Master in Informatics Engineering
MSc. Thesis
Final Report

Multimedia Content for Online Courses

Pedro Cioga
pcioca@student.dei.uc.pt

Supervisor:
Alberto Jorge Lebre Cardoso

24 de Agosto de 2011

FCTUC DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Abstract

Learning is the act of acquiring or improving skills and knowledge, through a synthesizing process of several types of information. The technological-dependent era in which we currently live has led to a significant turn in the habits of learning. The word "e-learning" was coined and became increasingly mainstream. Emerging technologies, such as web platforms for educational and training frameworks, provided the necessary means for developing online education modules.

This report intent to disclose the work carried out during the year-long internship that is part of the Master in Informatics Engineering. The main objective of this internship can be divided in two branches: an online learning platform and an application for building virtual laboratories.

The online platform is intended to gather different types of online courses and resources, combining several kinds of multimedia techniques. The developed application allows teachers to design and create their own virtual laboratory representations. This application was tested on a real classroom and was yet integrated with a haptic device. Improving the student's capabilities of perception and cognition, and easing the setup of attractive online courses and respective contents by the teachers, was also one of the main commitments of this project.

In the end, a thorough analysis concerning how these developed tools can truly benefit both the apprenticeship and the teaching within the e-learning was made.

Keywords

"e-learning", "haptic devices", "learning management systems", "multimedia", "online courses", "online experimentation", "remote and virtual laboratories", "virtual and augmented reality";

Table of Contents

List of Figures

List of Tables

Glossary and Acronyms

Term	Description
API	Application Programming Interface
AI	Artificial Intelligence
CMS	Course Management System
Framework	Conceptual structure used to support software development
GUI	Graphical User Interface
HTML	HyperText Markup Language
I/O	Input/Output
IDE	Integrated Development Environment
JRE	Java Runtime Environment
JSFL	JavaScript Flash Language
JVM	Java Virtual Machine
LMS	Learning Management System
Moodle	Modular Objected-Oriented Dynamic Learning Environment
NPAPI	Netscape Plug-in Application Programming Interface
ODE	Open Dynamic Engine
OS	Operating System
SVG	Scalable Vector Graphics
USB	Universal Serial Bus
VL	Virtual Laboratory
VLE	Virtual Learning Environment
WWW	World Wide Web
WYSIWYG	What You See is What You Get

1 Introduction

1.1 Context

The work described in this report took place at the Department of Informatics Engineering of the University of Coimbra, under the guidance and supervision of Alberto Jorge Lebre Cardoso, professor at the Department of Informatics Engineering of the University of Coimbra. The work here exposed was the outcome of the joint efforts of the development team, composed by two Design and Multimedia students, José Nuno Monsanto and Miguel Delgado, beyond the author.

1.2 Motivation

Since the beginning of mankind, Man was able to learn, by processing information obtained from his surroundings, and retaining that knowledge. The primal trial and error learning method evolved to increasingly more complex ways of learning, which allowed Man to draw conclusions and theorize about them [24]. Knowledge led to progressively deeper knowledge and with it, technology arose.

Technology grew exponentially, serving as a foundation for the new technologies to come and the world took a critical shift towards a technologically-dependent society. Therefore, all society's cornerstones such as Economy, Health and Education, mandatorily followed alongside this growth. On the specific field of Education, every kind of learning, teaching or training delivered by any electronic means started to be referred as e-learning.

Nowadays, e-learning is undoubtedly, the most comprehensive way to share knowledge at a worldwide scale. On the United States alone, this activity has been increasingly growing for the past 8 years and in 2011 the number of students learning online has surpassed the six million barrier, with nearly one-third of all students in higher education taking at least one online course [44].

So what can be added to this field? How can the currently used methods be improved in order to enhance both the student and teacher experience?

Keeping the students motivated is a crucial part of the learning process, as an unmotivated student is more likely to quit or fail a class than a motivated one. There are several motivational and pedagogical techniques that a teacher should always keep in mind, however they are not covered by this project's scope. Offering the student a fresh and rewarding experience is a great way to get him motivated. For that purpose this project intended to create a platform to host all the multimedia contents developed, while offering a personalized experience to the student, and simultaneously easing the teacher's task of setting up a new online course and its respective resources.

The multimedia contents produced on this project were mainly directed to the development of Virtual Laboratories (VL), allowing the users to assist or conduct a virtual experiment. Initially, the project sought simple solutions that would enable teachers to implement their own scenarios, but the fact is that not all the teachers have the necessary skills to implement a virtual laboratory scene by their own means. Therefore this project intended to develop a tool that enabled the user to easily create a virtual laboratory scene, becoming imperative for it to be as most intuitive and simple as possible. Also, with the goal of improving the user experience, this project aimed to integrate a haptic device with the designed VLs, creating foundations for the augmented reality concept.

For the validation of the developed tools, this project relied on the support of two high school teachers of the physics: Adriana Maria Pinto Marque Nave and Margarida Figueiredo from school 2,3/Sec do Agrupamento de Escolas de Penacova. Therefore the designed VLs become more focused on this scientific area.

This project also benefited of the cooperation with the Department of Mechanical Engineering of the Faculty of Engineering of the University of Oporto, that beyond all the help offered, provided the two haptic devices to this project.

1.3 Proposed Goals

The goals for this dissertation can be divided into two main interconnected parts:

- **Internship**, which focuses on the knowledge and experience acquired by the intern;

- **Project**, which corresponds to the development of all the defined requirements;

1.3.1 Internship

The main goals of this internship are the ability to consolidate knowledge about Software Engineering to improve the software development experience, while consolidating the developed work with a thorough search on the areas covered by the project. The integration on a software development team, and all the benefits and disadvantages which arise from this experience, was also a substantial goal for this internship.

1.3.2 Project

The main goals for this project can be divided in two main categories and their respective main objectives:

- **FLOCK**
- **Carbono**

These two categories are described in more detail below.

FLOCK

Flexible Learning & Online Collaborative Knowledge (FLOCK) is the main platform intended to provide online courses and respective contents. Hosted by the University of Coimbra server can be accessed on the address flock.uc.pt. FLOCK was intended to be capable of meeting the requirements outlined, while promoting teacher and student communication, knowledge sharing and direct interaction with the VL tool, Carbono.

Although the author was an active presence on all the development process steps and has contributed with his work on a regular basis, the development of FLOCK was mainly of the responsibility of Miguel Delgado. The author's most visible work on FLOCK was the integration between this platform and the Carbono tool.

Carbono

This tool emerges from the necessity to equip teachers with an instrument that would enable them to create their own virtual laboratories. Initially the project looked at available technologies that would allow one to create a laboratory scene, but quickly came to the conclusion that a tool for this specific matter didn't exist, and all the available programming languages or technologies for that end, require a considerable learning period. This preliminary work achieved by this project, can be consulted on the appendix: Appendix B– Preliminary Work.

Carbono arose from the joint effort between the author and José Nuno Monsanto, both responsible for the application's functionalities specification and requirements. José undertook the application's design while the author developed its architecture and implementation.

So the project followed with the intention to create an intuitive tool that offered basic options for creating and editing a scenario capable to meet the following features:

- Object creation;
- Object manipulation: rotation, position and scale;
- Interaction between objects;
- Multiple camera options;
- Creation of at least one virtual laboratory;
- Interaction with FLOCK;
- Haptic device integration;

Carbono is intended to become a tool capable of represent several types of scenarios from different science fields. Due to this project collaboration with the physics teachers, the first version of this tool highlights important features for the mentioned field.

Creating a VL editor from scratch was expected to be a very arduous and complex task besides the inherent time consumption.

With that in mind, besides the Carbono editor was also created a laboratory scene from scratch, using the same technology of the editor development. The objective was to not be entirely dependent upon the completion of a first version of the Carbono editor.

This way was possible to get important results, relevant to the research, while showing at the same time, a demonstration of what the Carbono editor could accomplish. This demonstration benefited from the direct collaboration of the two physics teachers aboard this project, what was one of the most important goals for this project. This matter will be further discussed later on this report.

The same policy was applied to the haptic device integration. The availability of a haptic device by the development team was an opportunity that could not go to waste, since it could transcend our project to other multimedia field, e.g. the augmented reality. This matter will also be discussed in more detail, later on this report.

Carbono eventually became a tool fragmented into three different types of utilization. These types are described below

The editor

In the future, Carbono intends to become a powerful editor capable of creating different kinds of VLs, independently of the science field addressed. Due to the partnerships established by this project, this early version's main focus was the physics field. Very briefly, the main idea for this editor's version was to allow the user to add some objects to a scene and allow some simple physical interaction between them.

The virtual laboratory

In order to take advantage of the feedback that could emerge from the physics teachers and their respective classes, was essential that the created VL represented part of their curricular plan.

After some meetings with the teachers, was decided that the project would address the matter concerning an inclined plane system, its respective forces, velocity, kinetic energy, etc.

A more detailed information about this type of system and also the physics experiments protocols in which this VL was based, can be consulted on the appendix: Appendix D- Laboratorial Procedure

The haptic device laboratory

This project had at its disposal the following haptic devices: the Novint Falcon[42] and the PHANTOM Omni[26]

The main goal of the haptic device laboratory was to integrate one of these devices, and its particular features, with Carbono, taking the application's first step into the augmented reality field. A basic scene where the user would interact with some object, while using the haptic device, was considered sufficient to meet the minimal requirements.

The development of Carbono was sustained by a game engine as will be discussed below. The Carbono's guidelines concerning its architecure, design and user interface, features and functionalities emerged from the cooperation between the author and José Nuno Monsanto, member of the development team.

1.4 Achieved Goals and Results

Overall is safe to say that this project obtained very positive results as will be documented along this report. The main proposed objectives were achieved, the project planning proved to be effective, all the plotted contingency plans resolved foreseen issues, and the conducted tests returned a very positive feedback.

Contrary to what was expected the FLOCK platform was not built from scratch and instead was hosted by a Learning Management System, in this case Moodle.

The virtual laboratory creation tool, Carbono, was implemented taking advantadge of a game engine, Unity 3D. Three distinct applications were developed sharing the same basis: The editor, the virtual laboratory and the haptic device laboratory.

The editor, still in an early development stage, was the cornerstone of Carbono and the starting point for all the future versions of this application.

The virtual laboratory created served the purpose of demonstrating the real capabilities of the Carbono tool. Taking advantadge of the project's collaboration with the school 2,3/Sec do Agrupamento de Escolas de Penacova, this application was tested in a real classroom with its respective students and teachers.

The haptic device laboratory was this project's first step into the augmented reality field. Leveraging the collaboration with the Department of Mechanical Engineering of the Faculty of Engineering of the University of Oporto, this project integrated a provided haptic device with the developed application.

