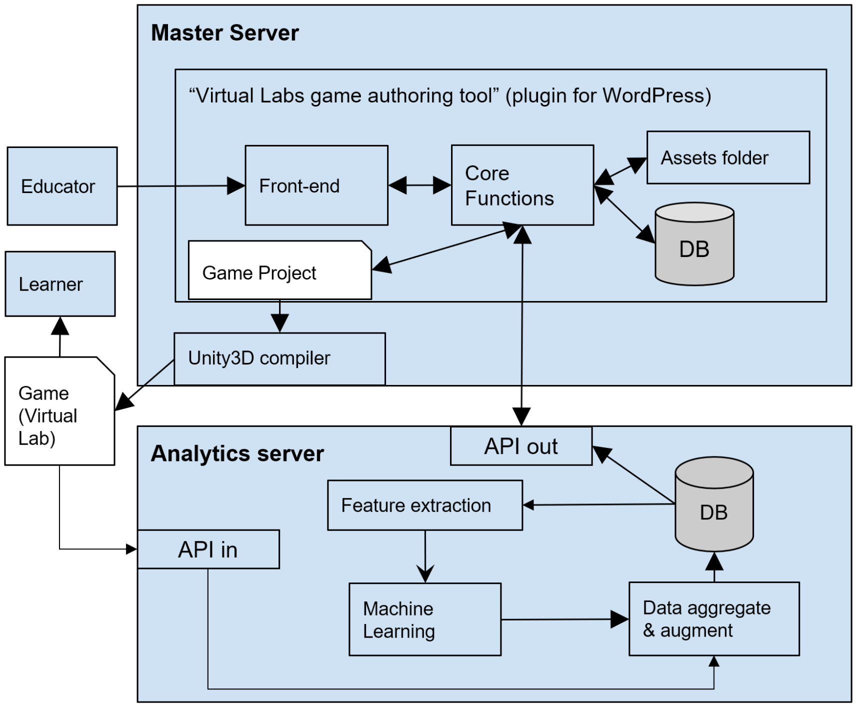

Fig. 1. Overall architecture of the system.

server, it is a commercial system and it is not provided for free[2]. An instance of the Master Server connected with the Analytics server is accessible through the following link:

https://envisagelabs.iti.gr

Upon login in and making an new game project, e.g. a Chemistry Lab, the interface of Figure 2 is shown. The upper part is the 3D editor (based on three.js framework) of the lab scene which consists of several widgets that help into authoring a lab scene. Briefly, the upper part consists of buttons that allow to view the lab in 2D, 3D, first person, or in third person. The left bar has an hierarchy view widget that allows to manage all the objects in the scene. The right-side bar contains all the available assets that can be dragged-n-dropped in the scene. Ray-casting allows drag-n-drop to be efficient, i.e. the assets are placed where the mouse pointer intersects an object. By right clicking on objects of the scene, their properties popup. For example, a laptop or a textbook are objects that serve as gates to other scenes, and therefore the option to select one among the rest of the scenes, namely the Exam scenes, of the game will be shown. More details are explained in the following.

