MPEG-V Based Web Haptic Authoring Tool

by

Yu Gao

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc degree in
Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Nowadays the World Wide Web increasingly provides rich multimedia contents to its users.In order to further enhance the experience of web-users, researchers have sought solutions to integrate yet another modality into the web experience by augmenting web content with haptic properties. In those applications, users are able to interact with web virtual environments (such as games and e-learning systems) enriched with haptic contents. However, it is not easy for designers without proficient web programming background and basic knowledge regarding haptics, to develop a web application with haptic content enabled. Additionally, there is currently no standard to describe and reuse a well-described haptic application which can be played in web browsers. In this thesis, I present an MPEG-V based authoring tool for facilitating the development procedure of haptics-enabled web applications. The system provides an interface for users to create their own application, add custom 3D models, and modify their graphic and haptic properties. Haptic properties include the specification of collision detection mechanism and object surface properties which in turn directly affect the force simulations. Finally, the user is able to export a haptic-enabled 3D scene in a standard MPEG-V format which can be reconstructed in a web haptic player. A detailed experiment is conducted to evaluate the force simulations, application development process and design of user interface. The results not only verify my proposed methodology, but also show the high acceptance level by users with all levels of programming knowledge of the system.

Acknowledgements

I would like to give my sincerest gratitude and appreciation to my supervisor Prof. Abdulmotaleb El Saddik for his continuous guidance and support not only in academic level but also in my personal life.

Special and sincere thanks go to Dr.Haiwei Dong and Dr.Hussein Al Osman for the invaluable assistance, guidance, and feedback they provided throughout my research, as well as the revision of this thesis.

I would also like to thank all colleagues in DISCOVER and MCRLab and all my friends for their contributions throughout the research, participation in my experiments, and their help on my life.

Finally, I am grateful to my family. Their inspiration, understanding and support made this thesis possible. This work is dedicated to them.

Contents

List of Tables

List of Figures

Glossary of Terms

2D Two Dimensional

2.5D Two-and-a-half-dimensional

3D Three-dimensional

API Application Programming Interface

DOF Degree of Freedom

GUI Graphical User Interface

HD High-definition

Hz Hertz

IDE Integrated Development Environment

IM Instant Message

I/O Input/Output

MCRLab Multimedia Communications Research Laboratory

PC Personal Computer

QoE Quality of Experience

SMS Short Message Service

SOAP Simple Object Access Protocol

USB Universal Serial Bus

VR Virtual Reality

XML Extensible Markup Language

Chapter 1

Introduction

1.1 Background and Motivation

Web applications, containing a wide range of data and multimedia content, are extremely popular in recent years, due to the ubiquity of web browsers and their inherent support for cross-platform compatibility. The performance improvements of graphics hardware, as well as new techniques in web graphic rendering, have allowed the creation of complex virtual web environments. Such applications include games, social networks, mobile applications, Smart TV platforms, etc. They utilize the large quantity of multimedia content provided by the World Wide Web such as text, images, videos, audio and animations. Haptics, meaning sense of touch in Greek, refers to the science of manual sensing and manipulation through touch, via the human body. It is used in a wide range of domains including medical, learning, military, gaming, and communication media [55]. In the field of haptic technologies, the sense of touch can be seen as a bidirectional flow between haptic interfaces and human bodies, which provides unique sensory information of the users interactions with a virtual environment [31]. Figure 1.1 from El Saddik et.al [50] shows the bidirectional flow of information defining the distinguishing feature of haptic technology. In recent years, the topic of transmitting the sense of touch over the internet through the use of various applications has become increasingly more popular

[40]. By integrating haptic contents to the web, users can get a realistic experience of touch in his or her sensory system by interacting with virtual environments enriched with haptic contents such as haptic-enabled games [27] and E-learning systems [36]. Several works have sought to integrate haptic content into web applications in order to enrich the applications with an added layer of information exchange modality, specifically through the sense of touch [30]. Al Osman et al. propose the enablement of web haptics through an HTML5 [11] web haptics plugin [38] (abbreviated as H5H), which aims to promote the wide deployment of the haptic modality in web applications by open sourcing the plugin, and thus allowing access to a wide audience of interested developers. The plugin is a javascript API (Application Programming Interface) that supports numerous web platforms and haptic interfaces by exposing an existing haptic library to web browsers. With the API provided, developers can create web applications enabling force feedback by sending touch events to the haptic device. Furthermore, for an experienced web graphic programmer, a virtual environment in which users are able to explore and feel the virtual objects is built by enabling collision detection. The high level representation of the plugin is shown in Figure 1.2. Another work named HapticWeb [32] proposed a

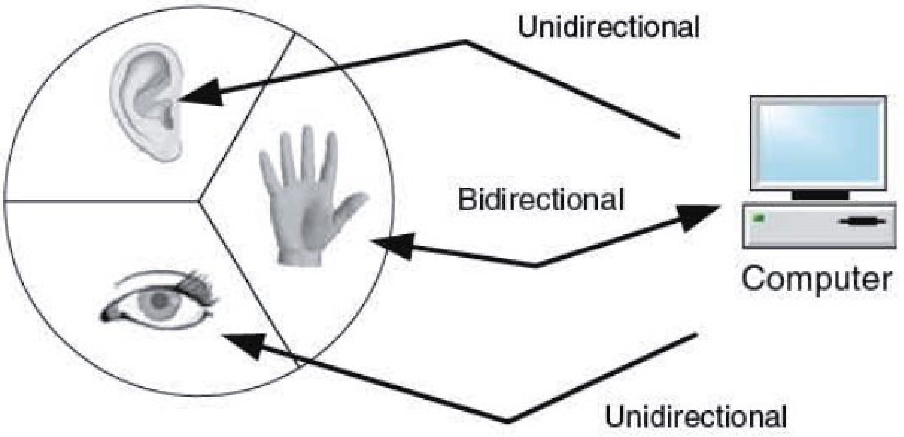


Figure 1.1: Bidirectional flow of information in haptics

script based framework for haptic interaction on the Web by combining a virtual reality engine, eXtreme Virtual Reality (XVR) [4], with a haptic rendering engine and haptic interfaces. It provides low level control over 3D graphics and haptic rendering for developers by publishing an API module supported by core XVR, which allows developers to integrate prepared script programs and graphic resources with a plugin for 3D graphics in the browser. Besides, reference [45] presents a simplified authoring system allowing the integration of three dimensional haptic contents into the World Wide Web using a modeling language called Virtual Reality Modeling Language (VRML) [24] for the creation and manipulation of three-dimensional content. It provides the haptic navigation interface for users to navigate 3D scenes in web with the help of force feedback. With the plugin system, the web pages with embed tags of VRML content can be viewed haptically in a Netscape browser [18]. Besides, several systems [46][41] [54] enable the access to internet of visual impaired people by providing a browser enhanced with haptic effects. However, the development of haptic-enabled web applications demands that web developers be skilled in both web 3D and haptics programming, which is not always the case. Also, it is complex and time-consuming to complete the development process of such applications. This process includes the creation of a virtual environment in a browser, the addition or modification of 3D models, the definition of animations for each virtual object, the assignment of haptic properties (such as stiffness, static friction, and dynamic friction) for the haptic-enabled objects [33], the implementation of collision detection

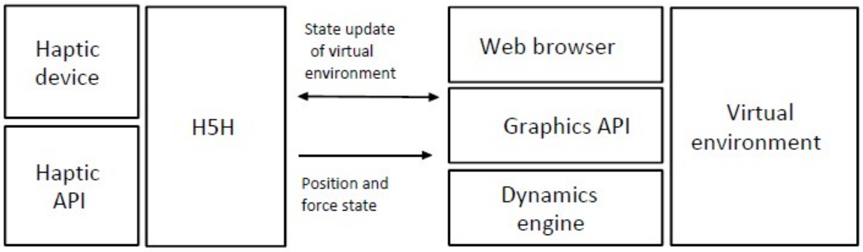


Figure 1.2: Conceptual overview of H5H plugin

algorithms and the specification of collision events, and the synchronization between the graphic and haptic contents. Additionally, there is a need to improve the portability and reusability of such applications, most of which are coupled with a specific device and corresponding API. In this thesis, we present a web-based authoring tool that can build haptic-enabled web applications in a simplified and standard way. It also provides an import/export engine for fully described haptic applications in MPEG-V [16] xml format.

1.2 Existing Problems

Due to the different objectives of each system, existing works on haptic content web integration have several problems. The problems analyzed are summarized as follows:

- **Lack of authoring capability** Most systems appear as plugins for browsers to enable the support of certain haptic APIs in web. They do not have the authoring capability to let users assign haptic properties to virtual objects in the webs virtual environment. Without authoring capability, such systems can only be used by experienced researchers and developers with a good knowledge of haptic programming.

- **Lack of standardization** The framework and scripting language used by each system differs, thus limiting users to specific graphic APIs and haptic devices. It also causes a lack of reusability for haptic applications based on these different systems.

- **Outdated formats** Works such as HapticWeb use eXtreme Virtual Reality (XVR) and Virtual Reality Modeling Language (VRML) formats which are not supported by current web3D standards. Most web application developers are not familiar with these formats since they are not naturally used in web development. Additionally, these formats only support a limited number of haptic interfaces.

- **Inconsistent descriptions** Most systems do not describe haptic applications in a uniform way, and some rely on their own framework, which is not widely supported. Nowadays there are many unified web multimedia standards such as MPEG-V, which provide architecture and associated information to define a complex virtual environment and enable interactions between virtual worlds and real worlds. A standard way of enabling haptics on the web is to combine such standards with an existing haptic framework in order to integrate haptic contents with other multimedia contents.

1.3 Objective and Contribution

In this thesis, we design and implement an authoring tool to simplify the development of haptic applications in web. It allows users without a haptic programming background to build a haptic application with a 3D haptic-enabled virtual environment by simply importing 3D models for each virtual object, and assigning haptic properties to it in a graphical interface on web. The system also provides a uniform standard of describing a haptic application using an MPEG-V xml standard. With the help of our system, designers can create their homemade haptic applications and export them as description files to the local machines, and end users can play them with supported haptic devices and feel force interactions enabled in the haptic web applications. The main outcomes of the research work are summarized in the following points:

- **Haptic application authoring tool** The proposed system enables the creation and manipulation of haptic contents on the web. It allows users to create a haptic-enabled virtual world in web browser through the upload and manipulations of 3D objects in the scene. In this process, user can define the haptic interactions in the application by assigning haptic properties including stiffness, damping, static and dynamic frictions to the virtual objects. Once the design of haptic application is completed, data of the virtual world is stored in an MPEG-V file which can be reused and rendered in a web haptic player. The authoring tool is also portable for mobile computing in different platforms.

- **Simplified development of haptic applications** As mentioned above, our system enables users, other than researchers in the field of haptics and experienced developers, to easily build a haptic application. I provide a standard and simplified developing process of five steps: First, users choose the mode of development according to the type of haptic application they are planning build. Once the mode is selected, users import 3D models by uploading local 3D model files into the browser. Then, users assign graphic and haptic properties to the virtual objects. After that, the applications are exported into an MPEG-V description file. Finally, in order to test and play the application, users can simply import the description files to the haptic player in the system.

- **Design of an MPEG-V haptic descriptor** MPEG-V is an xml standard that provides descriptions for multi-sensorial content to control multimedia presentations by using sensory devices in a virtual world and its associated interactions [15]. With the help of MPEG-V, the sensed information, such as haptic properties, are described and transformed to sensory effects such as haptic forces and applied to the real world using sensory devices (e.g., haptic interfaces). In our system I designed an MPEG-V haptic descriptor used to describe the haptic applications. For each virtual object, the graphical information such as location, scale, rotation angle, and haptic coefficients such as stiffness, friction, and damping are described in an MPEG-V format, which is easily preserved and extracted by any xml parsers.

- **Design and development of a web haptic player** I present a web haptic player, along with the authoring tool in the system, in order to import an MPEG-V file exported by the authoring tool and play it as a haptic application by the supported haptic devices. In the haptic player, end users can explore the scene by a controlled proxy using a haptic device, and feel different force feedback when the proxy moves into a virtual object in the scene. The haptic and graphic rendering loops work together, with the haptic device, to make the scene feel flawless.

- **Development of an MPEG-V haptic parser in web** An MPEG-V haptic parser is attached to the web haptic player to parse the static information stored in the MEPG-V xml file and extract specific data in graphic and haptic schemes, to build a dynamic haptic-enabled virtual environment in web.

- **Integration of physics into haptic applications** In applications for gaming and educational purposes, it is important to apply real world laws of physics into the virtual environment to enable real world feelings, both graphically and haptically. In our system, I combine the haptic applications with existing graphic physic engines to create physic effects such as velocity, gravity, and collisions between virtual objects, and make them not only appear real in the graphic scene but also feel real. I also enable collision detection to ensure force feedback events happen when collisions occur in the scene, to simulate a better sense of hapto-visual effects in the virtual world.

1.4 Thesis Organization

This thesis is organized as follows:

- **Chapter 2** presents a review of the literature of related works. It discusses similar works in haptics and on the web, including haptic authoring tools and graphic authoring in web, the different methodologies to integrate haptic contents into web, different types of haptic content authoring, and web-based graphic content authoring. We then conclude the motivation of our research.

- **Chapter 3** gives a detailed introduction of the proposed system. It starts with a component overview, presents our proposed methodology to facilitate the development of a haptic application, and describes the MPEG-V for the web haptic applications. It then describes the design of two types of 3D haptic scenes and the properties of the objects inside. After that, it introduces algorithms used to enable collision detections in the scene, simulating different types of forces in the system. At last, it concludes with the steps needed to develop a haptic application in web using our system.