The project developed work led to the writing of a paper entitled "Demonstration of Online Educational Modules with Online Experiments" by the authors Cardoso, Alberto (University of Coimbra); Restivo, Maria Teresa (UISPA, IDMEC-Pólo FEUP, Faculty of Engineering, University of Porto); Cioga, P. (University of Coimbra); Delgado, M. (University of Coimbra); Monsanto, J.N. (University of Coimbra); Bicker, J. (University of Coimbra); Nunes, E. (University of Coimbra); Gil, P. (University of Coimbra, Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

The written article was later presented on an exhibition session of the International Conference REV 2012- Remote Engineering & Virtual Instrumentation and will soon be published in a special issue of the online journal "International Journal of Online Engineering (iJOE)" – http://www.online-journals.org/i-joe/

1.5 Risks

Every project needs to contemplate the possible risks that may occur during the development process, and needs to create effective strategies to overcome, prevent or workaround them. This analysis should be handled periodically in order to reflect on risks not foreseen on earlier stages of the project.

The most significant risks appointed during the development process and the solutions conceived to overcome them, are listed below:

- **The FLOCK platform**
  To meet the expected requirements of the project, FLOCK had to be able to host online courses and respective resources and simultaneously, interact with Carbono.
  - Attenuation plan: In case the FLOCK platform doesn't met the expected requirements, it should be hosted on a system capable of achieving the appointed goals, e.g. a LMS.

- **Carbono tool**
  To achieve two major objectives for this project, the Carbono tool had to be able to create a functional virtual laboratory, and allow interaction with a haptic device. This could take too long to develop, especially when this project had the duration of one year.
  - Attenuation plan: If Carbono couldn't meet the expected requirements in a valid time, it should be created a functional virtual laboratory and a haptic device controlled application, using the same architecture of the Carbono tool. This way these objectives could be achieved and the integrity of Carbono's architecture was not endangered.

- **FLOCK and Carbono integration**
  One of the main features that this project intended to accomplish was the opportunity to interact with a virtual laboratory as a resource for an online course. With that in mind the created laboratories should always be available to students even if there isn't an available internet connection.
    - Attenuation plan: Besides the web embedded version of the VL, a standalone version should always be available for download. If a student doesn't have an internet connection momentarily, he should be able to progress his learning process. This might also be an efficient solution for some browser incompatibility issues with the Unity web player that may arise.

1.6 Documentation overview

This document is organized according to the following chapters, accompanied by their respective descriptions:

- **Introduction**
  This section presents an overview of the work carried out during the internship's project, including its contextualization, motivation, goals and risks.

- **State of the art**
  This chapter will define a list of products and services that could bring some enrichment to the project.

- **Requirements**
  This chapter describes the features supported by the project platform, as well as the features supported by the Virtual Labs tool and the technologies used during their development and validation process.

- **Architecture**
  This chapter describes the developed applications architecture and provides an overview of their technical behaviour.

- **Project Planning**
  This chapter describes the project development methodology and its initial and revised planning.

- **Deployment**
  This chapter describes the tests made to guarantee the project's quality and to confirm that all the established requirements were achieved.

- **Conclusions**
  This chapter summarizes the work done during the development process, the accomplished goals, the contributions made to the field. Furthermore it outlines the future work.

- **References**
  This chapter describes the references used on the present report.

1.6.1 Documentation

The present document and all its appendixes are an integral part of the work undertaken during the author's one year internship for the Department of Informatics Engineering of the University of Coimbra. The following work was supervised by Alberto Jorge Lebre Cardoso, professor at the Department of Informatics Engineering of the University of Coimbra.

Attached to this document are the following appendixes:

- **Appendix A – Project Planning**
  This document contains the Gantt diagrams created during the planning phase of this project.
- **Appendix B – Preliminary Work**
  This document features the preliminary work developed on an early stage of this project.
- **Appendix C – Requirements Analysis**
  This document corresponds to a description of analysis developed to define the applications requirements.
- **Appendix D – Laboratorial Procedure**
  This document summarizes the laboratory procedure addressed by the project.
- **Appendix E – Application Architecture**
  This document schematizes the architecture of each of the project's applications.
- **Appendix F – Application Overview**
  This document summarizes the features and functionalities of each of the project's applications.
- **Appendix G – Functionality and Usability Tests**
  This document documents the tests and respective results, conducted over the project's applications usability and functionalities.
- **Appendix H – Published Paper**
  This section presents the paper "Demonstration of Online Educational Modules with Online Experiments" published with the contribution of the referred project.

2 State of the Art

In order to create the platform that would embrace the online courses and all the developed contents, FLOCK, it was necessary to have guarantees that this project was built on a solid basis. After the decision of basing this platform on a Learning Management System, it became crucial to examine the strengths and weaknesses of many LMSs available, which could support the main features established for this project. Only then a safe step could be taken into developing the work previously proposed. None of the studied LMS were originally capable of representing a laboratory scene, although some of them provided plug-ins to that end, they couldn't deliver the desired results.

Before the decision of creating a tool for the VLs representation from scratch, a search for programming languages or technologies which could cover this issue and be web based simultaneously was conducted. It was fundamental that the simulations could be displayed on the web, since the courses are logically online, and it was important that the chosen language or technology could give a positive answer to any kinds of simulation, in order to reduce the learning period.

Afterward was handled a research for the best way to create the VL tool, Carbono, browsing for a light and tough game engine, which could consolidate the project's objectives and offer the possibility of web integration at the same time. As mentioned before the project is inserted in a year time internship, and as such, time was a constant limitation. To work around this issue, was early determined that a Game Engine was the best solution to our problem. May there be other qualified alternative solutions like Simulation Engines or Physics Engines, those were early discarded since they oblige a longer learning curve and most don't offer important features such as graphical user interfaces (GUI), browsers and OSs compatibility, graphics rendering, etc.

All this technologies should be able to run over any Operating System (OS) and simultaneously in any web browser supported by it. With that goal in mind, an analysis of the most-commonly used web browsers was conducted. Finally, the application was tested in the latest versions of the following browsers: Internet Explorer 9.0, Google Chrome 16.0, Mozilla Firefox 8.0, Opera 11.52 and Safari 5.1.2.

These approaches and their main characteristics will be discussed below.

2.1 Learning Management Systems

The learning management system's main purpose is to administrate, document, track and report training courses, online classrooms and any other branches of the e-learning context [40]. While was intended to bring new multimedia concepts for this type of systems the project had to rely on a robust basis able to support the following features:

- Open source;
- Web based;
- Easy administration;
- Assemble and deliver learning content quickly and easily;
- Personalize content and enable knowledge reuse;
- Offer different evaluation means like quizzes and exams;
- Provide different ways of knowledge sharing like forums, text chat and surveys.

With the extensive offer of competent LMS's around the web, it was imperative that the choice relied on an open source solution. This way the learning environment could be personalized while having a closer control of the user decisions, input data, etc. As mentioned before, it had to be a web based platform to reach a larger audience while meeting the fundamental e-learning requirements.

Features like uncomplicated administration, fast delivery of resources, personalization of contents and knowledge reuse, were truly important since it permit to create a simple online course demonstration, where the developed multimedia features could be fit, without wasting too much time or resources.

2.2 Virtual Lab Simulations

As mentioned before, the project initially searched for programming languages, frameworks or technologies able to represent laboratory scenes. It's was very important that the developed work displayed a satisfactory performance on the most common OS's and web browsers while, if possible, reduce to a minimum the system updates necessary for it to run.

2.2.1 Adobe Flash

Adobe Flash[1] is a well-known multimedia platform that increases animation and interactivity to web pages. This platform provides animation of text, still images and drawings, supports bidirectional streaming of audio and video, and it can capture user input via I/O devices such as keyboard, mouse, camera, etc. This technology could give a satisfactory answer to our needs since it contains an object-oriented language, Action Script, supports automation via JavaScript Flash Language (JSFL) and as such it's compatible with the most commonly used browsers. Its capability to capture user input via I/O devices is also well regarded.

However, in some cases Flash can take a considerable amount of time to load, despite the high speed internet services available nowadays. One must install the cross-platform browser plug-in, Adobe Flash Player, on his system to be able to benefit from this technology on the web browser of his choice. Some flash applications may require an update of the installed version.

2.2.2 Java

The Java[13] language presents yet another valid solution to the existing problem. With the Java Applets[12] feature, this language offers different means to represent the laboratory scenes. Java Applets are delivered to users in the form of Java bytecode and can run in a web browser using a Java Virtual Machine (JVM). They also can use 3D hardware acceleration available from Java. For all this is absolutely required to have the Java plug-in installed on the system.

Since Java's bytecode is OS independent, Java applets can be executed by any web browser. As long as properly implemented it also enables listening to I/O devices events. Applets can be quick to load as most web browsers cache applets, and therefore accelerate the loading process when returning to a web page. The same applet runs on all installed versions of Java rather than just the last plug-in version available. However, if an applet requires a newer version of the Java Runtime Environment (JRE) than the one available on the system, the user will need to wait for the considerable large JRE download to update his version.

2.2.3 HTML5 + Processing.js

HTML5 is a language for structuring and presenting content for the World Wide Web (WWW). This technology is currently in its fifth revision and in January 2012 it is still under development. With the intent to make it easy to include and handle multimedia and graphical content on the web without having to resort to plugins and APIs, this revision has added new features such as video, audio and canvas elements, as well as integration of Scalable Vector Graphics (SVG) content.

In this context arise the Processing[21] language and the Processing.js[22] project. Processing is an open source programming language with an Integrated Development Environment (IDE) built for electronic arts and visual design such as images, animations and interactions. It can be used to create the desired scene models and has OpenGL[1] integration for accelerated 3D.

Processing.js is the JavaScript port for the Processing language and allows the web browsers to display the designed visualizations without the need to install any kind of plug-in. It uses JavaScript to render 2D and 3D content on the HTML canvas element and as such is supported by browsers that have this element implemented.

A program written with the Processing language and interpreted by Processing.js doesn't require any kind of version updates, as long as the browser is compatible with HTML5 canvas and supports JavaScript. Hence an OS's runs a web browser with this features, it consequently supports this technology.

2.2.4 Microsoft Silverlight

This application framework has features and purposes very similar to those of Adobe Flash and its run-time environment is available as a plug-in for most web browsers. Though an upcoming Opera (web browser) support was promised since 3 May 2007, as of January 2012 Silverlight still does not officially support this web browser.

This technology may as well represent the lab environments, even if it does not currently support 3D hardware acceleration. Also sustains user I/O events listening, capturing the user behaviour with the application. Some applications may be version dependable and require an update. In some OS's like Mac OS 10.4 we encounter some browser incompatibilities, being Mozilla Firefox 3, and previous versions, and Safari the only browsers that support this technology. On a Microsoft Windows OS, the Safari browser only supports Silverlight via Netscape Plug-in Application Programming Interface[2] (NPAPI).