[2] http://Goedle.io

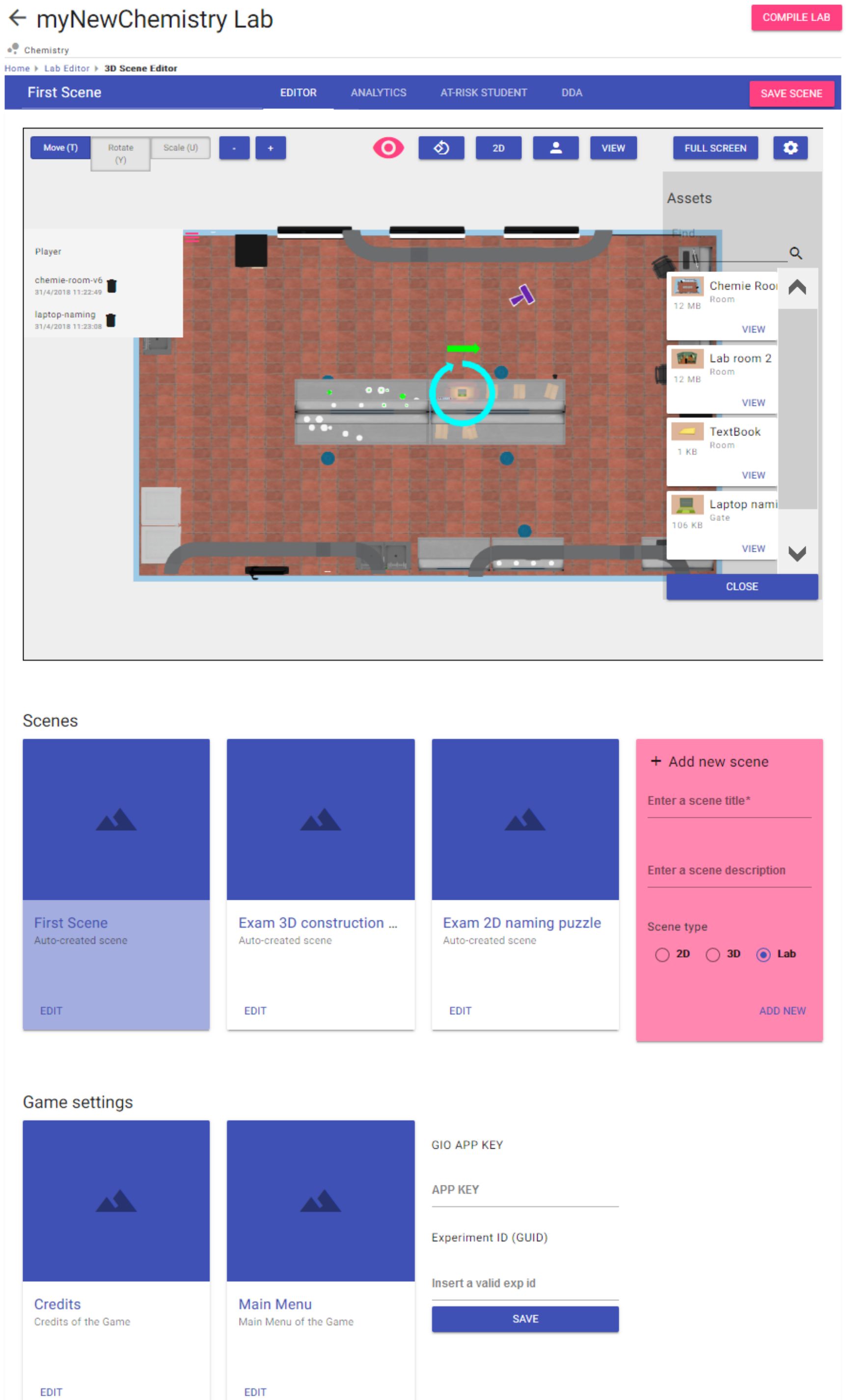

Fig. 2. Authoring a Chemistry Lab.

In the middle part, the Scenes of the Game can be managed. Apart from the First scene where the player can wonder around and place Gates, there are two other scenes, namely two Exam Scenes. The "Exam 3D construction of molecules" and the "Exam 2D naming of molecules" are scenes where the learners test their skills in Chemistry. The Exam scenes contain molecule assets uploaded in Protein Database 3D Format (PDB), where puzzles for stereo-chemistry and molecule naming are posed [2]. These Exam scenes are interlinked with a Gate object (laptop or textbook) when right clicking on a respective object in the scene. In the lower part, the Game Settings are defined. It is mainly configurations about the texts and images in the standard scenes of the games such as as the Main Menu, Credits, and Help scenes. Finally, when all configurations are made, the educator can press the button in the upper right part named as "Compile lab" in order to select the targeted exporting format, and to compile the lab as a Unity3D game.

4 User Evaluation

In this section, we present the results of the evaluation of the authoring tool and the virtual labs created using it. Two methods are applied to evaluate the authoring tool: heuristic evaluation and user testing. Next, we describe both methods along with the evaluation results.

4.1 Heuristic Evaluation

Heuristics are broad design guidelines, which can be used either for creating a user-friendly design or evaluating an existing solution in order to increase its usability. Moreover, heuristics are used as a rule of thumb for either making decisions for new designs or for pinpointing weak points when evaluating existing ones. Their inspection helps identify issues in the UI (User Interface) and is often performed by a group of reviewers analysing the interface based on the heuristics principles. This is typically called an *expert review*.

Usability experts performed a heuristic evaluation on the authoring tool in an effort to improve the user-friendliness of its user interface. More specifically, they utilized the 10 usability heuristics originated by Jakob Nielsen [15] as shown in Table 3. The evaluation focused on the functionalities of the authoring tool, which were: a) Create a new game project, b) create/edit/save 3D scenes, c) create/edit 3D assets, d) insert assets into scenes, e) delete game projects. The most common issues found during the analysis were related to missing previews, a lack of tool tips and help functions and inadequate descriptions of the authoring tool's functionalities. Such issues are natural to appear at this point of development. Many of the issues are related to missing content or ambiguous terminology and will hence not require a substantial amount of recourses to correct. In future, we are going through the issues presented in the heuristics reports and will then rename functionalities, change the design and add more tool tips where it is needed, leveraging thus the findings from the heuristic evaluation to improve the next iteration of the authoring tool.