- **Chapter 4** describes the implementation and evaluation process. It first discusses the implementation details of the proposed system and a HTML5 game created by the authoring tool as the proof of concept project. Then it describes the evaluation of force simulations generated by the system. Lastly it discusses a usability study performed with developers who build applications using our tool and gave their comments about the experience.

- **Chapter 5** summarizes the contributions in this research work and proposes conclusions. It also discusses the future works that could further contribute to this field.

Chapter 2

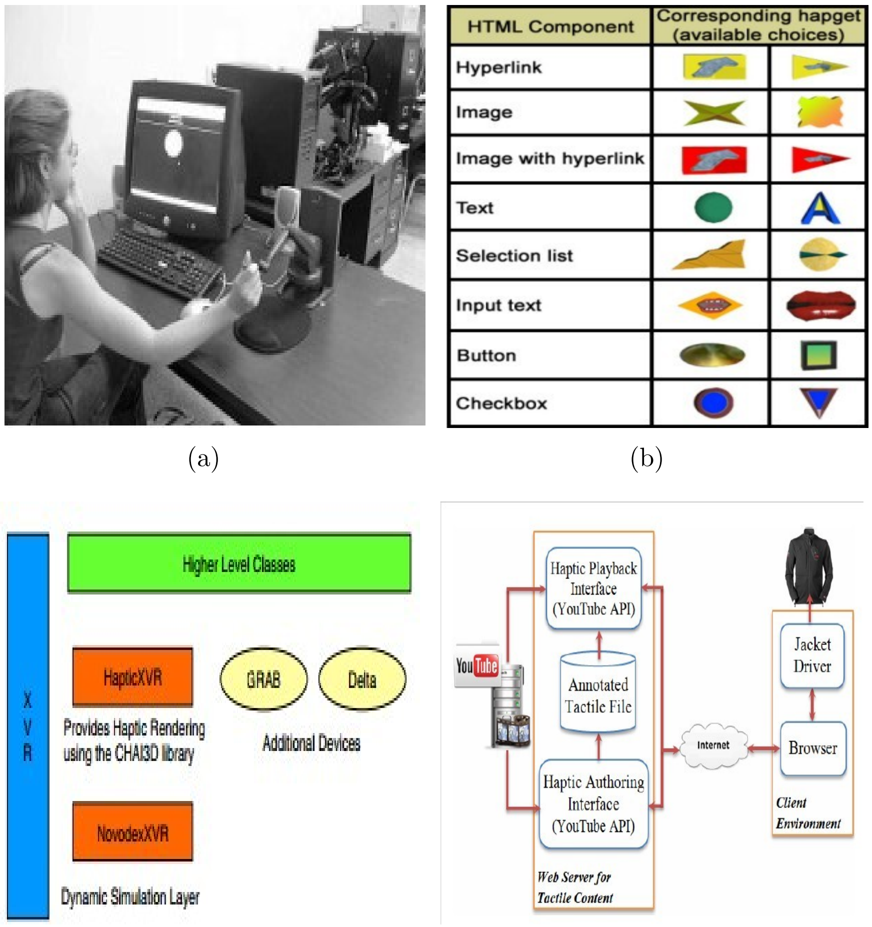
Related Works

2.1 Integrating Haptic Contents to Web

The sense of touch, containing a bi-directional flow of information and energy, is one of the most complex feelings in the human body. There are two kinds of touch sensations: kinesthetic and tactile [37]. Kinesthetic is related to the feeling of motion originating in human muscles, tendons and joints. Tactile refers to pertaining sensations of pressure and temperature or pain by the skin. The haptic technology, an emerging interdisciplinary field focused on human touch sensory and the development of touch related computer-controlled systems, is in its next step of integration with the World Wide Web. The Word Wide Web plays an important role in most peoples daily life for its easy access to a wide range of multimedia contents, from text and pictures to audio and video [53]. Such multimedia data is transmitted over numerous web applications that use web browsers as clients. Also, the new version of the web markup language HTML5[11] aims to improve the support of the latest multimedia, by keeping it easily readable by humans and consistently understood by computers and devices. With the advent of HTML5, programmers can create richly interactive environments within browsers. Included in the list of new features are native audio, video and animations, as well as improved error handling. Also, the recent techniques such as HD video, 3D video and high quality audio appear continuously. Thus, people can have a rich multimedia experience while exploring internet content in web applications. However, improvements in user experience obtained by advancements in the quality of audio and video have reached their limits. To allow users to experience a more realistic internet exploration, many researchers are merging sensory information, such as touch, into web applications [44]. To achieve this goal, we can integrate haptic contents to the internet in order to make a haptic-audio-visual environment available in web. Such applications consider haptic as a new media channel in a multimedia system [48]. Following are several approaches used to integrate haptic contents to the web.

2.1.1 Simplified Authoring of 3D Haptic Content for the World Wide Web

It is a system for the simple authoring and displaying of 3D haptic contents in web pages [45]. In the system, users can create three dimensional content using a modeling language called VRML. Such contents are displayed haptically by the internet browser. The system comes with a haptic browser plugin for parsing and processing VRML scripts with basic elements such as nodes and meshes, and an openGL [20] graphic rendering engine. The haptic scene can thus be displayed haptically. The system allows the web developer to create haptic contents by writing VRML contents as nodes of three major types: appearance, geometry, and transform grouping. Once the VRML script is written, the developer can embed it by adding an EMBED element into the web page. Figure 2.1(a) shows a user integrating with a web page in which a haptic sphere has been embedded. This system aims to facilitate the creation of three-dimensional haptic content for the web.



(a) (b)

(c) (d)

Figure 2.1: Related works of integrating haptic contens to web: (a) User interacting with 3D haptic contents in web browser (b) Mapping methodology of haptic navigation (c) Conceptual overview of the hapticWeb framework (d) Tactile content web service of haptic Youtube

2.1.2 Haptic Navigation in the World Wide Web

Blind people can explore a web page by touch sensation with the help of haptics[57]. A methodology for mapping a web page onto a haptic-enabled virtual reality environment was proposed by Nikolaos et al [47]. According to the mapping methodology, a 3D virtual scene is created and each HTML element included in a web page is represented by a 3D object that contains haptic information. The mapping strategy of HTML elements is shown in Figure 2.1(b). Based on this strategy, a 3D haptic browser is developed to transform certain HTML components from a given URL, and give blind users the chance to explore the scene through the use of a haptic device. By combining haptics with elements of the HTML page in the proposed haptic web navigation, the user experience is enhanced in different ways: a normal sighted person benefits from a haptic information channel combined with visual and audio data in the interaction with the virtual reality scene. By touching and hearing, a visually impaired person is also able to explore the virtual reality scene. Also, the visual channel reinforced by the sense of touch enhances the experience of a person with a moderate visual impairment.

2.1.3 HapticWeb

HapticWeb is a haptic toolkit for the development of immersive and Web-Enabled Games [32]. It appears as a script based framework for the development of applications that enable haptic interactions with additional modules of haptic and graphic interfaces. A conceptual overview of the hapticWeb framework is illustrated in Figure 2.1(c). It is based on the Virtual Reality engine eXtreme Virtual Reality (XVR), which enables several core 3D graphic features, along with a deployment strategy for the distribution of web applications in a simple virtual machine. Based on HapticWeb, developers are able to control the object models of haptic, and access graphic resources through low level OpenGL commands. The graphics and haptic rendering are therefore completely controlled by the developer, and he is responsible of the logical design and the definition of the interactions between the models. In the execution process, the application is rendered by a plugin for graphics inside the browser, which loads the script program and the associated 3D model. Haptic rendering is done by CHAI3D[2] haptic API, which supports multiple haptic interfaces.

2.1.4 Web-Based Touch Display for Accessible Science Education

Immersion Corporation [8] has developed a low-cost force-feedback computer mouse to allow visually impaired people to seek the scientific world. The product, containing haptic display functionality with improvements to accessibility, costs less than 100 dollars. A system for incorporating haptic feedback into e-learning was developed to help improve the quality of the physics curricula in college and high school [35]. The system includes haptic simulations in the force-feedback mouse with the development of software over the World Wide Web. The system was evaluated by a panel of educational experts and visually impaired students who where invited to use the system and give feedback. The researchers developed a demonstration curriculum module , which guides the student into several tutorial web pages and lets them go through an experimental stage of the laboratory, which aims to give blind students the understanding of the behavior of the electric charge happening on the surface of a sphere, using the force-feedback mouse. The electric charge in a particular point of the scene was simulated by the haptic device generating a force, either attracting or repelling the users hand from the sphere, when he clicked the mouse button. The data analysis stage, combined with the auditory information, then began to let users study and analyze the data collected to get a better understanding of the lesson.

2.1.5 Adding Haptic Feature to YouTube

Online videos are an important method of presentation of multimedia on the World Wide Web. Youtube [26] is the largest online video site and provides rich APIs for developers to integrate numerous multimedia prototypes into the video. Researchers in MCRLab of the University of Ottawa [17] have built a web-based framework that allows users to create tactile contents, which can be simulated by a wearable tactile device force [42]. They can feel the force feedback on the tactile jacket while watching the Youtube video. The system is comprised of a web-based authoring tool for the manipulation of tactile contents, as well as a web browser including a device driver for the activation of tactile feedback on the haptic jacket. The tactile information, composed of timestamps and tactile intensity as well as magnitude of actuation, is stored as a sequence of arrays in an XML schema. Online viewers wearing the haptic jacket can feel the haptic feedback by exploring the video with tactile content using the custom made web browser. A block diagram of the tactile video web service is shown in Figure 2.1(d).

2.2 Haptic Authoring Tools

Like other multimedia contents, the creation of haptic contents for specific haptic systems needs a detailed methodology to define the presentation format of associated properties and values such as stiffness and frictions, and allow it to be stored in a database. Such methodologies, including haptic applications meta language (HAM-L) [43], MPEG-4 [14], MPEG-V[16], MPEG-7[15] etc, have already been done by researchers in the field of haptics. According to those methodologies, a professional tool for the mass production of interface independent haptic contents[48] is needed. Thus far, several works[34] have been done to allow users to create and manipulate haptic contents for a wide range of purposes.

2.2.1 HAMLAT

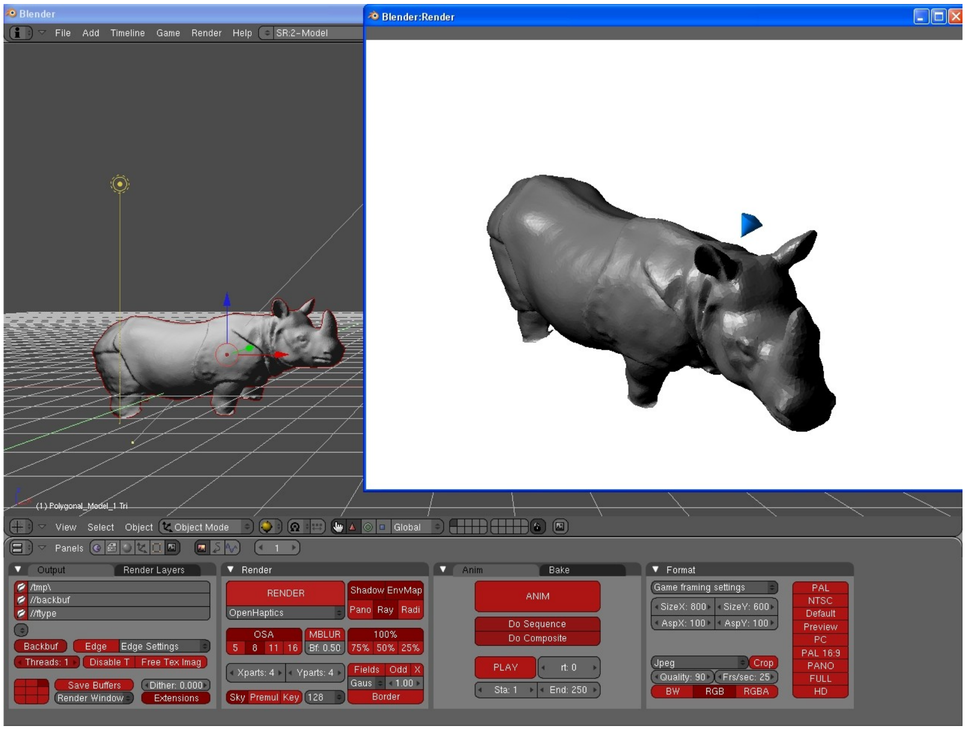


Figure 2.2: A snapshot of HAMLAT editor

HAMLAT[51] is a haptic authoring tool which was presented by researchers in the MCRLab of the University of Ottawa. The authoring tool, based on the haptic application meta language (HAML), is an extension of a desktop modeling platform Blender[1] and is intended to ease the development process of haptic applications and allow users without programming skills to design and implement a haptic application rendered both visually and haptically. As shown in Figure2.2, the system extended the user interface components of Blender to add and modify haptic properties, where users can apply haptic properties including stiffness, damping, and friction to the virtual objects in the graphic scene. After the creation and modification process, users can export the application into an HAML xml file, which contains haptic attributes along with general application parameters, including haptic support and copyright information, in the same way they modify it. It is then stored in the local machine. After that, users can import the HAML file generated earlier in order to play back the application. An additional renderer is plugged into the system so that the scene is constructed as a hapto-visual scene in which users can explore with the use of a proxy controlled by the haptic device, and feel the shape of the virtual objects. The current version of HAMLAT supports only static scenes.