Currently there are unconfirmed rumors that Microsoft will abandon Silverlight[17] in its upcoming OS version of Windows 8.

2.2.5 Others

There were other technologies taken into account, like Adobe Shockwave for instance, but were not part of the analysis as they do not represent a valid alternative with

---

[1] Cross-platform API for writing applications that produce 2D and 2D graphics.

[2] Cross-platform plug-in architecture that handles certain content types (e.g. audio, script).

better features to our objectives. Libraries like Java 3D and engines such as 3DzzD or Paper vision 3D, were left aside as their foundations settle on the studied languages or technologies or just because they could not bring a valid solution to 2D and simultaneously 3D simulations.

2.3 Game Engines

Game engines are systems specifically designed for the development of video games. Some are developed to work on video game consoles, on personal computers or both. The main functionalities provided by these platforms are 2D and 3D graphics rendering, physics engine and/or collision detection, sound, animation, memory management, threading and a scene graph. Some game engines supply a suite of visual development tools provided in an IDE to enable simplified and fast development of games. Conveniences like graphics, textures, sounds and AI (Artificial Intelligence) functions are often provided by these suites in the interest of ease the development.

In order to create the Carbono tool, without duplicate work previously created or optimized by others, was assembled an analysis over some available game engines which provided a development kit and outlined some specific parameters to be achieved. It was again important for the game engine to be free and web based, and as long as it granted a simple and easily understandable editor, the closed source liability became meaningless. Compatibility with the key OS's and web browsers was, once more, a critical concern for this project.

Specific game engine's services like physics engine or collision detection are very well-regarded. The plug-in/installation issue is not possible to get around, since all of the addressed game engines need some kind of plug-in to be visible on the web browsers. At best, the developed work can be exported for Abode Flash technology, which is more frequently used.

On the conducted analysis only two game engines acceptably fulfilled the stipulated requirements: ShiVa3D[27] and Unity 3D[32]. The result of the individual analysis for each one of these engines was as it follows.

2.3.1 Unity 3D

Unity 3D consists of both an editor, for developing 3D video games or other interactive content such as architectural visualizations or real-time 3D animations, and a game engine for executing the final product.

Its development environment is compatible with Microsoft Windows and Mac OS X, and the produced contents can be performed on Windows and Mac, as well on gaming consoles like Xbox 360, PlayStation 3, Wii, and also on mobile platforms Oss such as iOS and Android. The created applications can be directed for web browsers using the Unity web plug-in, supported by the most significant browsers.

A great disadvantage is that despite several indications that a Linux port is in the works, as August 2012, Unity Technologies has not yet made any official public statement regarding an estimated release date. In February 2011 was announced that Unity will be able to export to Adobe Flash, although this development is still ongoing. This will allow users to play Unity's content with a Flash browser plug-in, being a suitable solution for the Linux compatibility issue. Presently, neither the Adobe Flash nor the Linux port has yet been released.

Unity's IDE present a What You See is What You Get (WYSIWYG) philosophy, displaying on screen, while edition is taking place, a closer representation of the final product. It comes with an integrated physics engine, bump mapping[3], reflection mapping[4], lightmap[5] and dynamic shadows, all important tools to bring a more realistic feel to Carbono.

Unity supports integration with different modelling and rendering applications like 3dsMax, Blender, Cinema 4D, and Photoshop, smoothing the importation of 3D model objects. Its graphics engine uses Direct3D[6], OpenGL and OpenGL ES[7] (iOS, Android) technologies.

Unity's shader[8] can include multiple variations and a fallback specification, allowing it to detect the finest settings for the current video card and if none are compatible, fallback to an alternative shader, that may sacrifice features for a wider compatibility.

2.3.2 ShiVa 3D

ShiVa 3D is a game engine with an incorporated graphical editor designed for the creation of video games and 3D applications for the web, consoles and some mobile devices. It offers compatibility with the key OS's, Windows, Mac and Linux, with mobile devices OSs such as iOS and Android and some gaming consoles. Its applications can be standalone or embedded in a web browser, presenting consistency with the all main browsers, through ShiVa web plug-in.

This engine is also provided with a WYSIWYG visualization on board of its IDE, and has integrated lightmap control, bump mapping and reflection mapping offering this way a real representation of reality to the project. However, ShiVa 3D doesn't have an in-built physics engine. It compensates this flaw with a support for the Open Dynamics Engine (ODE) physics engine which is free.

Shiva also offers support for importing object models from other applications as 3dsMax, Blender and Cinema 4D. This game engine uses OpenGL, OpenGL ES or DirectX[9], and can also run in software mode, featuring that way a wider range of compatibility.

2.4 Haptic Devices

The main reason for the inclusion of haptic devices on the project was to turn the virtual laboratories into an even more interactive experience. In order to enter the augmented reality field, it was necessary for these devices to simulate three dimensions movement and also provide force feedback.

2.4.1 Novint Falcon

[3] Technique for simulating bumps and wrinkles on the surface of an object.
[4] Image-based lightning technique for approximating the appearance of a reflective surface
[5] Data structure which contains the brightness of surfaces in 3D graphics applications
[6] Part of DirectX API, used to render three dimensional graphics
[7] Subset of the OpenGL API, designed for embedded systems (e.g. phones, tablets)
[8] Computer program that is used primarily to calculate rendering effects on graphics hardware
[9] Microsoft platforms API's collection for handling tasks related to multimedia (e.g. game develop, video)

The Novint Falcon is a USB haptic device intended to replace the mouse in video games and other applications.

This device has removable handles that the user holds to control the Falcon. The handle has four different buttons that can be assigned with any function. The user can move the handle in three dimensions: right-left and forwards-backwards, like a mouse but also up-down. This way, the device has the ability to move along the three axis lines of the Cartesian coordinate system (x,y,z). The Falcon software keeps track of where the handle is moved and can create forces that the user can feel, by controlling the motors in the device. The motors are updated 1000 times per second giving a realistic sense of touch. This way it can be used to simulate the touch on different textures of an object as well as its weight and dynamics.

These features are within the requirements sought by the project, for the haptic device module.

The picture below shows a Novint Falcon device, similar to the one used by this project.

Figure 1- The Novint Falcon

2.4.2 PHANTOM Omni

The PHANTOM Omni is a haptic device, developed by Sensable, connectable to any PC or laptop with an available FireWire port.

The user controls the device by holding its pen-shaped handle, and due to its six degrees of freedom positioning, it's possible to accurately touch and manipulate objects in 3D space. This allows control over any position of the Cartesian coordinate system. Its handle features two buttons which can assume different functions. The device is capable of applying forces which can be controlled to create a range of effects, such as weight and dynamics.

The Figure 2 shows the PHANTOM Omni from Sensable.

Figure 2- The PHANTOM Omni

2.5 Conclusions

2.5.1 Learning Management Systems

From the research carried out, arose many LMS's which offered a positive response to our demands, came across our analysis, but we ended up choosing Moodle[18] (Modular Objected-Oriented Dynamic Learning Environment).

The Moodle platform has not only met all the main requests, it surpassed them. Unquestionably the fact that it is an open source, free and web based system weighed in the final decision, but after working with, its great administration tools, customizable learning styles, communication channels, social networking and activities tool, had to be highlighted. Due to Moodle, creating the backbone of the demo online course has become an easy and simple task.

2.5.2 Virtual Lab Simulations

After analyzing all the previously cited technologies the results were objectively summarized on the following table:

	Adobe Flash	Java	HTML5 + Processing.js	Microsoft Silverlight
2D scene	✔	✔	✔	✔
3D scene	✔	✔	✔	✔
Capture user input	✔	✔	✔	✔
Web based	✔	✔	✔	✔
OS independent	✔	✔	✔	✘
Installation or plugin free	✘	✘	✔	✘
Version incompatibility free	✘	✘	✔	✘
Web browser incompatibility free	✔	✔	✔	✘

Table 1- 2D and 3D simulation technologies comparison

As it is visible, the collaboration between HTML5 and the Processing language is the technology that best suits this project's interests. All the studied technologies allow developing the main features for this project's purposes.

The fact that it is OS independent and supported by the most commonly used web browsers was a critical requirement for the chosen technology, and as such the Microsoft Silverlight did not get a positive evaluation. Adobe Flash was a very suitable alternative but the final decision hung on the author personal preference and by the fact that this technology is not yet fully supported by Apple's mobile OS. Therefore it can represent a disadvantage if in the future this project were to be extended to mobile devices, although Adobe has been trying to deal with this incompatibility with iOS[10], natively.

The need of a plug-in or installation and the possibility of version incompatibility, in the Java case JRE incompatibility, represent a negative aspect between all the studied technologies and the HTML5 and Processing collaboration.

2.5.3 Game Engines

The results obtained over the individual comparison of the two analyzed game engines are reviewed on the following table:

	Unity 3D	ShiVa 3D
Free	✔	✔
Web embedded	✔	✔
Main web browsers compatibility	✔	✔
Main OS's compatibility	✘	✔
Installation or plugin free	✘	✘
Object model importation	✔	✔
Physics engine built-in	✔	✘
Hardware compatibility module	✔	✔

Table 2- Game engine comparison

As is clear there are very few differences between these two game engines. Both of them would give a suitable response for the Carbono development. The final decision hanged between two main concerns: spare more time and resources to embrace the ShiVa 3D IDE and simultaneously the ODE behaviour, having the final product compatible with all the key OS's, or otherwise save time and resources while assimilating the development over the Unity 3D IDE, knowing that the outcome may not be compatible with the Linux system's in the near future.

In the end the decision fell upon Unity 3D. As this project is part of a six months program, time became a very important concern. To achieve the goals earlier stated is fundamental to have an early functional Carbono version, which can be tested and upgraded over time and thus show some actual features and functionalities, than it to be fully compatible with all OS's. Nevertheless the Unity Technologies statements regarding the Linux port and the Adobe Flash export possibility gave some assurance over this decision. Even for the integration of Carbono with a haptic system, another goal for this project, was

[10] Apple Operating System for Mobile Devices (e.g. iPhone, iPad). More info at: http://www.apple.com/ios/

truly crucial that the development of the application doesn't took too long. The vast resources, tutorials and community around this engine were also a fundamental aspect.

The fact that both engines can export the application to a mobile device was not decisive but taken in account since it may be very useful for future improvements of the main project.

2.5.4 Haptic Devices

The haptic devices available on this project matched the established requirements: both were able to manipulate and control objects on a 3D environment, and both were provided with force feedback.

Hereupon, the decision over which device would be used on the project was held up by which had the most comprehensive development support and more available information. However this wasn't enough to make a definitive choice, since both also presented very comparable features at this point.