Table 3. Distribution of usability issues across Nielsen's 10 heuristics in the authoring tool, as found by the inspectors.

No.	Heuristic name	Issue count	No.	Heuristic name	Issue count
1	Visibility of system status	1	6	Recognition rather than recall	0
2	Emphasis on realism	7	7	Flexibility and efficiency of use	1
3	User control and freedom	3	8	Aesthetic & minimalist design	6
4	Consistency and standards	4	9	Help recovery from errors	0
5	Error prevention	2	10	Help and documentation	15

4.2 User test of authoring tool

In this section, the results from the user tests are presented. Questionnaires were used to collect the feedback from the test participants for evaluating the authoring tool, with questions based on [10]. In the questionnaires, the teachers were asked to give a score to statements based on Likert-like scale with rating options ranging from unlikely (1) to likely (7) and "strongly disagree" to "strongly agree". For the tests, the test participants were given a set of tasks in the form of scenarios that took them through the different functionalities of the authoring tool from creating a new lab, adding 3D objects to finally compiling a lab. The tests of the authoring tool were conducted with 5 test participants all of whom are teachers. In Tables 4 and 5 the different questions and the frequency of the ratings by the testers are presented.

One of the participants consistently ranked the statements relating to the usability of the tool in a positive manner, while all others had a more neutral in the response to the authoring tool's usability. The questions that ask directly about the user friendliness of the authoring tool's interface are among the questions that get the most negative feedback. Several of the comments described the need for extended help functionalities such as tool tips on the different elements of the interface, a user manual, tutorial or step-by-step guide to using the tool.

Some comments specifically mention that it was difficult to interact with the 3D scenes. Rotating, scaling and placing objects as well as moving the avatar around in the 3D view is described as confusing and difficult. Some testers found the 3D view hard to understand and were unsure what the students would see in the compiled version of the created lab. Since then, we have increased significantly the user-friendliness of the tool.

The questions related to the usefulness of the tool show that most teachers have a positive attitude, most being above the mid-point and one being very sure about the usefulness. While there were quite a few comments that remarked negatively on the tools user-friendliness in the tested iteration there were also several comments that expressed a confidence that the authoring tool could offer benefits for the teachers going forward once more iterations are completed.

In general, what the answers reveal is that the test participants encountered difficulties working with the authoring tool but at the same time they expressed an interest in the tool going forward and enjoyed the possibility being able

Table 4. Summation of answers in relation to the perceived usefulness of the authoring tool. The score range is 1 (unlikely) to 7 (likely), representing degrees of agreement.

	Rating						
Question	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Using the system in my job would enable me to accomplish tasks more quickly	1	2	6	4	6	1	0
Using the system would improve my job performance	2	1	6	6	2	2	0
Using the system in my job would increase my productivity	2	2	9	4	3	0	0
Using the system would enhance my effectiveness on the job	2	2	5	6	3	0	0
Using the system would make it easier to do my job	2	2	8	4	3	0	0
I would find the system useful in my job	3	3	6	5	1	1	0
The system would enable me to accomplish tasks more quickly	1	3	7	1	4	4	0
I would find it easy to get the system to do what I want it to do	2	1	4	5	3	5	0
My interaction with the system was clear and understandable	1	1	4	6	6	3	0
I would find the system to be flexible to interact with	1	3	4	5	5	2	0
It would be easy for me to become skillful at using the system	2	3	7	3	4	1	0
I would find the system easy to use	3	1	4	4	5	3	0

Table 5. The table summarizes the testers' responses to the questionnaire on system usability of the authoring tool. The score range is 1 (min) to 7 (max), representing degrees of agreement.