2.2.2 Authoring Framework for Haptic Broadcasting

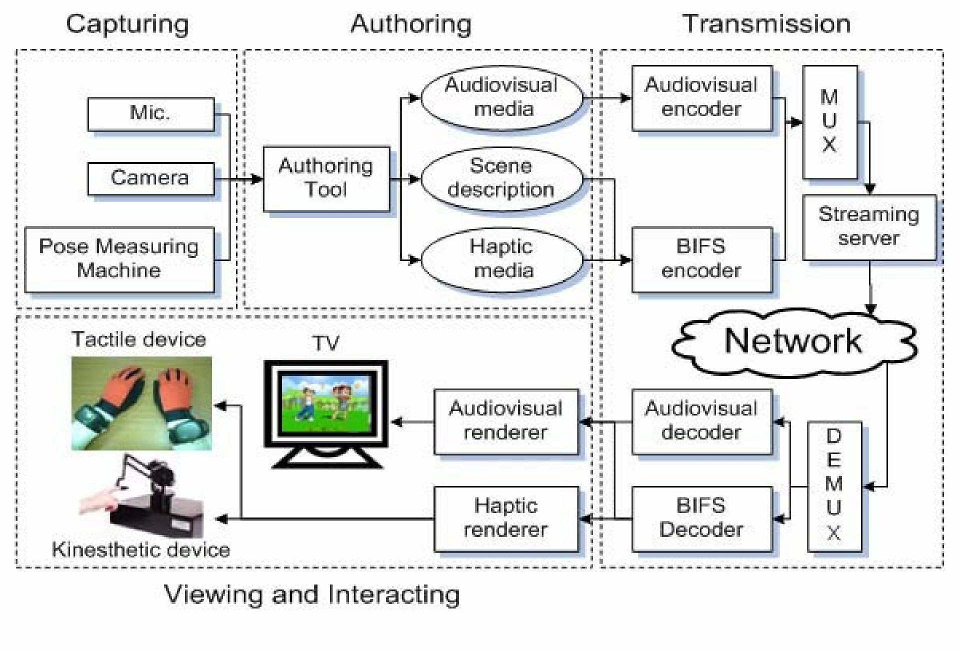


Figure 2.3: Overview of haptic broadcasting system

Authoring for haptic broadcasting [39] was proposed by Jongeun et.al and was intended to allow broadcast viewers to have the experience of synchronized passive haptic feedback while watching audiovisual media. The overview of this system is shown in Figure 2.3. The framework extends MPEG-4 BIFS[29] (Binary Format for Scenes), a standardized format providing the presentation of a wide range of multimedia contents, by adding kinesthetic and tactile playback nodes into the broadcast media via the associated authoring tool provided by the system. The node in the MPEG-4 description contains information including file type, node data type and node name. Once the haptic-enabled broadcast contents are authored by the system, there are two scenarios for the rendering of a stream context. If the video is authored with a kinesthetic playback in a musical conducting lesson, the viewers are guided to learn the motions by experiencing the gestures from the PHANTOM[5] device while watching the musical video. In the scenario for tactile playback, authors of haptic contexts can draw tactile image sequences on a canvas, and a movie with tactile feedback can be displayed by the screen and a haptic glove with a Bluetooth module inside.

2.2.3 Haptic communication Authoring Tool for an Instant Messenger

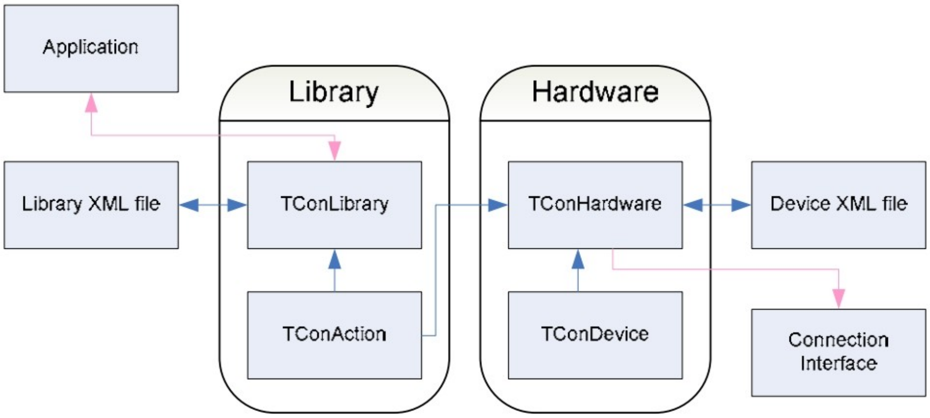
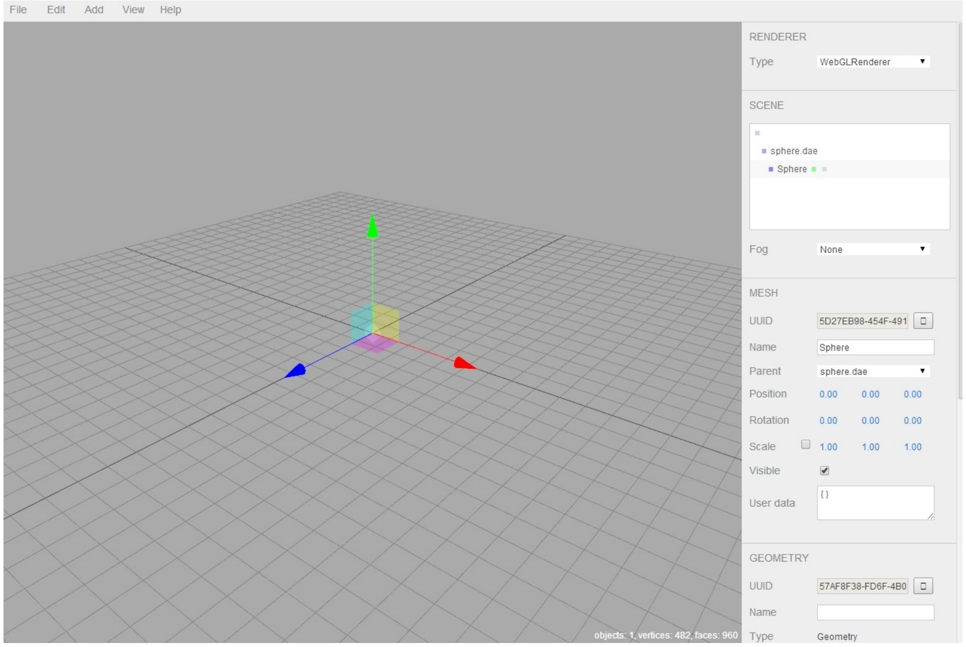


Figure 2.4: Architecture of touchCon authoring tool

In the area of online chatting, Youngjae et. al proposed a bidirectional haptic authoring framework for a haptic-enabled instant messenger [56]. The system allows users to send a haptic emotional context called touchCon[23], which can be authored by a timeline-based authoring tool. The architecture overview of the touchCon authoring tool is shown in Figure2.4. Researchers enabled bidirectional haptic communication by synchronizing the haptic behavior of both sensors and actuators using touchCon. With the help of the touchCon authoring tool, everyone can easily create his or her own haptic expression and use it to let friends living miles away feel their emotion while chatting with them through a peer to peer instant messenger. This framework is supported by haptic hardware, consisting of both sensor and actuator, and contains specific XML schemas that support haptic devices and TouchCon interactions.

2.3 Graphic Authoring Tool in World Wide Web

In recent years, the performance improvements of graphics hardware, as well as the new techniques in web graphic rendering, has allowed the creation of complex virtual web environments in modern web applications. To ease the creation/modification of the 3D scene and reuse it for the development of multimedia applications in web, we need a professional tool that supports different 3D formats. Thus far, the most popular Web3D standard for web pages and web applications is WebGL, proposed by Khronos Group [25]. It is a cross-platform web standard based on OpenGL ES 2.0. It renders interactive 3D graphics and 2D graphics within any compatible web browser, without the use of plugins, by integrating into the canvas element of HTML in web. However, most desktop applications for 3D creation and rendering do not support the new web 3D standards. For the above reason, several graphic authoring tools are proposed by researchers.



Type: WebGLRenderer

sphere.dae
Sphere

Fog: None

UUID: 5D27EB98-454F-491
Name: Sphere
Parent: sphere.dae
Position: 0.00 0.00 0.00
Rotation: 0.00 0.00 0.00
Scale □ 1.00 1.00 1.00
Visible ☑
User data: { }

UUID: 57AF8F38-FD6F-4B0
Name:
Type: Geometry

objects: 1, vertices: 482, faces: 960

Figure 2.5: Snapshot of threejs editor

2.3.1 Threejs Editor

After the wide deployment of the new 3D rendering technique called WebGL in HTML5, plenty of third party APIs were released to provide enhancements and ways to simplify it. Threejs[22] is a framework based on WebGL and defines its own format to describe a 3D scene. The official threejs site provides a web-based 3D editor able to add simple graphic models such as planes, spheres, and cubes into graphic scenes, exporting them into a three.js standard format[21]. The tool, as is shown in Figure2.5 provides a graphical user interface and allows users to modify the properties of the 3D objects in order to perform basic operations such as clone, convert, delete. Users can also select from several renderers, including WebGL renderer, WebGL3 renderer, Canvas renderer, software renderer, and SVG renderer, which are used to render the 3D scene to set the rendering engine and preview their effects. The tool is limited to the API and file format provided by threejs.

2.4 Comparison and Conclusion

To better understand and compare the related systems described in previous section, we created a table that indicates the following characteristics of the stated works:

- **Objective** General objective of the research work.
- **Type of System** Description of the system. If it is portable or independent of other systems.
- **Authoring ability** If the system allows user to author haptic contents.
- **Web-based property** If the system is web-based or if it has a web-based component.
- **Device independent** If the system is independent for different haptic devices.
- **Targeted Group** What kind of users (designer, viewer, blind people etc.) the system are designed for.

From Table 2.1, it is clear that there is research that focuses on haptic integration in Web and other research that focuses on haptic content authoring. However, there are only few of that aim at integrating haptics with haptic content authoring in Web. The HAMLAT system allows users to author a great deal of content in a hapto-visual scene. Unfortunately, the authored 3D scene files are not portable in the World Wide Web, and require users to install the graphic modeling platform Blender. The work of adding haptic features to Youtube is a closely related research, which provides Web-based haptic content authoring in online movie viewing. However, the system is limited to tactile content authoring in videos on the Youtube website, and it is not device independent as it relies on a specific haptic jacket. Another related research is Simplified Authoring of 3D Haptic Content for the World Wide Web. The work provides 3D content authoring in a custom

Table 2.1: Comparison of research works table

Title	Objective	Type of System	Authoring ability	Web-based property	Device independent	Targeted Group
Simplified Authoring of 3D Haptic Content for the World Wide Web	Haptic integration in web	Browser plug-in	Yes	Yes	No	Web designers
Haptic Navigation in the World Wide Web	Haptic integration in web	Web browser	No	Yes	No	Visually impaired people
HapticWeb	Haptic integration in web	Script based toolkit	No	Yes	Yes	Web developers
Adding Haptic Feature to YouTube	Haptic content authoring in online video	Web based authoring tool	Yes	Yes	No	Web designers, viewers
HAMLAT	Haptic content authoring	Blender extension	Yes	No	Yes	Modelers
Authoring Framework for Haptic Broadcasting	Haptic content authoring for audio	Audio authoring and play system	Yes	No	No	audio content author and viewer
Haptic communication authoring Tool for an Instant Messenger	Haptic content authoring for IM	Online instant messenger	Yes	Yes	No	IM users
Threejs editor	Graphic authoring	web service	Yes	Yes	N/A	Graphic developers
Proposed system	Haptic content authoring in web	Web service	Yes	Yes	Yes	Web designers and viewers

Web browser, but the creation of such hapto-visual content relies on a VRML scripting language, which is not supported by current World Wide Web standards. HapticWeb toolkit integrates haptics into Web-Enabled games, however it requires users to possess knowledge of XVR programming and it relies on an outmoded plugin for graphic rendering. In comparison to other research, the proposed system is both device-independent and portable on the World Wide Web. This work provides authoring of 3D and haptic contents in a web-based system with a customer-friendly editor interface. The system is supported by HTML5, the newest cross-platform World Wide Web multimedia standard, and MPEG-V virtual reality representation format. Our work contributes to the topics of the integration of haptics in Web and haptic content authoring by designing a web-based authoring framework for hapto-visual web applications and implementing a modeling platform according to the framework, with most up-to-date techniques. The design of this proposed system is presented in the next chapter.

Chapter 3

Design of The Proposed System

This chapter gives details about the proposed system. It starts by an overview of the proposed methodology of web-based haptic application development, i.e., how this design pattern facilitates the development process. We then discuss the architecture of the proposed system. A detailed introduction of the system components is given. Next, we introduce the two types of hapto-visual scene designed, along with graphic and haptic properties of virtual objects in the scenes, and explains how they are described in MPEG-V format in the proposed system. After that, we explain the collision detection method and force simulation mechanisms used in order to keep the haptic and graphical rendering processes synchronized to perform realistic and stable haptic simulations. Finally, we concludes by presenting the steps demanded to build a haptic-enabled application by our system.