The project moved forward and started to experiment with both the devices and respective software. The Unity Engine didn't offer any kind of support for these devices. Hence, the project needed to develop a plug-in capable of establishing a communication bridge between the hardware drivers and the Unity Engine.

In the end, this task proved to be a lot less complex to achieve for the Novint Falcon, instead for the PHANTOM Omni, much due to its development kit. The scarcity of information regarding the integration of the Unity Engine with this kind of device remained a constant problem along the process, but the Novint Falcon and Unity communities proved to be an added value to this project.

3 Requirements

3.1 Introduction

In every project, the first step on the software development stage involves the explicit definition of the project needs and objectives, i.e. its requirements. These are always present during the entire development phase, serving as a guideline. In order to accomplish this was developed a task analysis and documented different user scenarios.

A task analysis refers to "any process that identifies and examines the tasks that must be performed by users when they interact with system"[30]. The general term Task Analysis can be applied to a variety of techniques for identifying and understanding the structure, flow and attributes of tasks.

User scenarios are stories about how a user would possibly interact with a system. They describe environments and situations where a user would use a certain application, consequently providing a description of the application's interaction model from the user's point of view.

Both these approaches were conducted with the active collaboration of the development team members Miguel Delgado and José Nuno Monsanto.

3.2 Task Analysis

A task analysis can go from precisely detailed to something a little more relaxed, depending on the scope and objectives of the analysis. This analysis can become very time consuming if used with a high degree of detail, being easy to get caught in an "analysis paralysis", where the details are increasingly investigated and the job is never really finished[31]. Besides helping identifying the tasks that applications must support, this analysis is useful to later on ensure that the design supports all the tasks required.

For avoiding unnecessary time wasting and given the dimension of the project a simpler approach was sufficient. A task analysis can consist in a simple, raw list of features that the final application will have to carry[38].

The data for this type of analysis can be assembled from many places including business requirements, user research or even brainstorming. For a simple system, like the one this project comprehends, tasks were simply identified by brainstorming sessions between the development team members as well as by questioning users. This project task analysis contemplated all the previous steps (meetings, the study of competitive solutions, etc) from where the tasks were thoroughly identified and jotted down.

The tasks were then sorted in two moments: during edition/creation and after publication. This task list is a conceptual list of all the tasks the user might perform in each stage. The development of this analysis was a continuous process, since the list was changed repeatedly throughout the whole project, especially after the test results made to the prototypes. After these test sessions some features were considered unnecessary and ideas for new features emerged. This reflected a constant evolution of the task list.

The complete task analysis can be found in the appendix: Appendix C-Requirements Analysis

3.3 User Scenarios

Documenting user scenarios can be really useful, since it help removing the focus from the technology, opening up the design possibilities. Since they allow seeing past the technology, user scenarios promote the discovery of new features or different ways of doing something, that otherwise couldn't be perceived[37].

Is fundamental to have an understanding of the system's users and the tasks they need to perform in order to create user scenarios. Is also very important to have a perception of the context in which the application will be used. This kind of information was only possible to gather by the previous steps taken by this project on the Analysis stage.

The scenarios created for this project helped to explore several situations in which the application could be used, and how the typical user could interact with it. The process of imagining the application utilization in some of these scenarios also helped decision-making when choosing how to implement some features.

The user scenarios produced for this project can be found in the appendix: Appendix C- Requirements Analysis

3.4 FLOCK

3.4.1 Requirements

Although the author had always an active presence and an incisive opinion, the development of FLOCK, was mainly of the responsibility of the develop team member Miguel Delgado. For the reported project the main requirement for this platform was to establish a bidirectional communication with the Carbono application.

3.5 Carbono

3.5.1. Requirements

This section enumerates which features should be included on the application and presents the results of the deeper analysis carried out on the previous stages.

The requirements divided Carbono in three main applications: the editor, the laboratory scene and the haptic device scene. Although there are some common features, since they share the same origin, some of the requirements were specific for each case.

The functional requirements were then divided according to the following groups:

- Creation
- Selection
- Edition
- Calculation
- Visualization
- Publication
- Platform Compatibility

Below are presented the requirements corresponding to the set of operations available for each group.

The Editor

Creation

Name	Requirement
Create	The application should allow the user to add an object from a available set (e.g. cube, sphere, cylinder) to the scene.
Delete	The application should be able to delete a selected object.

Selection

Name	Requirement
Select	The application should permit the user to select any object on the scene.
Drag	The application should allow the user to pick an object and drag it around the scene.
Drop	The application should allow the user to drop the object on any place in the scene.
View Properties	The application should show the selected object main properties.

Edition

Name	Requirement
Position	The application should enable the user to change the object position on any of the three axis lines of the Cartesian coordinate system.
Scale	The application should enable the user to alter the object scale on any of the three axis lines of the Cartesian coordinate system.
Rotation	The application should enable the user to rotate the object on any of the three axis lines of the Cartesian coordinate system.

Visualization

Name	Requirement
Choose Camera	The application should grant the user the possibility for him to choose the best suited camera for the scene, from a set of predefined cameras.
Free Camera	The application should smooth the control of a free cam, allowing the user to better explore the presented scene.
Zoom	The application should be able to zoom in and out the scene, from the selected camera.

Publication

Name	Requirement
Publish	The application should allow the user to publish the scene anytime he intends to.

Platform Compatibility

Name	Requirement
Web browser target	The application must be supported by the following browsers versions or above: Internet Explorer 9.0, Google Chrome 16.0, Mozilla Firefox 8.0, Opera 11.52 and Safari 5.1.2
Operative system target	The application must be supported by the following Operative Systems versions or above: Windows 7and OSX Lion.

The Virtual Laboratory

Edition

Name	Requirement
Inclination	The application should allow the user to change the ramp inclination.
Distance	The application should permit the user to select the initial distance between the released object and the sensor.
Mass	The application should enable the user to select the object's mass.
Drag	The application should allow the user to select the drag coefficient between the object and the ramp.
Gravity	The application should let the user choose the gravity of the scene.

Calculation

Name	Requirement
Instantaneous Velocity	The application should be capable of calculate the instantaneous velocity of the released object.
Time Variation	The application should be able to calculate the time between the moment the object is released and it activates the sensor.
Kinetic Energy	The application should be able to calculate the kinetic energy of the system.

Visualization

Name	Requirement
Play	The application should enable the user to play the scene, observing the system's behaviour.
Choose Camera	The application should grant the user the possibility for him to choose the best suited camera for the scene, from a set of predefined cameras.
Free Camera	The application should smooth the control of a free cam, allowing the user to better explore the presented scene.
Zoom	The application should be able to zoom in and out the scene, from the selected camera.

Publication

Name	Requirement
Publish	The application should allow the user to publish the scene anytime he intends to, making it impossible to make further changes to the scene.

Platform Compatibility

Name	Requirement
Web browser target	The application must be supported by the following browsers versions or above: Internet Explorer 9.0, Google Chrome 16.0, Mozilla Firefox 8.0, Opera 11.52 and Safari 5.1.2
Operative system target	The application must be supported by the following Operative Systems versions or above: Windows 7and OSX Lion.

The Haptic Device Laboratory

Selection

Name	Requirement
Select	The application should permit the user to select any object on the scene using the haptic device and pressing the button #0.
Drag	The application should allow the user to pick an object and drag it around the scene using the haptic device while having the button #0 pressed.
Drop	The application should allow the user to drop the object on any place in the scene using the haptic device and releasing the button #0.

Edition

Name	Requirement
Position	The application should enable the user to change the object position on any of the three axis lines of the Cartesian coordinate system using the haptic device.

Visualization

Name	Requirement
Choose Camera	The application should grant the user the possibility for him to choose the best suited camera for the scene, by cycling between the available cameras by pressing the button #1

Platform Compatibility

Name	Requirement
Operative system target	The application must be supported by the following Operative Systems versions or above: Windows 7.

4 Architecture

The application architecture shows how an application is organized, including all its components and how they work together, structured in meaningful layers. In a few words, it illustrates the application's overall composition.

The present section is divided into three sub-sections, each one corresponding to the developed applications:

- The Editor
- The Virtual Laboratory
- The Haptic Device Laboratory

As understood, although the three applications sprouted from Carbono, and therefore share a similar architecture, each one has specific architectural properties.

4.1 The Editor

The editor application consists in four main modules and each one addresses one or several specific features. Each module is composed by a set of Classes or GameObjects, as they are called by the Unity Engine, each one with an assigned behaviour script.

These are described on the Table 3 and are schematized on the Figure 3.

Module	Function
Input Manager	This module is responsible by handling every type of input sent to the application. The received command is processed and communicated to the proper module, may it be the Virtual Camera Controller Module or the Virtual Lab Controller. The input can be received by the applications GUI or by interpreting the input device signs.
Virtual Camera Controller	The user has at his disposal, five different types of camera views. Depending on the user choice, this module communicates with the Renderer module, the intended perspective view.
Virtual Lab Controller	Depending on the command received, this module is responsible to manipulate the entire virtual lab scene as well as the integration with the physics engine. The main requirements for the laboratory are handled here. The actual state of the scene is communicated in each frame to the Renderer Module.
Renderer	This module receives information of the scene state on every frame of the application runtime. Then it processes this information, generates the image from the received models and communicates to the output device what is to be displayed.