	Rating						
Question	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Overall, I am satisfied with this system	2	2	2	3	3	0	1
It was simple to use this system	0	3	3	1	5	1	0
I can effectively complete my work using this system	1	2	2	1	5	0	2
I am able to complete my work quickly using this system	1	2	2	3	3	0	2
I am able to efficiently complete my work using this system	1	1	5	2	3	0	1
I feel comfortable using this system	2	2	3	4	2	0	0
It was easy to learn to use this system	2	1	2	6	1	1	0
I believe I became productive quickly using this system	1	0	5	4	1	0	1
The system gives error messages clearly tell me how to fix problems	1	1	2	1	0	4	4
When making a mistake, I recover easily and quickly	1	3	0	4	1	2	2
The provided information is clear	2	0	4	4	2	1	0
Easy to find the needed information	2	1	1	3	4	1	0
The information provided for the system is understandable	1	2	4	2	2	1	0
The information is effective in helping me	1	2	3	4	2	1	0
Organization of the information is clearly presented	1	2	3	5	2	0	0
The system interface is pleasant	2	0	6	1	2	2	0
I like using the system interface	2	0	5	3	0	3	0
This system has all the expected functionalities	1	1	3	6	1	1	0
Overall, I am satisfied with this system	0	2	5	3	1	1	0

to create a 3D experience for their classroom. The most negative comments regard to the user-friendliness of the tool. Suggestions such as adding tool-tips, more feedback from the system, a simpler help functionality, as well as an undo function could greatly improve some of the issues and should be considered in future developments.

4.3 Virtual labs and learning content

To evaluate the type of virtual labs and learning content that can be created using the authoring tool the teachers were asked to assess a demo version of such a lab, i.e., the Wind Energy Lab [1]. The test participants evaluating the virtual lab were the same as for the user test for the authoring tool, see Section 4.2. They were given a questionnaire (not given in this paper due to the space limitation) and asked to consider how much they (dis)agreed with statements regarding the students' engagement with the lab, quality of educational contents, the fit in terms of the students' abilities, and the teachers' expectations.

Regarding the quality of the learning content and the fit with the curriculum, the teachers agreed that the virtual lab's can be integrated into a learning context. The interface however was rated as needs improvement, thus, we have improved dramatically since then the interface of the authoring tool according to the evaluator rating and comments. The statements relating to the students' engagement and enjoyment of the virtual lab, are concentrated on the neutral midpoint. Though the teachers were satisfied with the content in the lab, they all proposed to further develop the content of the labs so as to better support their teaching.

4.4 Game analytics

Game analytics is something that educators are not familiarized with, and therefore we have conducted a research-evaluation in order to find which types of game analytics visualizations would be useful to them. The visualizations shown to the participants were a dashboard, bar charts, force-directed graphs, chord diagrams and an absolute time-line. For example, a dashboard as shown in Figure 3, is used as an overview of KPI's connected and often also customized to fit a particular objective of its user. Linked to a database, the dashboards can be updated constantly and is frequently used for websites to tack user retention, daily users, revenue, page views etc.

For evaluating the visualizations, the test participants were given a questionnaire with three metrics, each being visualized in two to three different ways. Participants were asked to rank the visualizations internally with the metric and in relation to, e.g., best overview and most informative hereof. The participants were also encouraged to add more in-depth descriptions of why they had ranked the visualizations in the order they did. This evaluation therefore helped us narrow down which visualization is more useful.

Most of the testers had experience in reading and extracting the information from a visualization, which helped them understood them better in our case. Thus, a central conclusion was that providing data analytics efficiently depends on not only the data being visualized but also on the receiver. Another conclusion