3.1 Methodology Overview

In comparison to other development methodologies of haptic-enabled web applications[52], most of which demand complete client-side programming logic based on specif-

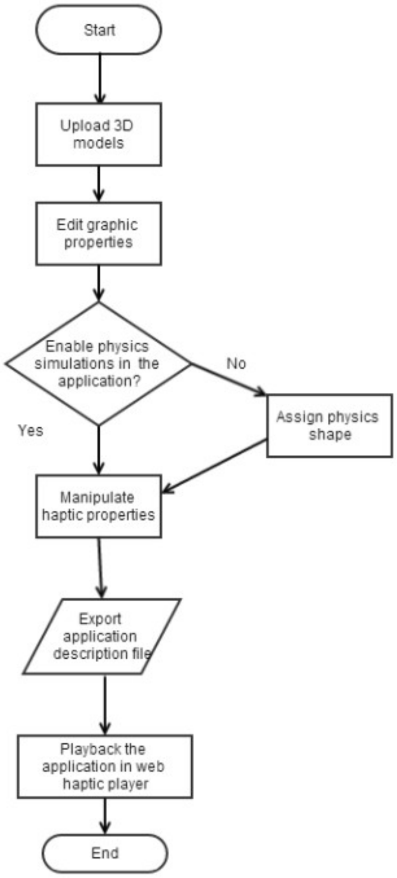

Start
Upload 3D models
Edit graphic properties
Enable physics simulations in the application?
No
Assign physics shape
Yes
Manipulate haptic properties
Export application description file
Playback the application in web haptic player
End

Figure 3.1: Flowchart of the proposed methodology for development

ic haptic or graphic APIs in browser to enable haptic forces and load graphic resource[28], whereas the proposed development methodology provides a simple and standard way for users to create their applications. In normal circumstances, such development process is operated by experienced researchers or programmers who are good at web 3D programming and at least have basic knowledge of haptic. Also, the simulations of haptic interactions between users and virtual worlds which are based on properties of virtual objects in the scene demand tremendous work of assigning graphic and haptic properties in the programming logic. Our goal of designing the methodology is to ease the development of haptic applications in web and make it possible that modelers without programming background in haptics and web graphic are able to create their own applications by easily import graphic resource and edit properties of the virtual world, and then allow users to experience rich haptic effects in 3D virtual environment on World Wide Web. The custom applications built by the proposed system, in which graphic and haptic interactions in the scene are generated automatically, are based on wide range of accessible graphic resource and the manipulation of them as virtual objects by user. This developing methodology is described in Figure 3.1. Firstly, users upload graphic resource to the web server. Then they could assign haptic and graphic properties to those objects. After completing the above-mentioned process, an application description file which contains the associated hapto-visual information of application can be exported into local machine. At last, to test and let others view the exported application, user will playback it in the World Wide Web.

3.2 System Architecture

According to the proposed methodology, we designed an authoring tool system that is web-based, device-independent, user friendly (professional programming skills not required). The system is composed of a web haptic authoring tool and a web haptic player. The authoring tool allows user to author, edit, and share haptic applications freely on web. Users are able to import 3D models into the scene and modify all of the graphic features which are shown in a 3D world on the screen. Then modification of haptic features from users are applied to each model as well as the whole scene. Thus, a complete haptic application is finished and can be exported as an MPEG-V file. The MPEG-V format describes the architecture of the virtual world by specifying the digital/physical contents for virtual worlds. It is a rich and well-designed xml standard for defining both graphical and haptic properties needed to describe a virtual environment. Additionally web-based haptic player is developed for reconstructing the web haptic application by importing the MPEG-V file. Consequently, a 3D haptic application with numerous kinesthetic simulations is displayed in the web browser. The overview and data flow of the proposed system is shown in Figure 3.2.

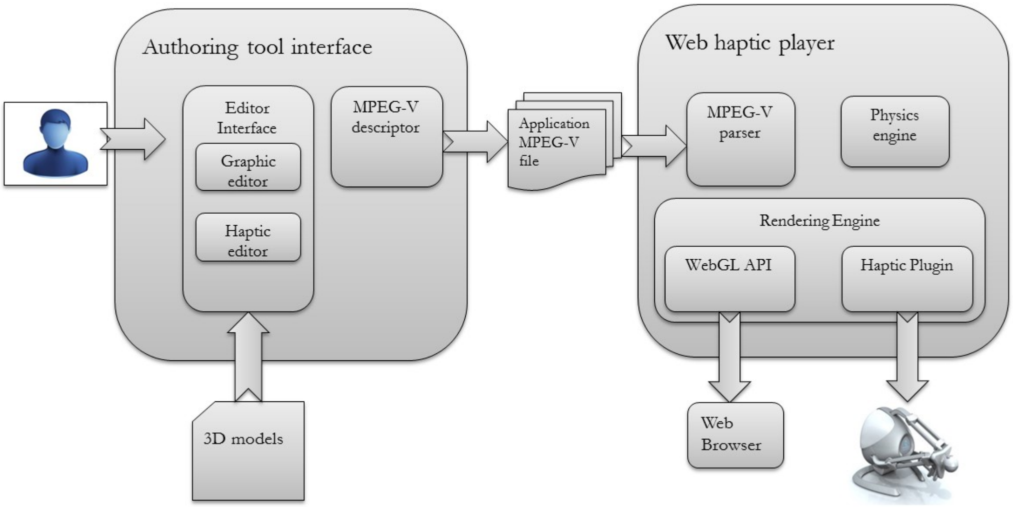


Figure 3.2: System overview and flow of data

3.2.1 Web Haptic Authoring Tool

The authoring tool consists of three internal components: an editing interface, an MPEG-V exporting engine and a rendering engine for graphic and haptic scene content. The editor interface is a graphical interface in the client-side let users design haptic applications by adding virtual objects, uploading 3D model files and assigning haptic properties to them. For the current version, the system supports 3D models represented in collada [3] format, which is a file format of static 3D models defined in an XML-based schema, easing the transportation of 3D assets between applications. The MPEG-V export engine is a web service that exports a MPEG-V file describing a haptic-enabled scene. It uses graphical and haptic attributes that users defined in the editor interface to generate the application description file. In the proposed system, we used MPEG-V xml schema [16] to describe haptic properties and part of the graphic properties. Besides, we added some schemas which are used by the WebGL library to describe other attributes of graphic scenes (e.g. virtual camera), which are not included in MPEG-V xml format. Figure 3.3 shows the interactions between data and the components of the authoring tool.

Inspired by other 3D modeling platforms [1], the design philosophy of the authoring tool is based on a data-visualize-edit development cycle. A flowchart illustrating the haptic-enabled 3D content authoring pipeline is shown in Figure 3.4. A 3D scene is displayed after graphic resource data is uploaded. Then the user views the scene and edits the data through the editing interface. The changes made directly modifies the data structures in the scene, and then the cycle repeats during the

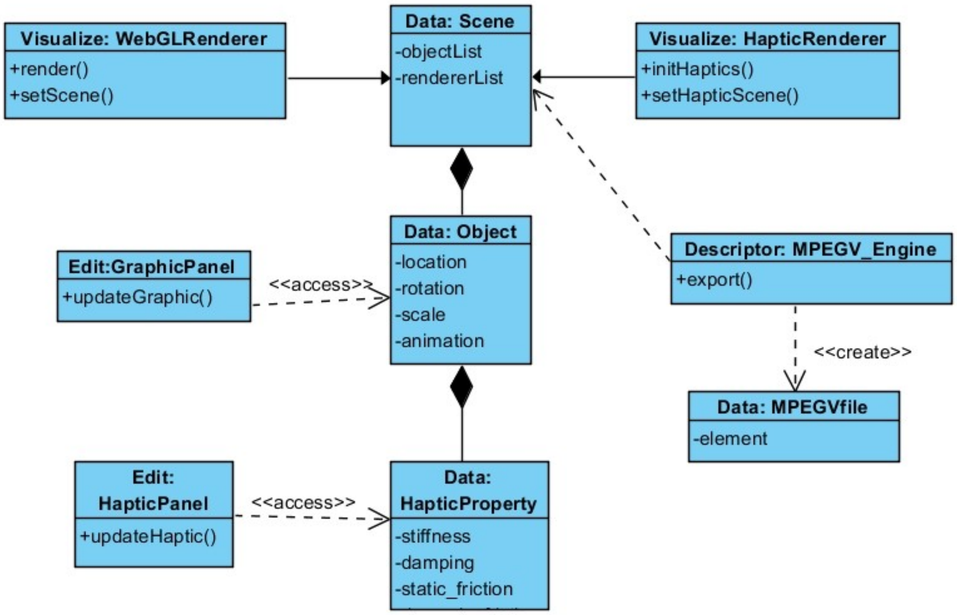


Figure 3.3: Class diagram of components in the authoring tool

modeling process. In the case of our authoring tool, this cycle works in the following way: a 3D virtual object may be defined by a 3D model file, represented by an array of vertices, and a material file representing the texture of object's surface. After the data is uploaded, User edits the location, scale and rotation parameters as well as haptic coefficients of it. The data is visualized by a WebGL renderer graphically which is capable of rendering the object in browser.

Based on the design philosophy, we designed the structure of data in the authoring tool. A scene is associated with a series of lists which represent same type of data in a linked-list. The data structure not only allows user to build complex hapto-visual scene using a simple structure, but also provides a way to reuse elements of data in each list. A model file or a texture file may be combined with different virtual objects which contains different graphic and haptic properties in the scene. The structure of the mentioned elements is shown in Figure 3.5. Three virtual objects, two of which sharing the same model file, are linked to two 3D model files and one texture file. As a significant improvement of the authoring tool comparing to 3D

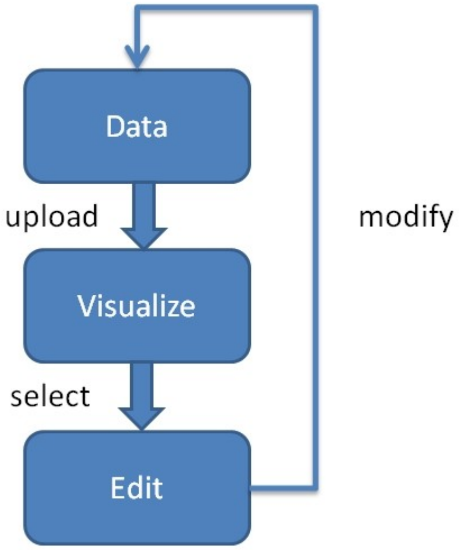


Figure 3.4: The data-visualize-edit development cycle of authoring tool

modeling platform, we augemented the data structures for representing object's material in graphical rendering by adding field for haptic properties. Those properties will be rendered in web haptic player by the built-in haptic rendering engine. Editing panels for modification of both graphic and haptic properties are provided in the GUI component. When user makes changes in the editing panel, those operations directly takes effects on the data structures used for haptic rendering. In this way we integrated haptic editing in the editing step of the modeling cycle for 3D modeling in the authoring tool.

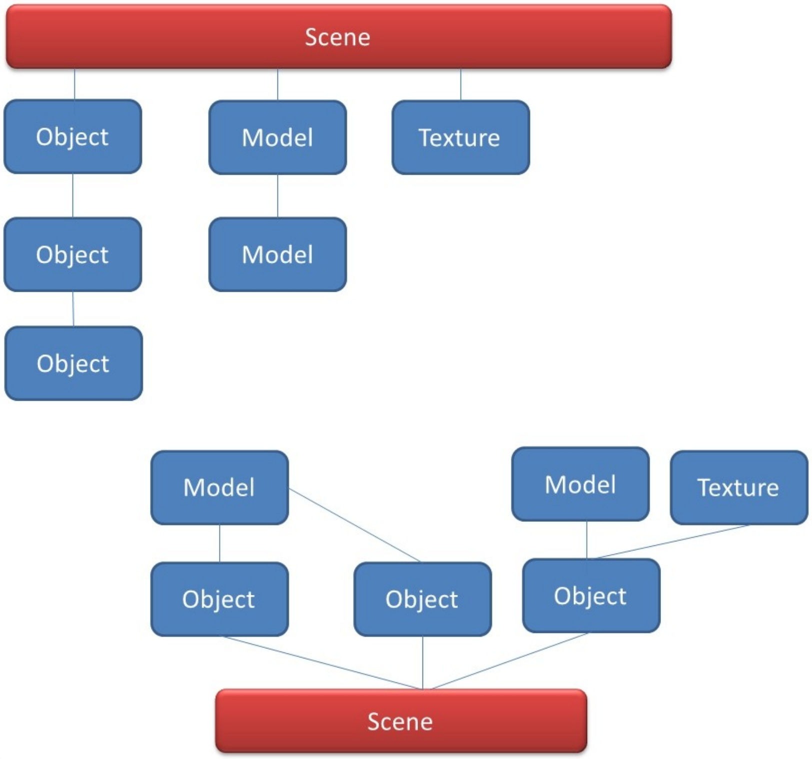


Figure 3.5: The organization of data elements in the scene of authoring tool

3.2.2 Web Haptic Player

The web haptic player is responsible for reconstructing the haptic virtual environment in the browser by simply importing an MPEG-V file generated by the authoring tool. Its internal MPEG-V renderer is able to map the information of virtual objects in the application file to the rendering engine. The rendering process of the 3D scene and haptic effects in browser are implemented by calling the WebGL API and haptic plug-in running in the background of the browser. We used the H5HPlugin[38] , which works as a javascript interface for the browser, to enable communication between the haptic device and the authoring tool. The mentioned plugin supports the following haptic devices: Sensable devices [6] Force Dimension devices Novint Falcon device [19] Moog FCS Robotics HapticMaster device [49] [38]. The web haptic player enables a virtual environment with rich haptic and visual interactions between virtual objects and the user. Once a well made application description file (already exported by the authoring tool) is loaded, the pre-defined virtual world and a visible proxy, representing the haptic device, appears in the 3D scene. The user then starts to explore the hapto-visual scene by moving the proxy into other virtual objects, watch and feel the interactions between virtual objects both visually and haptically. Four types of kinesthetic haptic effects are provided to users to feel objects with different shapes and material by touching them. Additionally, for the developers with javascript programming experience, we provide a coding area to allow them to create more complex applications. We provided part of the code for a graphic rendering loop and a haptic rendering loop in order to allow developers to add new functionality or new force effects into the application. The class diagram illustrating the interactions between the components in haptic player is shown in Figure 3.6.