Table 3- The editor modules and respective functions

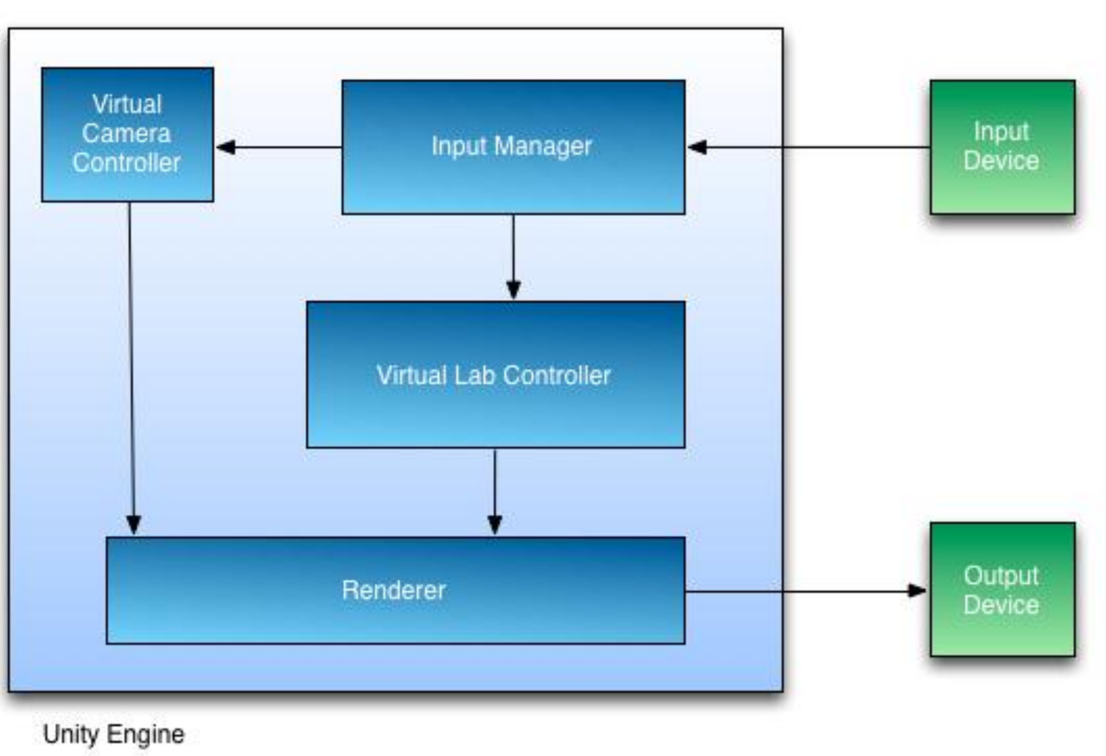

Figure 3- The editor architecture

As can be understood by the Figure 3, the Unity Engine involves all the main architecture modules. The input commands sent by any input device are handled by the input manager module. The input commands can be targeted for the virtual camera controller, if the command is camera related, or for the virtual lab controller, if the received command regards the creation or edition of the laboratory scene. The above mentioned modules after processing the received instructions, communicate with the renderer module, responsible for rendering the graphics and displaying the scene. This module in its turn communicate to the output device the actual state of the scene.

4.2 The Virtual Laboratory

The architecture of this application is very similar to the previous one. The main differences are due to the integration of the Virtual Laboratory within the FLOCK platform. Hence was created an additional module, responsible by communicating with the web browser.

This application architecture is described on the Table 4 and is represented on the Figure 4.

Module	Function
Input Manager	This module is very similar to its correspondent on The Editor application. The main difference is that besides receiving the input by the application GUI or by interpreting the input device signs, it also receives input directly from the HTML page by communicating with the FLOCK Communication module.
Virtual Camera Controller	This module has the same behaviour as the one of The Editor's application.
Virtual Lab Controller	Depending on the command received, this module is responsible to manipulate the entire virtual lab scene. On the Virtual Lab application this module edits the experimentation conditions and calculates the expected results. The actual state of the scene is communicated in each frame to the Renderer Module as well as the calculations results.
FLOCK Communication	This module interact bidirectionally with FLOCK. It can call functions on the HTML page in which the application is embedded or allows the HTML page to call functions within the application. It communicates directly with the Input Manager module.
Renderer	This module has similar behaviour as the one of The Editor's application.

Table 4- The virtual laboratory modules and respective functions

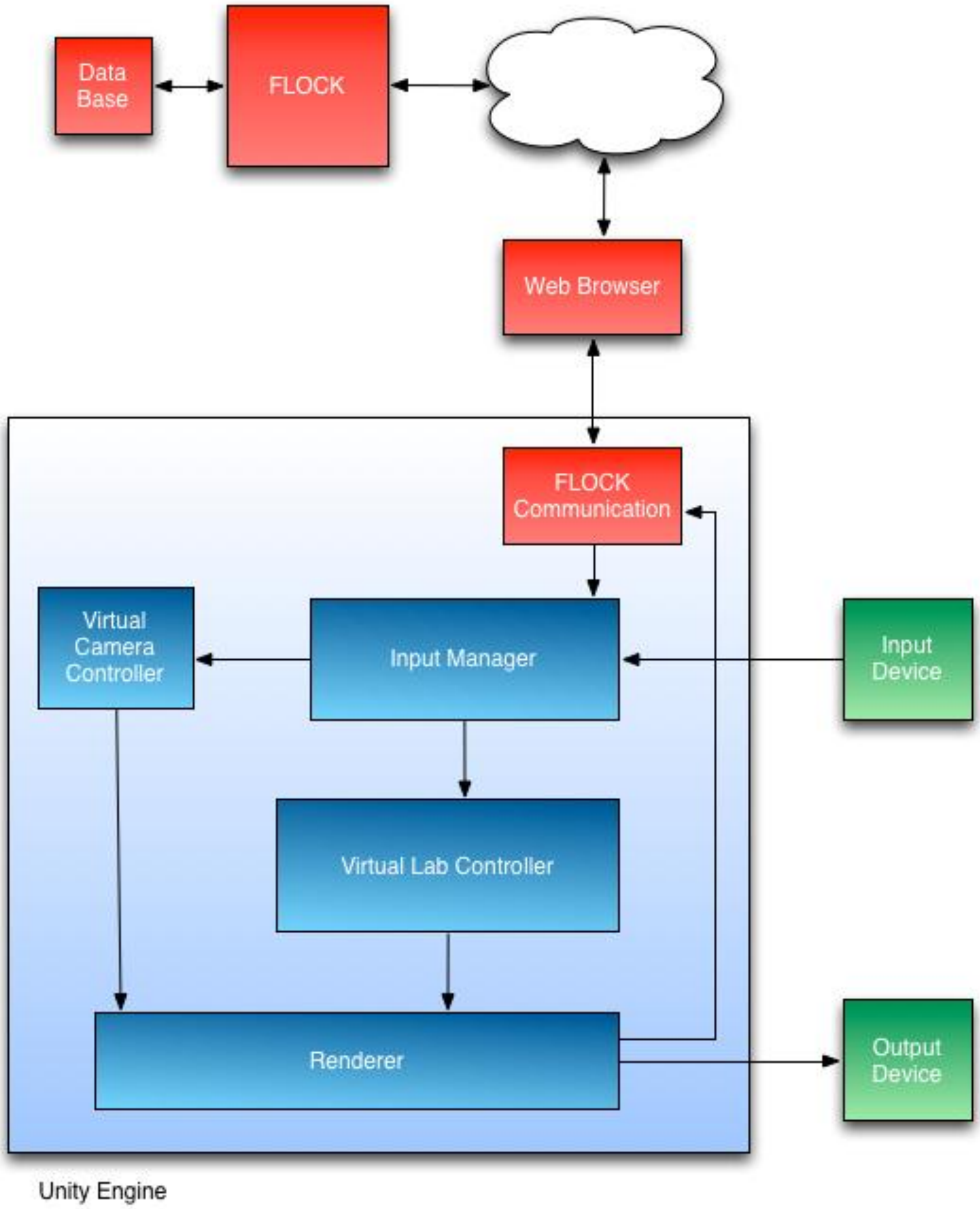

Figure 4- The virtual laboratory architecture

As easily observed the editor's architecture serve as base for the virtual laboratory as well. The main differences are due to the fact that this application is web embedded and therefore, was necessary to develop a module responsible for communicating with the web browser. Thereby, the application can communicate with FLOCK and exchange data bidirectionally. The Flock platform, can store some of this information on a data base or later use.

4.3 The Haptic Device Laboratory

This application's architecture has the same foundations as the previous ones, but with some particular features in order to communicate with the haptic device. It was necessary to create a plug-in to interact with the Novint Falcon drivers, so it could be possible to interact with the hardware device. The inclusion of the plug-in on the application is entirely handled by the Unity Engine.

This application architecture is outlined on the Table 5 and is shown on the Figure 5.

Module	Function
Input Manager	This module only handles the haptic device input commands, communicated by the Unity Engine's plug-in handler module.
Virtual Camera Controller	This module is responsible by communicating to the render module, the perspective view of the two cameras available on this application.
Virtual Lab Controller	This module allows the user to control some objects on the scene, using the haptic device.
Renderer	This module has similar behaviour to the previous ones.
Unity Plug-in Handler	This module is present on the Unity Engine and was the solution to wrap the created plug-in, responsible for the haptic device interaction, within the application.

Table 5- The haptic device laboratory modules and respective functions

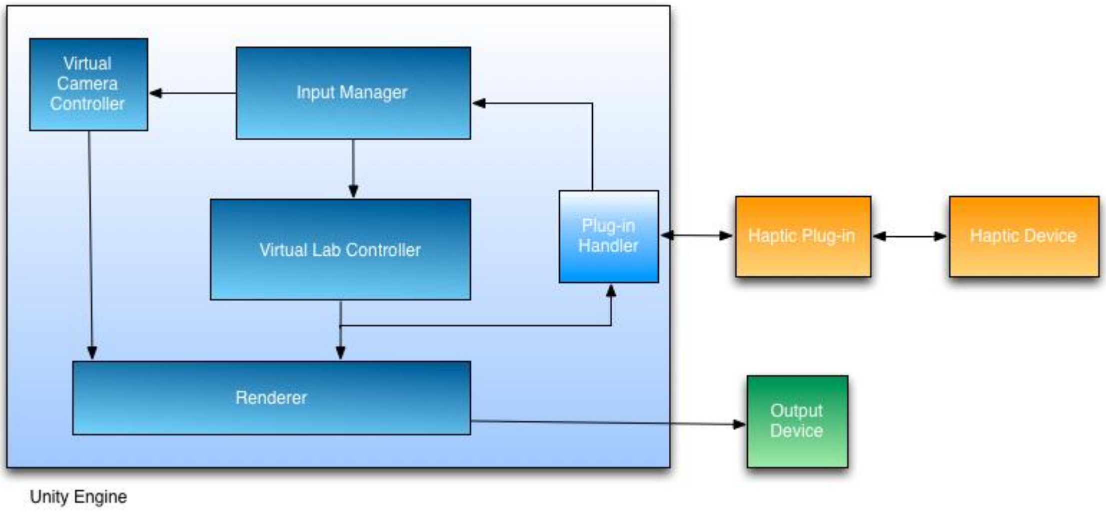

Figure 5- The haptic device laboratory architecture

Just like in the previous case, the editor's architecture serves as base for this application. In order to integrate the Novint Falcon within the Unity Engine was created a plug-in responsible for translating the haptic device actions. Taking advantage of Unity Engine plug-in handler module, the developed plug-in is easily integrated.

5 Project Planning

5.1 Methodology

Currently to achieve a successful software development it's essential to adopt the most suitable methodology for the process. Being part of a year internship, the portrayed project presented some peculiar conditions, somewhat different from those found on an enterprise environment. To this is added the fact that there is no real client and most of the decisions were taken by the project's team and its supervisor. This gave considerable room for creativity and experimentation, but at the same time required self-control and a well-structured work plan to avoid project drifts along the process. In order to circumvent this conditions was adopted an adaptable methodology inspired on the Agile[44] philosophy.