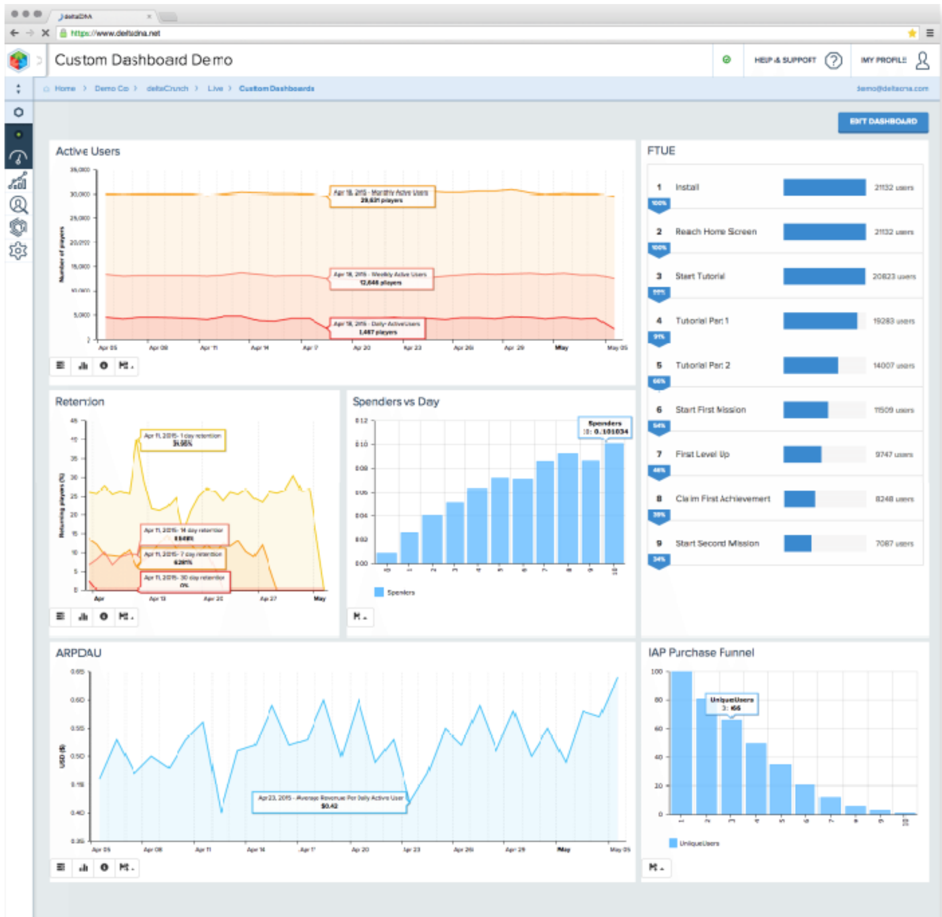

Fig. 3. Visualizations of a dashboard.

is that simpler visualizations were more understandable than complex ones. More details can be found in the respective technical report [3].

5 Conclusions

The authoring tool that we presented has been proven to be a complete solution for authoring Wind Energy and Chemistry labs. However, the development of a tool that allows inexperienced users to author 3D games is a great challenge. We saw by the answers to the questionnaires that the tool was accepted by the educational experts as a potential tool to be used in the class on the condition of several improvements in its user friendliness. Thus, we can conclude that the authoring tool architecture by the combination of the web based authoring capabilities with the compilation mechanism of a game project to a Unity3D virtual lab, was proven successful, but more have to be done in order to use them in real life educational context.

As regards game analytics, it was shown that analytics is feasible to be embedded automatically in a lab, and their use is understandable and meaningful. In the future, we aim to include more data analytics in a context of a larger scale, i.e., many schools, even belonging in different countries. Lastly, we plan to develop new virtual lab templates for our tool, designed for different scientific fields, such as physics, maths, etc.

References

1. ENVISAGE, Wind Energy Virtual Lab. http://www.envisage-h2020.eu/games/energy/v1_1_3/, online, 2017
2. Final version of the "Virtual labs authoring tool". http://www.envisage-h2020.eu/wp-content/uploads/2017/10/D4.4Final.pdf, project Deliverable, Online 2018
3. Implementation of the educational scenarios and evaluation report. http://www.envisage-h2020.eu/wp-content/uploads/2017/10/D5.2-Implementation-of-the-educational-scenarios-and-evaluation-report_Final-version_V3.pdf, project Deliverable, Online 2018
4. Preliminary predictive analytics and course adaptation methods. http://www.envisage-h2020.eu/wp-content/uploads/2017/12/D3.1.pdf, project Deliverable, Online 2018
5. User profiling and behavioral modeling based on shallow analytics. http://www.envisage-h2020.eu/wp-content/uploads/2017/09/D2.2Final.pdf, project Deliverable, Online 2018
6. Visualization strategies for course progress reports. http://www.envisage-h2020.eu/wp-content/uploads/2017/07/D2.3-Visualization-strategies-for-course-progress-reports.pdf, project Deliverable, Online 2018
7. Wikipedia official site for WordPress. https://en.wikipedia.org/wiki/WordPress, accessed 2017
8. WordPressUnity3DEditor plugin for WordPress. https://github.com/Envisage-H2020/Virtual-labs-authoring-tool
9. Bavelier, D.: Your brain in video games, https://www.ted.com/talks/daphne_bavelier_your_brain_on_video_games/transcript?language=en
10. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS quarterly pp. 319–340 (1989)
11. El-Nasr, M.S., Drachen, A., Canossa, A.: Game analytics. Springer (2016)
12. Labster: Empower and engage your STEM students, https://www.labster.com
13. Lovato, N.: What is the budget breakdown of aaa games?, https://www.quora.com/What-is-the-budget-breakdown-of-AAA-games
14. Miles, J.: Unity 3D and PlayMaker Essentials: Game Development from Concept to Publishing. CRC Press (2016)
15. Nielsen, J.: Usability engineering. Elsevier (1994)