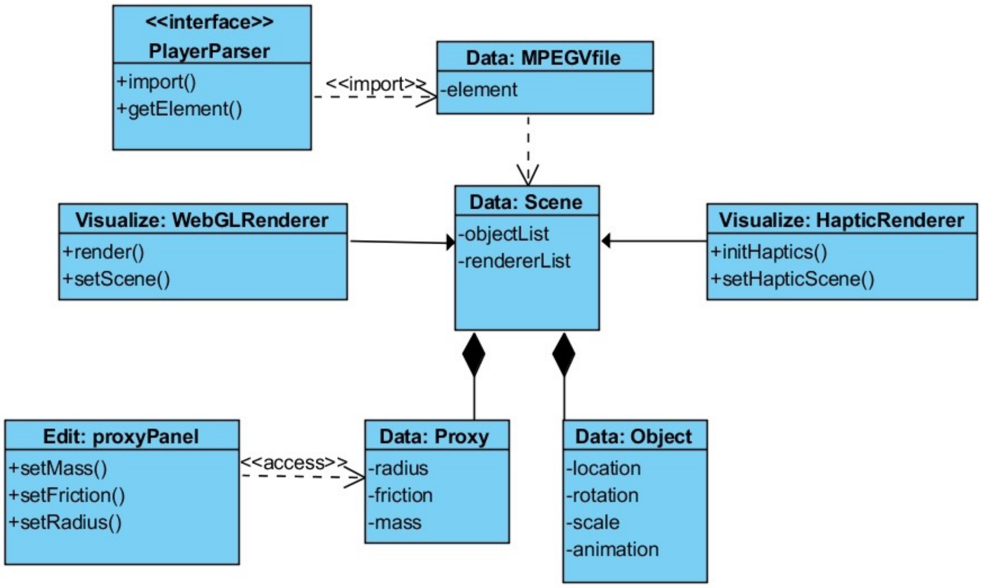


Figure 3.6: Class diagram of components in web haptic player

3.2.3 MPEG-V haptic descriptor and parser

As was mentioned in chapter one, to enable the standardization and reuse of the application, we have created components of MPEG-V haptic descriptor and parser in the system. The descriptor describes application by mapping the data value to associated data structure (e.g. haptic properties of the object) in XML schema. It is used for generating specific information into the MPEG-V schema elements when authoring tool exports the application description file. In the authoring tool system, the MPEG-V descriptor builds a tree structure to store the elements regarding to the information of haptic augmented virtual world. A root element is followed by two child elements containing different groups of data: ”VirtualWorld” and ”scene”. The ”VirtualWorld” element is the parent of a set of xml elements follow the MPEG-V standard to describe the virtual world, it is used in both graphic rendering and haptic rendering when it is parsed in web haptic player. The ”virtualWorld” contains a list of virtual objects with unique Id. Each ”Virtual

Object" element has three child elements of "Animation", "Control" and "HapticProperty" describing its specific information. The "scene" element is composed of elements which uses GLGE [7] xml standard to store additional information needed for graphic rendering. It is followed by three child elements to represent additional graphic information in the scene: "mesh", "material" and "camara". A detailed introduction to the structure of MPEG-V haptic description and specific xml schema will be given in section 3.6.

The parser is an internal component in web haptic player that imports input data

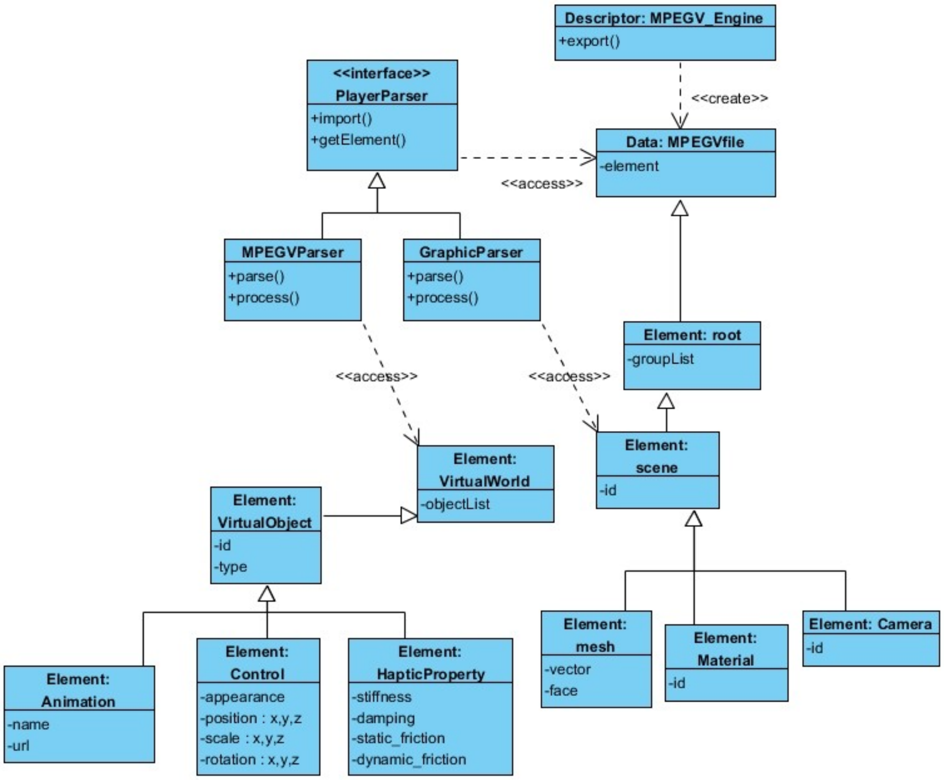


Figure 3.7: Class diagram describing the interactions between descriptor and parser

(MPEG-V xml file generated by authoring tool) and creates a parse tree which stores a structural representation of the virtual scene in the application. The interactions between descriptor and parser are shown in Figure 3.7. As we can see in the diagram, the MPEG-V description file is exported by the descriptor and is imported by the parser in web haptic player. Then the two sub-components, MPEG-V parser and graphic parser, parse the nodes under the root element of MPEG-V file respectively. Once the data is parsed, the parser can start processing the elements and send the needed data to the rendering engine. After importing the MPEG-V file, the working process of parser is preceded by creating tokens from the input file. In this stage, data is split into tokens based on xml schema in the system. Then a syntactic analysis is operated to the tokens by checking if the tokens form an allowable expression [10]. The parser checks the validity of attributes to ensure the data works properly in the rendering phase. Finally, a MPEG-V parse tree is constructed based on the description file exported by authoring tool and will be sent to the internal renderer in web haptic player. The flow diagram of parsing process is illustrated in 3.8.

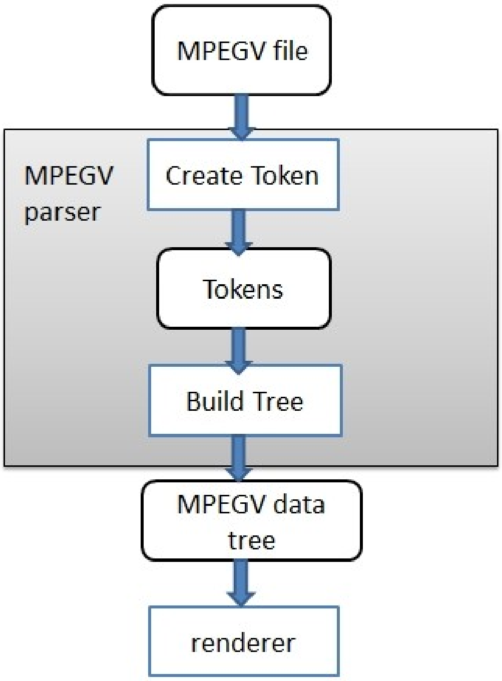


Figure 3.8: flow diagram of parsing process

3.2.4 Interface Design

The system comes with two main client interfaces in web browser for haptic authoring and viewing: an interface of the web haptic authoring tool and interface of the web haptic player. The interface of authoring tool allows user to build a haptic application step by step. Figure3.9 shows a snap shot of the authoring tool's GUI (Graphical User Interface): It includes several panels that are being used when user is designing the application and are sorted graphically according to the order of authoring steps in the system. Above each panel there is a title guiding the user to perform the right action for the purpose of customizing the application.

Firstly, the panel on the top-left area contains two button areas and a drop area for user to upload 3D models exported from other graphic modeling software. The first button area allows user to select the texture file of the 3D object, which user would like to manipulate later, in the local machine and uploads it to the server.

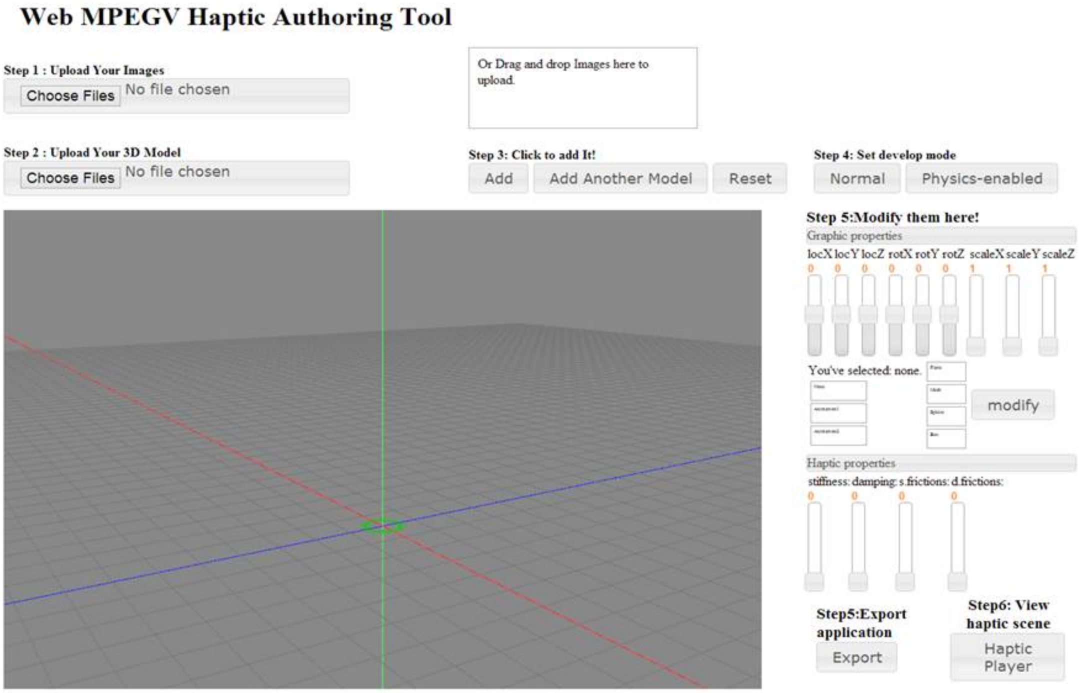
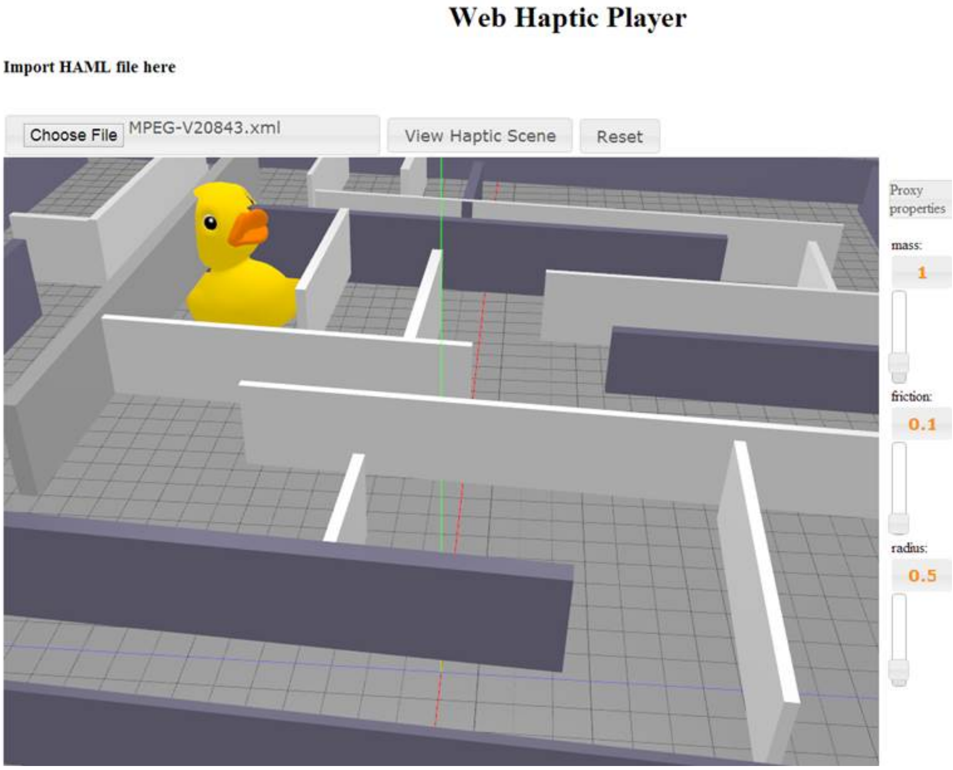


Figure 3.9: GUI of the web haptic authoring tool

Once the texture file is loaded, then the second button area is used to select and upload the model source in supported 3D format. The panel in the middle of the top area contains three buttons: "Add" button is used when user finishes uploading the texture and model files of a 3D object; the other two buttons are used to add another model and to reset the scene. On the top-right there is a panel for user to choose the development mode of the application: Basic and physics-enabled. The selected mode is not changeable before one application is exported. The main panel containing a graphic editor and a haptic authoring editor is located on the right side of the screen. It allows a modeler to modify the graphical properties and haptic properties of the selected object. In the graphic editor, we provided 9 sliders for the modification of graphic properties including locX(Location of the object on the X axis), locY(Location of the object on the Y axis), locZ(Location of the object on the Z axis), rotX(Rotation of the object on the X axis), rotY(Rotation of the object on the Y axis), rotZ(Rotation of the object on the Z axis), scaleX(scale of the object on X axis), scaleY(scale of the object on Y axis), scaleZ(scale of the object on Z axis). Modeler is able to modify those properties using sliders or by entering value into a text box on the top of slider when the desired value is out of the slider's range. Besides those graphic sliders, there is a spinner for user to assign animation to the 3D object. If a user selects the physics-enabled mode for the application, then a physics editor appears on the right of graphic editor. It contains a spinner providing choice of four types of physics shape for 3D objects. The selected hidden shape encloses the object in the scene when application is being rendered. Haptic editor provides sliders representing four haptic coefficients: damping (Kg/s), stiffness (N/mm), static friction and dynamic friction. By setting the values of those properties, user determines the feeling of the object when application is rendered later in the web haptic player. The panel in the bottom is composed of two buttons: the "export" button is used for exporting the application that user designed. After clicking that button, a generated link redirecting to the description file of the application will appear next to the button and user will be able to save the file in the local machine. The "haptic player" button is for redirecting user to the page of web haptic player to test and view the exported application. The authoring tool has a canvas (an HTML5 element which is used to draw graphics [11]) area containing a 3D scene with a three-dimensional coordinating system on the left side of screen to let users have a preview of the application while editing the objects in the scene. The scene is updated when user adds or modifies any object graphically.