This project required a flexible environment in which the team is not restrained to follow a top-down approach but instead upgrade or revise previous phases whenever was justifiable. This became extremely important since new issues, ideas or solutions arose during the implementation phase. The selected methodology also improved the communication and collaboration between the team members. As reported earlier the followed methodology is not included on the most common Agile methods, but used some of its features and philosophy.

5.2 Plan

Following the state of the art and the requirements stage, it was outlined the project plan. It was divided into two major components: the 1st semester and the 2nd semester, since this project was part of an internship with the approximate duration of an academic year, about ten or eleven months. Soon after, it was sub-divided on the following four different phases:

- **Analysis**- consists in identifying the problem and its requirements. It embraces the searching for suitable solutions and the state of the art document elaboration. It also includes the project global organization and planning.

- **Design**- is when the identification of a fitting architecture takes place. The initial brainstorms are held as well as the first sketches and mock ups. The main features to be implemented are highlighted. This phase holds the familiarization with the chosen languages and techniques and the creation of a low fidelity prototype. The type of users and their specific issues are covered.

- **Development**- it's the phase when code start to be produced. The strategy previously outlined is followed with the intention to achieve the stated goals. Is usual to occur some strategy corrections along the way, in order to solve some issues not previewed in the previous phase.

- **Deployment and validation**- the test stage is ongoing and occurs simultaneously with the previous phase. That way is usual to emerge new problems not previously foreseen. In this phase there's a more structured and complete test period. One of the objectives is that an early version of this project is presented to real professors and students with knowledge on the embraced areas, in order to give a more enlightened feedback.

The defined plan was as it follows.

5.2.1 First Semester

The first part of the project started on September 7th of 2011, through the first meeting with the project supervisor and all the project members. This component covered the first two items of the project plan: "Analysis" and "Design".

The first weeks of the semester were focused on providing a new insight into the e-learning and Virtual Labs fields and their integration, while raising questions on what could be improved, which were the major faults, and what value it could bring towards both student and teacher.

Simultaneously was held a research in order to keep track of other available solutions for our problem.

Soon after, the project advanced into the next stage by defining its requirements, which emerged mainly from team meetings and were later approved by the project supervisor. Once the requirements were defined they were converted to mockups which later served as a support for the low-fidelity prototype.

The finishing date for this section was 24th January 2012.

The image bellow shows the Gantt diagram projecting the first semester. This information is available on the appendix: Appendix A- Project Planning

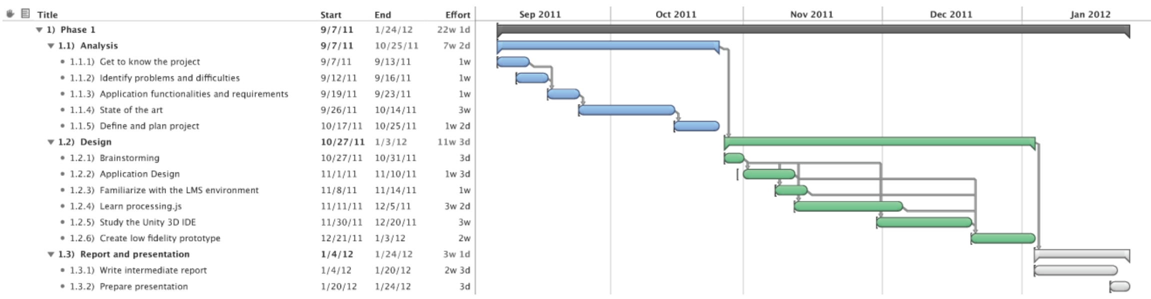

Figure 6- Gantt Diagram for Phase 1

An additional fourteen days phase was added to contemplate the writing of the intermediate report and the preparation of the intermediate presentation.

5.2.2 Second Semester

The second period was expected to start on February 10th of 2012. This component covered the last two items on the project plan: "Development" and "Deployment and Validation". It was initially defined to follow the work done previously, achieving the proposed objectives while meeting the established requirements.

The initial planning for this period was expected to be concluded around the 25th July 2012, and was as it follows:

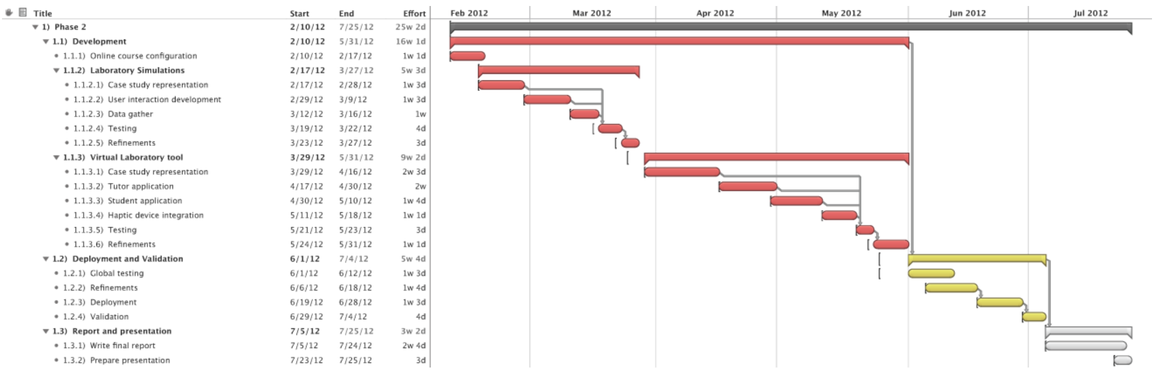

Figure 7- Gantt Diagram for Phase 2

An additional fourteen days phase was also anticipated, in order to contemplate the writing of the final report and the preparation of the presentation.

This information is available on the appendix: Appendix A- Project Planning

5.2.2.1 Revised Planning

Due to some unexpected issues and setbacks, later discussed, the initial planning for the second period suffered some significant modifications and the final deadline was postponed for about a month.

Therefore the original finishing date for this section was projected to be around 25th July 2012, but ended up to be around 26th August 2012.

The image bellow shows the Gantt diagram projecting the second semester.

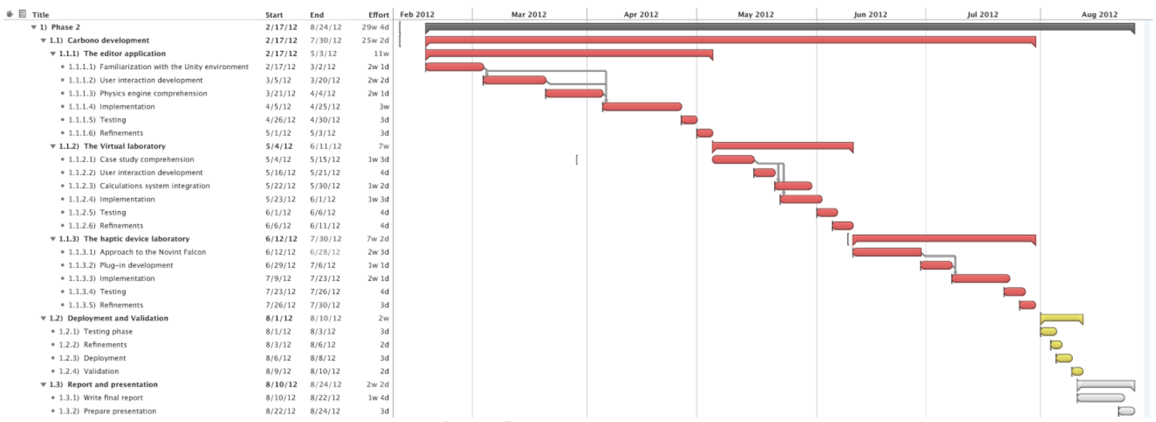

Figure 8- Revised Gantt Diagram for Phase 2

This information is available with more detail, on the appendix: Appendix A- Project Planning

5.2.3 Conclusions

The main objectives for this project were achieved, and the followed plan proved to be decisive, since it avoided major drifts and time wasting. However it was not possible to foreseen some delays, which eventually altered the planned course.

Also, the commitment made with the physics teachers caused the deployment and validation phase to be anticipated prior to the date originally scheduled. This was the only possible way to test the application in a real classroom with the respective students.

6 Deployment

The deployment phase of a project consists in installing, testing and making the application ready for use. At this point is also important to validate the product, i.e. checking that the developed applications meet all the requirements defined in the Analysis phase, and they fulfill their intended purpose. This is accomplished by identifying and testing all aspects of the applications.

6.1 Final Version

In the appendix Appendix F- Application Overview, that accompanies this report, the final result of the three Carbono applications was described, with the aid of screen captures, which illustrates the relationships between the various screens.

The following figures are screenshots of the developed applications.