Figure 3.10 shows the interface of web haptic player. The panel on the top has


Web Haptic Player

Import HAML file here

MPEG-V20843.xml

Proxy properties

mass: 1

friction: 0.1

radius: 0.5

Figure 3.10: GUI of the web haptic player

the functionality to select and import description file exported by the authoring tool, view and reset the scene. Besides, it provides a 3D canvas area for users to explore a complete virtual environment graphically and haptically. In the testing phase, user controls a proxy with a basic shape(sphere or box) by haptic device and get the feeling of objects inside the scene by touching it with the proxy. Along with the area to display the scene, the interface contains three sliders to modify the haptic properties of the proxy in order to let users feel the different level of forces when the haptic property of the object is changed. The properties attached to the sliders are mass, friction and stiffness. We will discuss those properties section 3.5.1.

3.3 Design of Hapto-Visual Scene

In our authoring tool, we provide two modes of virtual environments: basic mode and physics-enabled mode. The basic mode provides a 3D environment with virtual objects that can have haptic properties. Hence, the designer can create a solid hapto-visual scene. It does not simulate visual interactions between virtual objects. This mode is suitable for applications which users will use to probe static objects in the environment, such as virtual museums. Furthermore, the objects in this environment are static, and thus, they are not movable. The physics-enabled mode applies physics laws such as gravity, velocity, and spring force into the scene. All these properties can be pre-configured by the developer. This mode is suitable for development used for educational and gaming purposes, where multiple objects may collide in the environment. The two aforementioned scenes could both be designed and exported in the authoring tool and imported by the web haptic player.

The rendering mechanism of haptic enabled scenes is illustrated in figure 3.11. As shown in the figure, the hapto-visual scene is displayed in the web haptic player while the graphic rendering engine and haptic rendering engine are working on the background. When collisions between virtual object and the proxy controlled by the user happens, the graphic engine sends the collision event to the haptic rendering engine. Then the haptic rendering engine will send the calculated force

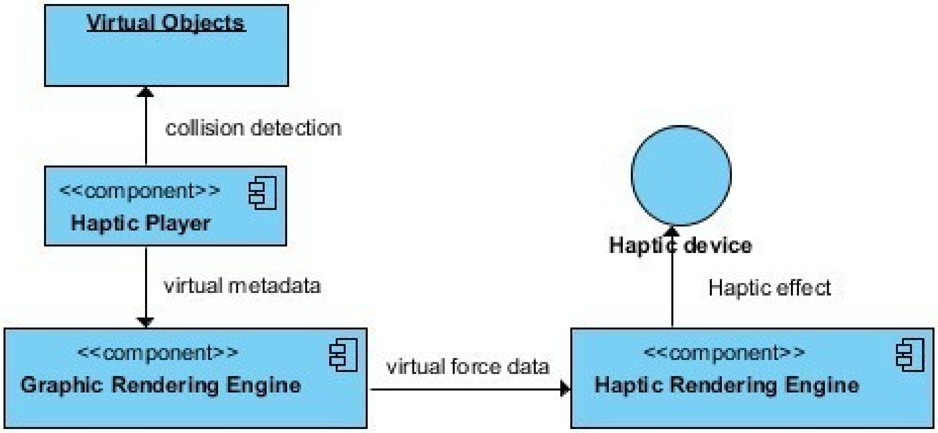


Figure 3.11: Flow of data for haptic interactions in the hapto-visual scene

data (based on force simulation mechanism in the system) to the haptic device through browser plugin.

We will talk about the details of collision detection algorithm and formulas of forces applied to virtual objects accordingly later.

3.3.1 Basic Scene

The basic scene is defined as the type of hapto-visual scene which contains a set of static virtual objects comprising of a set of haptic parameters. This type of scene are mainly used for applications for education, 3D modeling and surgical purposes. Figure 3.12 shows a simulated human leg model in the web haptic player which user from medical background can touch and feel by haptic device and then learn the architecture of human's leg from the sense of touch. In the basic scene, user is able to explore a 3D haptic scene and probe virtual objects inside it through an proxy controlled by haptic device. The proxy of user is a basic sphere whose radius is 0.5 in our 3D coordinating system. User can control the proxy's moving in 4 directions(up, down, left, right) by moving the touch controller on the haptic device. When the proxy moves into a virtual object in the scene, force feedback

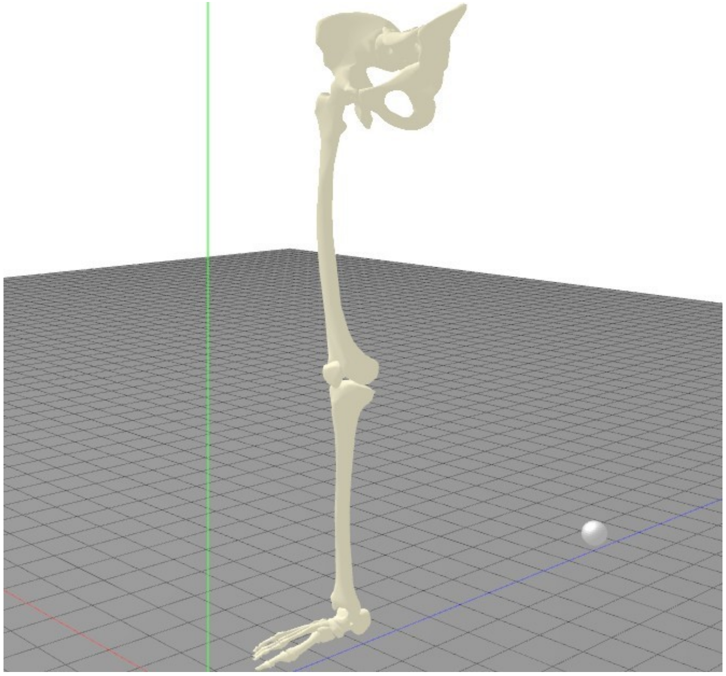


Figure 3.12: A human leg model for surgical simulations in the web haptic player

is applied to the haptic device and user feels it on the controller. According to the force feedback, user could feel the shape of virtual object by moving the proxy around on its surface.

3.3.2 Physics-enabled Scene

The latest improvements of graphic hardware and rendering software allow designers and developers of haptic application to simulate the real world with graphic and haptic effects used to create virtual worlds governed by real physics. To allow users to enable such physical simulations both graphically and haptically, we have defined physics-enabled scene with real-time dynamic graphic simulation and kinetic feedback simulation in the authoring tool for application of gaming or physical education purpose. Figure 3.13 shows a physics enabled scene created by the

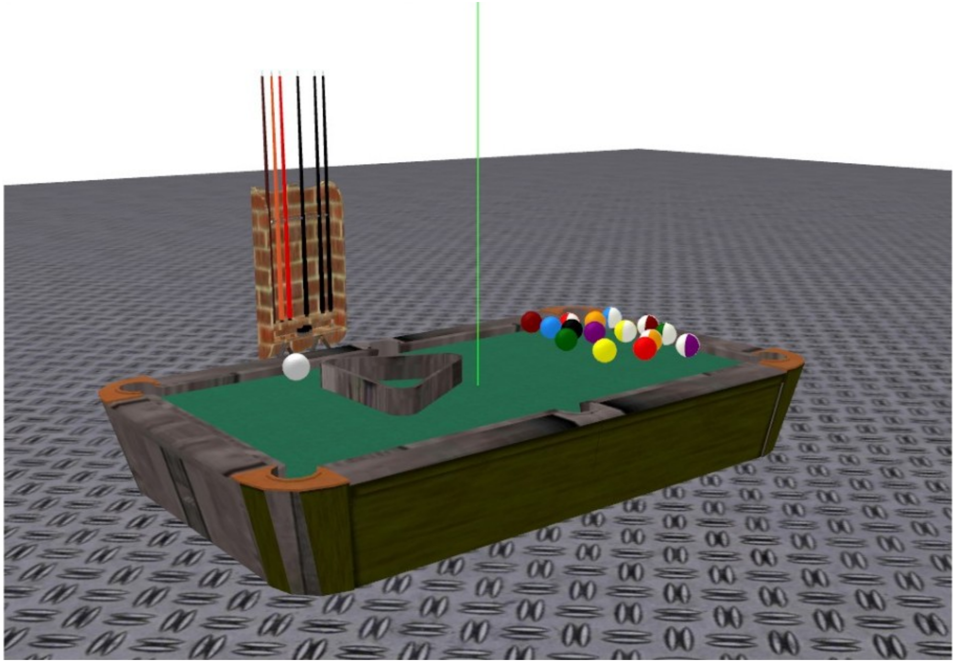


Figure 3.13: A pool game in web haptic player

authoring tool. It is rendered by the web haptic player as a simple pool game. A wide range of real-time physical simulations in virtual world are enabled to provide rich real-world experience for users. The physical simulations provided in the physics-enabled scene are as follows:

- Gravity: Gravity phenomenon applies in the real world consistently. In the proposed system, the gravity effect takes place towards negative Y-axis. It provides simulation of weight to virtual objects with mass and causes virtual objects to fall to the ground. Meanwhile, user would feel a simulated gravity force on the haptic device.
- Velocity: Virtual objects is able to move in the scene with an initial velocity which would be changed when resident forces (such as dynamic friction) acts on the virtual object in the direction of the relative velocity.
- Impulse based collision reaction simulation: In the physics-enabled scene, all

collisions are processed using impulse rule. When collision happens between two objects, the velocity of them are updated when impulse, defined as the product of the average force multiplied by the time it is exerted, applies on them. When such a collision happens between proxy and a virtual object, a specific force appears on haptic device according to the amount and direction of the impulse.

3.3.3 Properties of Virtual Objects

Objects, having real existence and value independent of the subject who observes it, are the key elements of constructing both real world and virtual world. Users are able to create an interactive haptic enabled 3D virtual world which not only presents perception stimulus but also gives them independence to manipulate elements inside it. The proposed system allows users to create virtual objects with wide range of properties in the authoring tool. Those properties are divided into four parts: graphic, haptic, animation and physics.

Graphical Properties

A basic 3D modeling functionality is provided in the proposed system. Each virtual object in the scene consists 3 kinds of three-dimension properties: location, rotation and scale.

- There are three location parameters of a virtual object: locX, locY and locZ. Each of them represents the position of virtual object on the x, y, z axis of three dimensional coordinate system.
- Similar as location parameters, rotation parameters contain three principal rotations: loc, which is a rigid body movement that keeps a point fixed, around

the x, y and z axes. In the proposed system the scope of each parameter is from -3.14 to 3.14, we.e. -360 degree to 360 degree in solid geometry.

- The scale parameters: rotX, rotY, rotZ of a virtual object represents the scale ratio of each linear dimension of the object to the same feature of the original.
- 3D model: 3D model is the file that represents a 3D object using a collection of points in 3D space. It is usually exported by modern 3D modeling software, such as Blender [1]. In the proposed system, we support the interactive 3D file format called collada, which end with a .dae filename extension and are widely used in modern 3D applications.
- texture source: in 3D modeling, the presentation of a model with complex material (e.g. colorful objects), is mapped to a texture file. The texture source property is required when user are importing a 3D model mapped to a specific texture.

Haptic Properties

We used four haptic parameters to let users design a virtual object and feel the kinesthetic perceptions accordingly: stiffness, damping, static friction and dynamic friction. Those properties, representing the material, hardness of object, provide a stable haptic simulation of solid objects in virtual world. In our system, the haptic properties are rendered both haptically and visually:

- Stiffness coefficient represents a minimum threshold required to emulate a virtual object in the scene.
- Damping coefficient of a virtual object defines the decay of an oscillation after a disturbance such as force applied on it.
- Static friction coefficient is the ratio of the static friction force between the two object and the normal force pressing them together. It is used to simulate

the materiel of a virtual object in the scene.

- Dynamic friction coefficient is the ratio of the dynamic friction force, unlike the static friction ratio, between the two object and the normal force pressing them together. The range of each haptic parameter in the proposed system is from zero to one.

Animation Properties

In order to enrich the experience of virtual world, we allow users to assign animation to the virtual object. we provided two pre-defined animations using change of linear positions in specific frame in the authoring tool: disappear and blow up. They provide visual effects of disappearing and blow-up of the model in the scene.

Physics Properties

As we have mentioned before, a physics enabled mode is provided for users to develop a real world like virtual environment in which physics laws in real life include gravity. Here the physics properties we defined for the virtual objects in the system are discussed. First of all, we defined four types of physical properties, they works as a proxy to make it possible to enable rich communications between virtual objects based on collision-detection mechanism in graphic rendering engine. The aforementioned physic types are: sphere, plane, mesh and box:

- Sphere type is to render the virtual object as a sphere with a specified radius, it is mainly used for virtual objects like balls and other simple objects.
- Mesh type is to render the virtual object as a mesh which consists of a set of points , it is mainly used for the objects which do not move or perform animations.

- Plane type is to add a plane into the scene by defining the normal and distance of it.
- Box type is to envelop the virtual object, especially for whose shape is nearly a cube, in a box with specified height, width, depth.

For every physics type, there are a set of parameters mentioned above to define the object along with a mass parameter to indicate the weight of it.
Here we give an example of the properties of a virtual object, where the property of this object is defined as follows:

- Model Source- 3D model: physicsBox.dae, texture source: box.jpg
- Graphic- locX: 3.0, locY: 3.0, locZ: 4.0, rotX: 0.83, rotY: -1.4 rotZ: 0.83, scaleX: 1.80, scaleY: 0.80, scaleZ: 1.30
- Animation- none
- Haptic- stiffness: 0.12, damping: 0.28, static frictions: 0.32, dynamic frictions: 0.26
- Physics- type: box, mass: 4.00, height: 1.60, depth: 3.60, width:2.60

3.4 Collision Detection Mechanisms

To allow users get the feeling of an virtual object while touching it in the virtual world, the first problem is to detect intersection between the proxy and virtual objects in the scene in graphic rendering process timely and correctly. Then force feedback applies accordingly when the event of the collision is sent to the haptic rendering engine. For this purpose, a real-time detection mechanism for the collisions between proxy and every virtual object has to be defined accurately.
In three-dimensional coordinating system, a virtual object is a specific mathematical representation of three-dimensional surfaces and points such as polygon meshes

which use a collection of vertices, edges and faces to defines the shape of a object. The raw algorithm to detect collisions between such complex objects is to detect the intersections of all faces of each pair of objects recursively. However this approach requires huge amounts of real-time calculations, leading to increased workloads in graphic rendering when adding more objects into the scene, that finally causes application running slow.

To solve this problem, we design different collision mechanisms for the two types of hapto-visual scene based their characteristics and requirements for graphic rendering.

3.4.1 Collision Detection Method in Basic scene

The collision detection algorithm in basic scene is based on the minimum bounding box method. The term minimum bounding box is a concept in geometry which means the smallest measure in N dimensions that covers a given point set. In computer graphics, a minimum bounding box for an object is the smallest closed volume(in this research, a three-dimensional box) that completely covers the object in order to enable simpler ways to test for overlap with other objects. Figure 3.14 from WIKIPEDIA [13] shows the minimum bounding box of a 3D object in the scene. When there is no intersection between two bounding boxes, the contained object cannot collide.

To ease the description of minimum bounding box of each object and the phase for checking intersections, we use axis-aligned minimum bounding box which defines all edges of the box locate parallel to the three dimensional coordinate axes. The axis-aligned minimum bounding box can be represented by a set of vertexes of the box which are determined by minimum and maximum value on three dimensions of the points in corresponding object. For example, in Figure 3.15, the object A whose vertex's minimum and maximum values on three

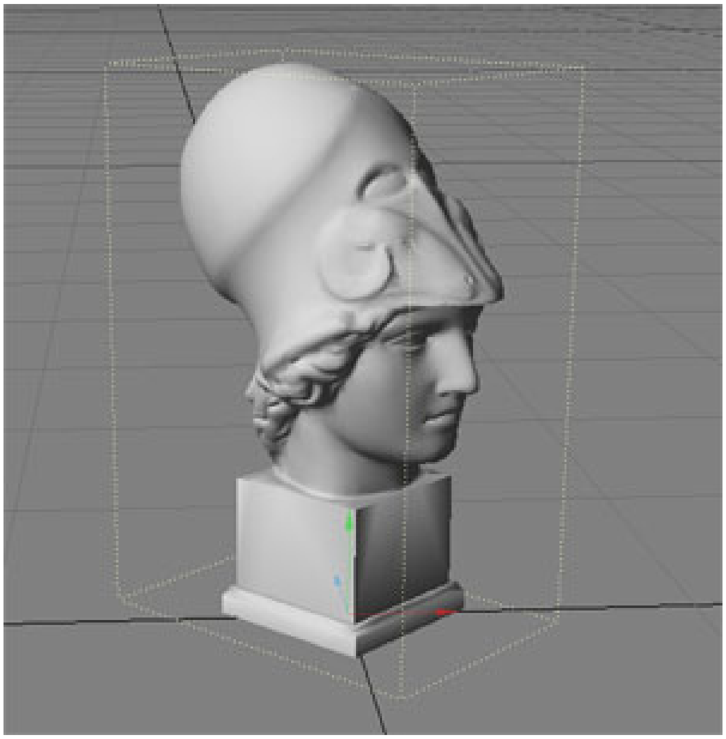


Figure 3.14: Minimum bounding box of a 3D object (wikipedia)

dimensions are $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$, is enclosed by its axis-aligned bounding box whose maximum and maximum points on three axises are $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$, and the center of the axis-aligned bounding box is $(x_{min} + x_{max}/2, y_{min} + y_{max}/2, z_{min} + z_{max}/2)$.

We choose the box as the bounding volume over other shapes such as bounding sphere for objects in the scene in the proposed algorithm due to the advantage of bounding box in simulating friction force which applies when the proxy is moving on the surface of the ground instead of having intersection with it. The advantage is obvious because bounding sphere method requires tracking and calculating the distance between the center of proxy and the ground all the time to ensure the object is moving on the ground and there is no intersection between them. However the bounding box method does not require any complex test because there is no relative motion between them on the normal of the ground.

When the basic scene is being rendered, the system firstly create an axis-aligned minimum bounding box for each virtual object by storing the minimal and maximal value in three dimensions for the object. Then for each bounding box, we check the intersections between it and the proxy of haptic device by solving collisions for each axis and finding the earliest possible collision in time that satisfies all axises. The basic law of checking intersection between two axis-alined bounding box is simple and fast. As shown in Figure 3.16. The two boxes are colliding only when all the projections of them on three axises are having intersections. Thus we just need to check if the first box's maximal value on each axis is greater than the second box's minimal value and that the first ones minimal value is less than the second ones maximal value. The earlist collision time is determined by the maximum earlist collision time of two objects on three axises which is $T_0 = MAX(T_{0x}, T_{0y}, T_{0z})$.

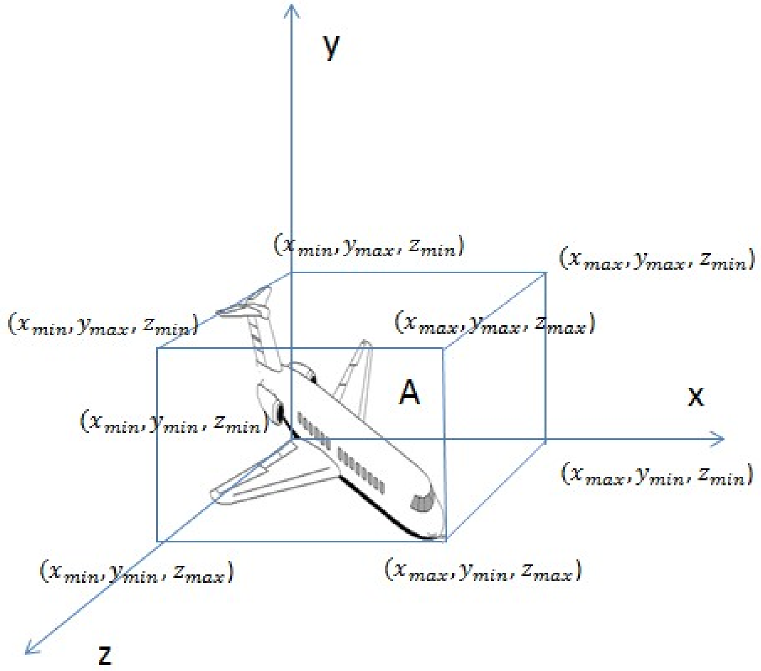

y
$(x_{min}, y_{max}, z_{min})$
$(x_{max}, y_{max}, z_{min})$
$(x_{min}, y_{max}, z_{min})$
$(x_{max}, y_{max}, z_{max})$
A
x
$(x_{min}, y_{min}, z_{min})$
$(x_{max}, y_{min}, z_{min})$
$(x_{min}, y_{min}, z_{max})$
$(x_{max}, y_{min}, z_{max})$
z

Figure 3.15: Axis-aligned bounding box for an object

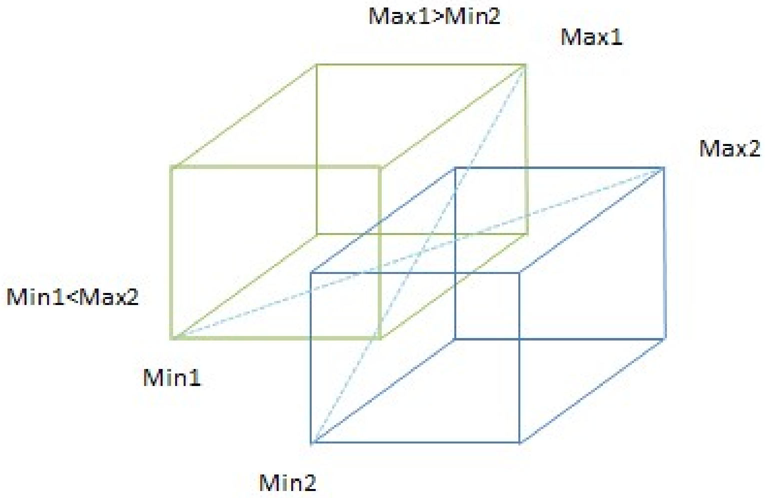


Figure 3.16: Collision condition for two axis-aligned bounding box

Similarly, the latest collision time is determined by the minimum latest collision time on three axises which is $T_1 = MAX(T_{1x}, T_{1y}, T_{1z})$. Now we have to check if the earliest collision time occurs earlier or at the same time as the latest collision time, and it takes place within 1 time unit: $T_0 < T_1 < 1 = true$. Finally, we have determined the collision period between the proxy and the object B in basic scene. After that, haptic force is applied in this period of time according to the intersection type and depth. We will talk about different types of forces happen in the basic scene in next section.

3.4.2 Collision Detection and Resolution in Physcs-enabled Scene

There are two types of collision detection conditions in a physics-enabled scene: collision detection between different virtual objects, collision detection between the proxy and a virtual object.

Physics simulations take place based on the collision detection process between different pairs of virtual objects. While the physics-enabled scene is rendered by the browser, for each virtual object in the scene we detect collisions by predicting the position of it in the next frame. In that case, the object is moved to a temporary position and the rendering engine checks the interference. Four types of physics shapes (sphere, box, mesh, plane) are defined to ease the collision detection phase. In the implementation phase of collision detection and visual physic effect simulations, we use the JigLib physics-engine [12] to render the 3D scene and enables physics effect visually. The collision event is handled inside the physics engine. If the collision happens between a pair of virtual objects, the velocity of objects is updated based on impulse between them. Then the average force in this collision procedure is calculated according to the impulse rule in physics.

Collision between the proxy of haptic device and virtual object happens when the sphere of proxy intersects with the bounding shape of a virtual object. We store a list containing all of the virtual objets and their corresponding shapes, and check for intersections between the proxy and objects in every frame. When such interaction happends, different types of collision resolutions take place according to the shape of associated object:

- Collision resolution between proxy and sphere: When the proxy collides with a sphere, new velocities are assigned to them according to impulse law. Then the haptic device simulates the impulse force on the haptic controller.
- Collision resolution between proxy and box: Similar as sphere, when proxy moves towards box with a high velocity , impluse force applies to the haptic device. However, when proxy moves on the surface of box, dynamic friction force appears in the contrary direction of proxy's velocity.
- Collision resolution between proxy and plane: When the proxy collides with plane, frictional forces applies when the proxy is moving or tends to move. In

most cases, we assume the plane is parallel with the ground.

We will discuss the related force simulations in haptic rendering for the above-mentioned collision resolution methods in the next section.

3.5 Force Simulations

For haptic applications, it is extremely important to ensure the force simulations which a user feels is as natural as real worlds because haptic technologies has a codependence between sensing and manipulation that is at the heart of understanding how humans can so deftly interact with the physical world [50]. Thus, the diversity and accuracy of force feedback simulations directly determines the quality of the haptic application.

Different types of force simulations are designed in our system according to how human feel when she or he touches an object in real world. Those forces include frictional force, spring force, damping force and impulse force. To implement those forces in haptic applications, the haptic players is responsible to keep track of graphic information including proxy velocity, beneath distance between proxy and other virtual objects, which is needed in the calculation procedure of forces. The forces are based on graphical information of proxy and virtual objects in the scene as well as haptic properties of 3D models such as stiffness, friction which user has defined by the authoring tool. Haptic player generates force feedback event in graphic haptic rendering loops and send them to the haptic plugin to simulate the force on the haptic device. In the end, user feels the force which is based on graphical interactions in the haptic application.