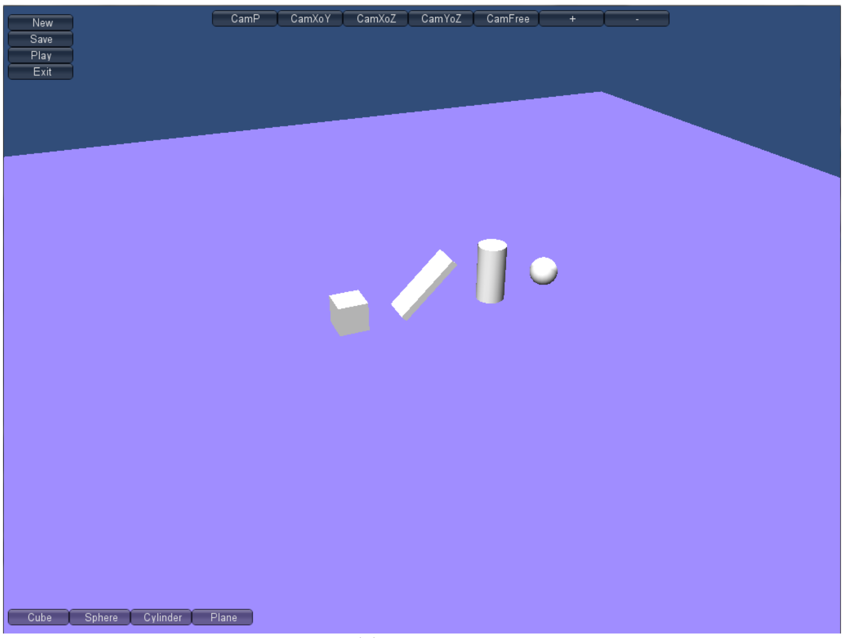

Figure 9- The editor screenshot

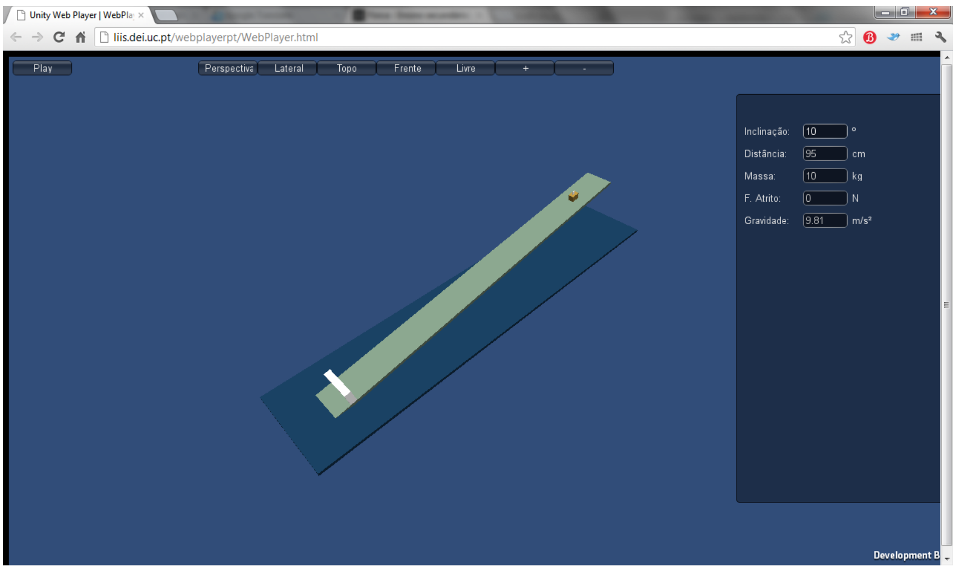

Figure 10- Virtual laboratory screenshot

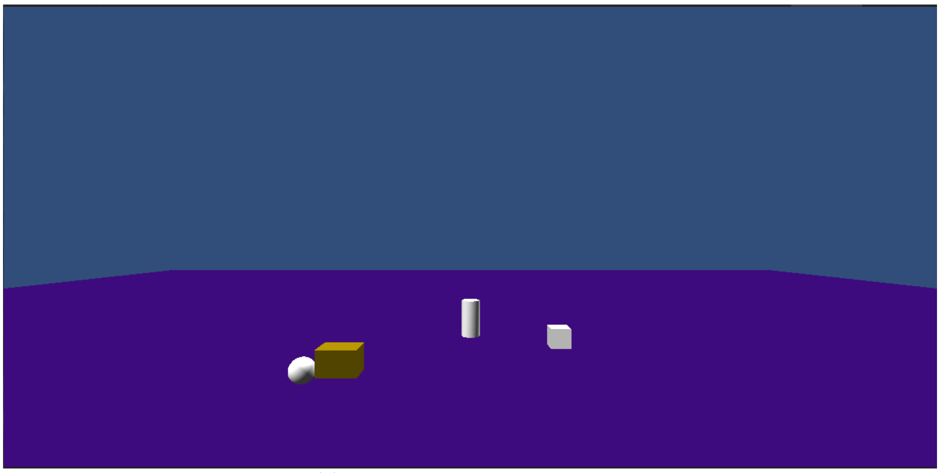

Figure 11- The haptic device laboratory screenshot

6.2 Functionality and Usability Tests

Several tests were completed since the beginning of this project. First the prototypes were tested and then, during the implementation phase, tests were done at the end of each phase. On a phase close to its conclusion, the project was tested by two classes from the school 2,3/Sec do Agrupamento de Escolas de Penacova and also by the respective teachers. These turned out to be the most important test phase since the applications were being tested by the targeted audience.

By the end of the project a more formal set of tests, with different types of users, was carried. This new set of users provided a fresh perspective and the tests were revealing, pointing to some issues that weren't detected during the development.

In the end, the users were able to use the product as intended, with relative ease, but some doubts and hesitations arose along the way. These issues were analyzed and a set of recommendations was created. Based on these recommendations some modifications were then made to the application, in order to produce a better product.

The different applications were tested with 5 persons that according to the usability guru Jakob Nielsen, is the number of people that yields the best result[43]. On the virtual laboratory application an exception was made, since there were a lot of students interested in contributing for the application testing. One of the project's collaborating teachers also provided input on these tests. The FLOCK platform was also tested on this session and the obtained results are documented on Miguel Delgado's report.

During test sessions was always important to listen to the users, especially while they were performing the tasks. It was employed a task-based think-aloud method, in which users were asked to communicate their thought processes verbally while they work. It was asked for them to vocalize what path they took to find information, what questions they had, and what surprised or confused them as they went through the application.

The main goals for this test were to determine what is or is not a usable functionality from the users' perspective. It was attempted to find information such as:

- Do users complete each task successfully?
- If so, how fast do they perform each task?
- Is that fast enough to satisfy them?

- What paths do they take in trying?
- Do those paths seem efficient to them?
- Where do they stumble?
- What problems do they have?
- Where do they get confused?
- What words or paths are they looking for which aren't supported by the navigation or copy?

The described usability tests were more focused on the applications functionalities, and not so much on their designs and user interfaces since these two complements were not developed in time for testing, as they will be discussed later on this report. However, some usability testing was made along the way, since the actual display is intended to be maintained after the design is integrated.

The test script, the results and notes of the test sessions, can be found in the appendix: Appendix G- Requirements Analysis.

6.3 Results and Findings

Overall the tests results were very positive, and none of the subjects showed difficulty or frustration on any particular steps. A careful observation over the obtained results may conclude that the implemented user interface is logical and coherent. Every time a user showed some difficulty achieving an objective, on a posterior similar task, in most cases, the user didn't reveal any difficulty at all. Also no user obtained a level 3 rate on any step (According to the rating system adopted, the level 3 would be assigned to tasks that couldn't be completed).

Below there is a more detailed interpretation of the performed tests in each application.

6.3.1 The Editor

This application obtained very positive feedback from the test session. These results can be associated to the application's practical user interface, but also to the fact that the application is still in an early version, and as such, doesn't allow very complex tasks.

Once the user comprehended the interface and the application functionality easily resolved the proposed steps.

6.3.2 The Virtual Laboratory

On this test session was taken into special consideration, the overall score of the 1st subject, since it corresponded to one of the collaborating teachers. The other nine tests corresponded to physics students, aged between 16 and 18 years old.

This proved to be a very important session because it was a great opportunity to test the potential of Carbono with its target audience and to spot its main virtues and defects.

The test results were very positive and encouraging since every subject, with more or less difficulty, was able to accomplish the proposed step.

The main difficulties that stood out were that sometimes the user took a little more time than expected to find out which form field should be edited. The project team concluded that this was due to the fact that the subjects were not familiar with the application and the form may contain too much information at once.

The overall scored obtained by the teacher was fairly positive, which resulted in a great asset of value to the project.

6.3.3 The Haptic Device Laboratory

Of the three tested applications, this was the one that obtained the lower overall scores but still fairly positive. The major difficulties denoted by the subjects were due to the fact that none of them had any previous experience with the Novint Falcon, or any other haptic device of this kind.

The difficulties with the haptic device control were not so much with the three dimensional control but more with the decision of which button to press for a determined task. There are no official guidelines for the Novint Falcon's button functions, so it was up to the develop team to decide the functionalities of each button.

In any case, after the subjects comprehended the Novint's Falcon behaviour were able to accomplish every task, resorting to the help of the test facilitator, whenever justified.

7 Conclusions

This chapter includes some final thoughts about the whole project and its accomplishments and contributions. It also includes a section where the risk contingency plan and setbacks are reflected and terminates with a section dedicated to future work.

7.1 Accomplishments and work done

From a goal perspective, it is safe to say that the internship has met the main goals defined for both semesters.

The following list presents the most important tasks, accomplished by this project, ordered chronologically:

- **State of the art analysis**
  The research conducted during the first semester about multimedia technologies and learning management systems that could represent an efficient solution for the explored problem.

- **Requirement analysis**
  The initial elicitation made, during the first semester, for the project's requirements. This analysis was later revised and updated in order to contemplate the new direction set by the project.

- **Application Architecture**
  The architecture designed for each one of the developed applications was a collaboration of all the members of the development team, in order to give the best possible response to the project needs.

- **Development of FLOCK**
  Online platform with the main purpose of hosting all the online courses built and respective contents and resources.

- **Development of Carbono as an Edition Tool**
  Edition tool designed to provide teachers with the opportunity to build their own virtual laboratories.

- **Development of a Virtual Laboratory**
  Virtual laboratory, designed with the same basis as Carbono, intended to demonstrate the edition tool possibilities, as well as testing its functionalities with the target audience.

- **Development of a Haptic Device Controlled Laboratory**
  Brief demonstration of the integration of Carbono along with the haptic device Novint Falcon.

- **Deployment and Validation**
  During the applications development phase, several tests were carried out in order to assure a consistent behaviour and a solid basis. When the application's features reached a satisfactory level, a more complete test phase was carried out,

highlighting the tests developed with the physics students and teachers of the school 2,3/Sec do Agrupamento de Escolas de Penacova.

- **Participation on a published paper**
  The paper entitled "Demonstration of Online Educational Modules with Online Experiments" written by the authors Cardoso, Alberto(1); Restivo, Maria Teresa (2); Cioga, P. (1); Delgado, M. (1); Monsanto, J.N. (1); Bicker, J. (1); Nunes, E. (1); Gil, P. (1,3) with the respective affiliations: 1: University of Coimbra, Portugal; 2: UISPA, IDMEC-Pólo FEUP, Faculty of Engineering, University of Porto, Porto, Portugal; 3: Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Campus de Caparica, Lisboa, Portugal, and a demonstration of the developed applications were presented on an exhibition session of the International Conference REV 2012- Remote Engineering & Virtual Instrumentation. It is expected that this paper will be published in a special issue of the online journal "International Journal of Online Engineering (iJOE)" – http://www.online-journals.org/i-joe/

7.2 Risk management and setbacks

The risk contingency plans plotted on an early stage of the development process, turned out to be very effective for the project's integrity. Some risks and setbacks that occurred during the development process and their respective attenuation plans are discussed below:

From the outset, the fact that this project didn't have a real client brought some advantages and some disadvantages. While opening room for creativity it also required detailed planning to keep away the possibility of losing focus on the main goals.

Due to the early withdrawal of one of the team members from the project, the one responsible for building the FLOCK platform from scratch, it was urgently needed to put in motion the provided attenuation plan. A LMS was adopted for the FLOCK foundations and only then the project could move forward. Even if the contingency plan has been effective, this represented one the project's major setbacks, since it forced a new planning and rethinking on some already defined approaches.

On January 2012, the development team attended a meeting held by the Department of Mechanical Engineering of the Faculty of Engineering of the University of Oporto. This meeting eventually changed the course of the project. The aforementioned department pitched this project with two haptic devices, the Novint Falcon and the PHANTOM Omni, creating an unmissable opportunity. The prospect of integrating a haptic device within the developed work and consequently opening the door to the augmented reality field, represented a great asset for this project. However it also brought the need for a new planning that would contemplate the haptic devices integration and a review over the state of the art document. This meeting marked the emergence of Carbono, a tool expected for creating virtual laboratory, and integrates it with a haptic device.

The late emergence of the Carbono initiative constituted another major setback. Time had always been an issue for this project, since it was part of a year internship, and the planning and development of such a complex tool proved to be very extensive for the time available. Therefore the charted contingency plan came into action. Since Carbono as an edition tool couldn't meet the expected requirements in a valid time, two new applications with the same foundations were created: the virtual laboratory and the haptic device laboratory. This was the solution established by the project's team in order to assure that the main objectives for the project were met. The virtual laboratory was essential to test the Carbono application with its target audience (physics teachers and students) and the haptic device laboratory was indispensable to take the first step into the augmented reality field.

During the FLOCK's hosting phase on the University of Coimbra's server, some issues arose due to the server's permissions and security settings. This caused some tasks to take longer than expected, particularly the integration of Carbono within FLOCK.

A standalone version of the virtual laboratory application was made available for download, in order to minimize the harm to the student's learning process in the case of an internet connection couldn't be established.

Due to some health issues, the team member José Nuno Monsanto, main responsible for Carbono's design, couldn't complete his task on a valid time. Consequently by the deployment and validation phase of the application, the design wasn't prepared to be tested. Anyhow this issue did not invalidate that the application's user interface, was tested during the performed tests.

During the integration of the Novint Falcon haptic device with the Unity IDE, it was necessary to develop a plug-in that would allow the Unity Engine to interpret the Falcon's actions. All the work developed for this project, was under the Unity3D free version which does not support plug-in incorporation. This consisted in another major setback for the project. To circumvent this issue, the project acquired a 30 days free trial license for the Unity 3D pro version. Although this version supported the plug-in incorporation, in return it offered a very tight time limitation.

7.3 Future work

An application as complex as the Carbono tool may never be satisfactorily considered as completed. Below is presented a list of possible improvements and features highlighted by the author:

- **Save and open scene**
  Currently Carbono doesn't save the actual state of the scene. This is a major fault for any edition tool.

- **Publish developed work**
  Carbono doesn't allow the user to publish his work. In the future it will be a major asset if the user can not only publish his work but also select which system conditions (gravity, mass, etc) and options (camera, controls, etc) are available after publication.

- **Provide new objects and allow model importation**
  Apart from the objects that already offer, Carbono will be more complete if provide new objects or allow them to be imported. Objects like sheaves, ropes or springs would make possible to approach other matters of the physics field.

- **Improve the bidirectional communication with flock**
  By sending to FLOCK more detailed information of the user's behaviour while using Carbono, this data can be used to alter the structure of the online course,

making it more personalized and user centered. This can represent a great base for implementing e.g. an Intelligent Tutoring System.

- **Import libraries in order to expand to other science areas**
  Carbono was idealized to be a multifaceted edition tool that could cover several scientific areas. By allowing it to import e.g. a physics library, it would add specific notions and options inherent to that specific science field. This way the tool would be shaped after the user's needs.

- **Play, replay, pause, forward and rewind system**
  During and after the creation of a virtual laboratory, the user should be allowed to observe the system's behaviour. The possibility to play, replay, pause, forward or rewind the scene would serve this purpose.

- **Technical improvements**
  Following are some technical improvements which would enhance Carbono's performance:
  - Pin or confine camera;
  - Change objects material and respective texture and consequently its system behaviour;
  - Improve Drag & Drop System;
  - Include sound;
  - Define relationship between objects (e.g. parallel, perpendicular, aggregated);
  - Implement the outlined design;
  - Make the application totally controllable through the haptic device;

Concerning the virtual laboratory, all future work is covered by the projected improvements for Carbono. For the haptic device laboratory the possible improvements are related to the control of the force feedback, since all the other main features of the Novint Falcon were covered by the project.

7.4 Final Thoughts

The overall balance of this project is positive. In the end, three different applications were developed and much was learned regarding the addressed matters and technologies. The experience also permitted to gain some insight into the workings of a real development team.

All the main features, with high or medium priority were implemented, leaving behind only the ones with low priority, which may be implemented in the future. The continuous tests made to the application throughout the duration of this project allowed to find not only usability issues but also software faults that were eventually corrected.

Some setbacks happen during the development process, and some adjustments had to be made. These setbacks translated into some considerable changes to the project's schedule. Nevertheless, due to the postponing of the project's deadline, it was possible to finish the project with satisfactory results.

The contribution made to the publication of the paper entitled "Demonstration of Online Educational Modules with Online Experiments" and its posterior presentation on the exhibition session of the International Conference REV 2012- Remote Engineering & Virtual, constituted a great asset not only for the authors but for the project itself.

One downside of this internship was the lack of a real costumer and hence not obtaining insight into the workings of a real company and of the business world.

In conclusion this project was a valuable experience that allowed the gradual learning of several new competencies as well as the consolidation of the ones already possessed.

References

1. Adobe Flash- `http://www.adobe.com/software/flash/about/`

2. Ahern, N., & Wink, D. M. (2002). Virtual learning environments: second life. *Nurse educator*, *35*(6), 225-7

3. Babu, S. C. (2007). E-learning Standards. *National Seminar on eLearning and eLearning Technologies*, *1*(1), 88-91.

4. Bonivento, C., Gentili, L. Marconi, L., & Rappini, L. (n.d.). A Web-Based Laboratory for Control Engineering Education 2 System Architecture. *IEEE Control Systems.*

5. Chi, X. (2006). A Web-based Intelligent Virtual Learning Environment for Industrial Continuous Improvement.

6. Creighton, R. H. (2010). *Unity 3D Game Development by Example*. *Development* (p. 384). Packt Publishing.

7. Díez, J. L., Vallés, M., Valera. (2009). Virtual and Remote Laboratory of the Ball and Beam System.

8. Gilibert, M., Picazo, J., Auer, M., Pester, A., & Ortega, J. A. (n.d.). 80C537 Microcontroller Remote Lab for E-Learning Teaching. *iJOE International Journal of Online Engineering.*

9. Goldstone, W. (2009). *Unity Game Development Essentials*. *Image Rochester NY* (Vol. 10, p. 298). Packt Publishing.

10. Gomes, L., Member, S., & Bogosyan, S. (2009). Current Trends in Remote Laboratories. *IEEE Transactions on Industrial Electronics, Vol 56*.

11. Hole, A. *Moodle JavaScript Cookbook*. Packt Publishing.

12. Java Applets- `http://java.sun.com/applets/`

13. Java- `http://java.com/`

14. Jin, S.-A. A. (2010). Effects of 3D virtual haptics force feedback on brand personality perception: the mediating role of physical presence in advergames. *Cyberpsychology behaviour and social networking*, *13*(3), 307-311

15. Lewis, B. A., Macentee, V. M., Delacruz, S., Englander, C., Jeffrey, T., Takach, E., Wilson, S., et al. (2005). Learning Management Systems Comparison 189-194. Informing Science.

16. Michelotti, M. B. (2011). Application of the Novint Falcon Haptic device as an actuator in real-time control.

17. Microsoft Silverlight- `http://www.microsoft.com/silverlight/`

18. Moodle- `http://moodle.org/`

19. Nwana, H. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, *4*(4), 251-277.

20. Ong, J. & Ramachandran, S. (2003). Intelligent Tutoring Systems: Using AI to Improve Training Performance and ROI. *Stottler Henke Associates, Inc.*

21. Processing- http://processing.org/

22. Processing.js- http://processingjs.org/

23. Reed, M. S., Evely, A. C., Cundill, G., Fazey, I., Glass, J., & Laing, A. (2010). What is Social Learning ? *Ecology And Society*, *15*(4), 10. RESILIENCE ALLIANCE.

24. Revans, R. W. (1982). What is action learning and what is it for ? *Journal of Management Development*, *1*(3), 64-75.

25. Rowell, E. (2011). *HTML5 Canvas Cookbook*. Packt Publishing.

26. Sensable PHANTOM Omni- http://www.sensable.com/haptic-phantom-omni.htm

27. Shiva 3D- http://www.stonetrip.com/

28. Silva, A. J., Ramirez, O. A. D., Vega, V. P., & Oliver, J. P. O. (2009). PHANToM OMNI Haptic Device: Kinematic and Manipulability. *2009 Electronics Robotics and Automotive Mechanics Conference CERMA*, 193-198.

29. Three Tank System DTS200- http://www.ict.com.tw/AI/Amira/threetank.htm

30. Kirwan B. and Ainsworth L. K A Guide to Task Analysis [Book]. - London : [s.n.], 1992.

31. Grey M L and Jensen S C Task analysis provides data for MRI curriculum planning [Report]. - 1992. - p. 22.

32. Unity 3D- http://unity3d.com/

33. Unity Technologies. (2011). UNITY: Game Development Tool. *Development*. Packt Publishing.

34. Valdes, R., Smith, D. M., & Knipp, E. (2010). HTML5 and the Future of Adobe Flash. *Analysis*, (April).

35. Vallv, J., & Costa, R. (n.d.). Laboratorio Virtual para Prácticas de Control por Computador.

36. Vargas, H., Jara, C. A., Candelas, F. A., Torres, F., & Dormido, S. (2010). A Network of Automatic Control, *6*(1). *Web-based Laboratories.*

37. Gaffney Gerry What is a scenario? [Report]. - 2000.

38. Saffer Dan Designing for Interaction: Creating Smart Applications and Clever Devices [Book]. - 2006.

39. Vargas, H., Sánchez-Moreno, J., Dormido, S., Salzmann, C., Gillet, D., & Esquembre, F. (2009). Web-Enabled Remote Scientific Environments. *Computing in Science & Engineering*, *11*(3), 36-46.

40. Watson, W. R., & Watson, S. L. (2007). An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become? *TechTrends*, *51*(2), 28-34. Springer.

41. Williams, D. D., & Graham, C. R. (2010). Evaluating E-Learning. In P. Peterson, E. Baker, & B. McGaw (Eds.), *International Encyclopedia of Education* (Vol. 3, pp. 530-538). ASTD.

42. Novint Falcon- http://www.novint.com/index.php/novintfalcon

43. **A mathematical model of the finding of usability problems** [Conference] / auth. Jakob Nielsen Thomas K Landauer // ACM INTERCHI'93 Conference. - Amsterdam, The Netherlands : [s.n.], 1993. - pp. 206-213.

44. Allen, I. E., Survey, B., & Seaman, J. (n.d.). Going the Distance

45. Agile process- http://www.agile-process.org/