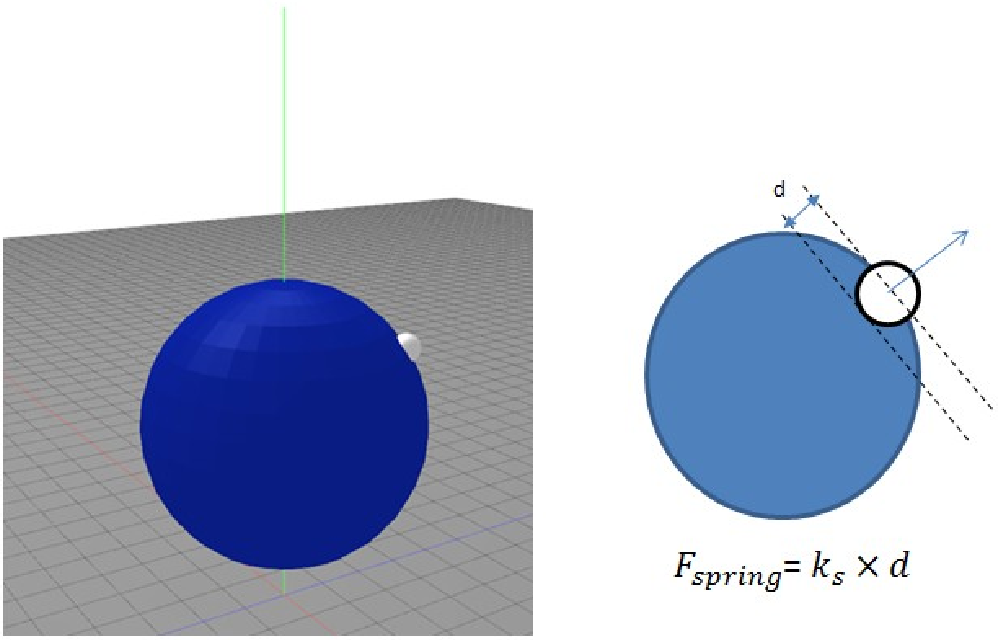


Figure 3.17: Spring force in basic scene

3.5.1 Force Simulations in the Basic Scene

In the basic scene, The haptic force feedback for the interaction between the proxy and the object is based on three forces: spring force, damping force and frictional force. The spring force is the force exerted by a compressed or stretched spring upon any object that is attached to it. An object that compresses or stretches a spring is always acted upon by a force that restores the object to its rest or equilibrium position. In the proposed system, we use spring force to let users feel the hardness of an object. Spring force is determined by the following Equation: $\mathrm{F}_{spring} = k_s \times d$ where k_s is the stiffness coefficient and d is the depth of the haptic proxy beneath the surface of the object. It is applied when the proxy controlled by the user is hitting or moving into a virtual object. The procedure is illustrated in Figure 3.17 where the proxy is moving into a virtual ball. Damping force is an internal restoring force resists the motion of on object in a harmonic oscillator system. It

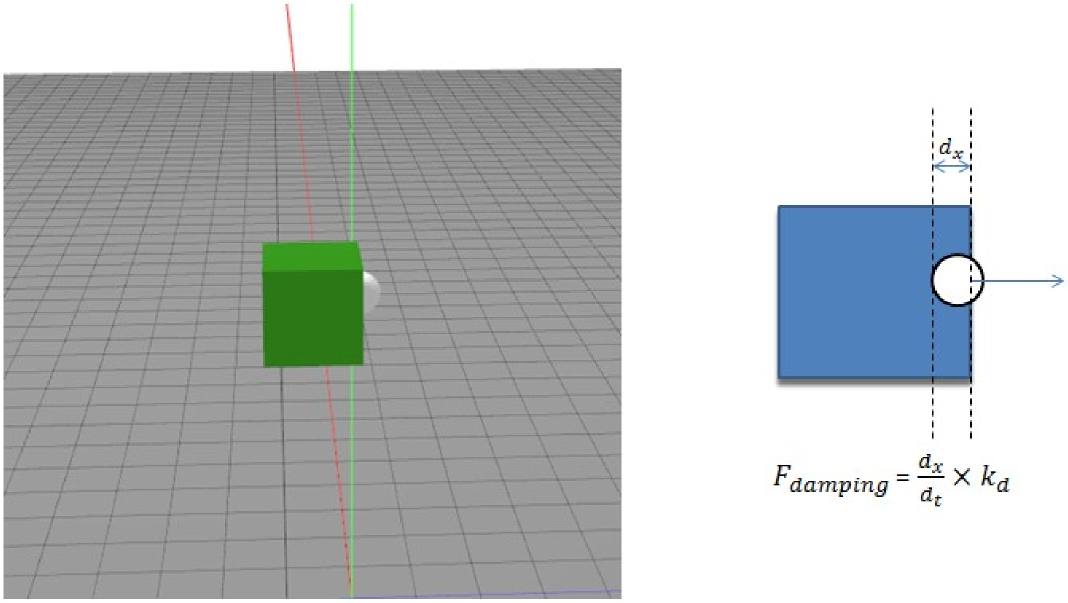


Figure 3.18: Damping force in basic scene

is linearly dependent upon the velocity and the damping coefficient of the object. In the proposed system, damping force applies when the proxy is moving into the object. It is rendered using following Equation: $\mathrm{F}_{damping} = k_d \times v = k_d \times \frac{d_x}{d_t}$ In the above Equation, k_d is the damping coefficient of the object which user assigned to it, $\frac{d_x}{d_t}$ is the first derivative of the penetration depth of the proxy inside the object. Figure 3.18 shows the force simulation mechanism when the proxy is moving into a 3D cube. Frictional force has two types: static friction and dynamic friction. They both resist the relative motion of solid surfaces as shown in Figure 3.19. The static friction is applied when the proxy is moving towards a virtual object on the contacting surface. It is the minimum force for the movement of a virtual object. The amplitude of the force is determined by the following Equation: 3.5.1 $\mathrm{f} = \mathrm{F}_N \times \mu_s$ where μ_s is a predefined static friction coefficient and is the normal force. In the physics system we used the product of the mass of the virtual object and the gravity coefficient to replace when applying the simple static friction force.

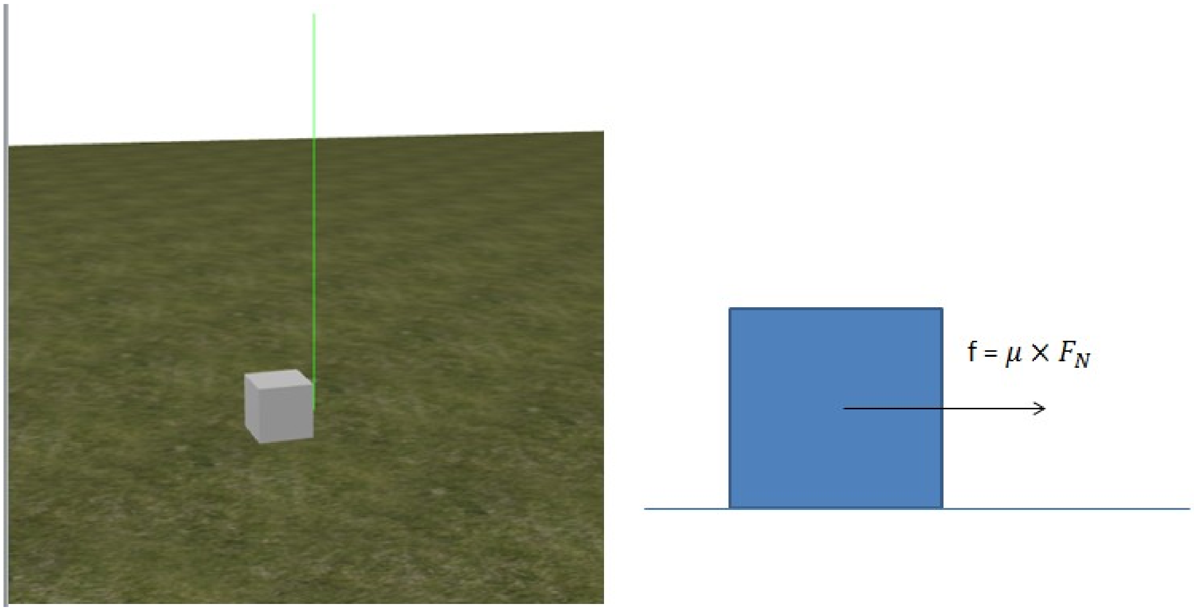


Figure 3.19: Friction force when proxy moves on a lawn

Dynamic force is applied when the proxy moves on the surface of a virtual object or mesh. It is determined by which is the user assigned dynamic friction coefficient to the virtual object as we can see in Equation 3.5.1 $f = F_N \times \mu_d$

3.5.2 Force Simulations in the Physics Scene

In the web haptic player, the four kinds of virtual objects enrich the virtual environment in web by performing physics law according to their shape. Physics effects happen not only with visual effects but also with haptic effects. To make that possible, we use an open source physics platform in WebGL in graphic rendering part. Haptic effects, which are based on the shape of virtual object and the specific type of force applied, take place when collision happens in graphic scene.

In a physics-enabled scene, the feeling of virtual objects is determined by haptic properties and imaginary physics shape that the modeler has assigned to the object. Two types of forces are applied to the haptic device when the proxy intersect with virtual objects in the scene: frictional force applies in the same way as it does in the basic scene, static friction applies in the contacting surface between proxy and a virtual object to resist relative motion of the proxy when both of the them are not moving. The static friction force is determined by by Equation 3.5.1.

Kinetic friction applies when either the proxy or the virtual object that is in contact with the proxy is moving. Similar as basic scene, the kinetic friction in a physics-enabled scene is determined by Equation 3.5.1

Force caused by impulse applies when momentum of the proxy is changed in collision. Momentum is the tendency of an object to continue to move in its direction of travel. Momentum equals the product of the mass and the velocity of an object: $p = mv$ Where m and v are the mass and the velocity, respectively. Force is formed when impulse takes place. The impulse of force is a vector quantity defined as the product of the average force multiplied by the time it is exerted [9].Therefore, impulse is equal to the change in momentum of an object. And force can be defined as: F=m×$a = m \times \Delta v/\Delta t$ Figure 3.20 shows the scene of a pool game developed by authoring tool. The force caused by impulse is formed at t_1 when proxy with velocity U_1 hits a ball with velocity U_2. After the collision, the proxy is moving with a constant velocity $V_1 mathwhiletheballismovingwithvelocity V_2$ to the right. According to the impulse formula, the force on the proxy at the moment of collision is $m \times \Delta v/\Delta t = m_1 \times \frac{U_1 - U_2}{V_1 - V_2}$. In physics-enabled application, when collision happens between proxy and virtual objects, forces formed by impulse apply firstly. The haptic player loads the velocity of proxy every frame and calculates the change of it before and after the collision, then generates force according to Equation 3.5.2 , where m is the mass of proxy user assigned to the proxy in the web page and Δt is change of time. Frictional force apply when the proxy move over the ground or the surface of virtual objects, it is determined by mass and frictional parameter which user assigned to the proxy as well.

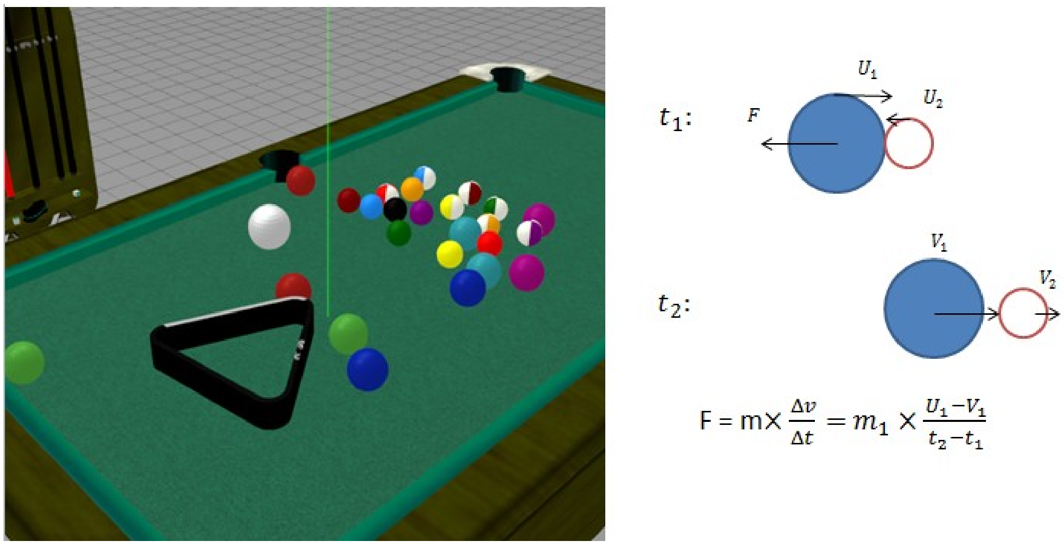

U_1
U_2
t_1:
F
V_1
V_2
t_2:
$F = m\times\frac{\Delta v}{\Delta t} = m_1 \times \frac{U_1-V_1}{t_2-t_1}$

Figure 3.20: Impulse force in pool game

3.6 File Format of the Description file

3.6.1 Haptic MPEG-V Description of Virtual Objects

MPEG-V is a multimedia standard in xml format for information exchange with Virtual Worlds.It provides the architecture of the virtual world with multi-sensorial content associated with audio/video data and proxy characteristics. The presentation allows user to control multimedia presentations and applications by using advanced interaction devices. For the authoring of web haptic applications, we specified a haptic MPEG-V format which defines a standardized xml format to describe a hapto-visual application in web. In order to do it, we extracted the elements of “Virtual World Object Characteristics” which specifies semantics and syntax of description schemes of the virtual objects in MPEG-V to describe a virtual world with a list of virtual objects and proxys. We provided an MPEG-V descriptor to characterize a list of metadata of a virtual world object, and enabling the storage and migration of a virtual world object between different world. When the description is imported by a comparable parser, a complete 3D haptic scene is reconstructed accordingly in the browser. This process ensures the applications designed by user in authoring tool are rendered and shared in the same way as they are created. Then end user is able to manipulate and touch a virtual object in a virtual world by haptic devices. The architecture of the virtual world in the haptic MPEG-V schema is illustrated in Figure 3.21 The semantics of the node element

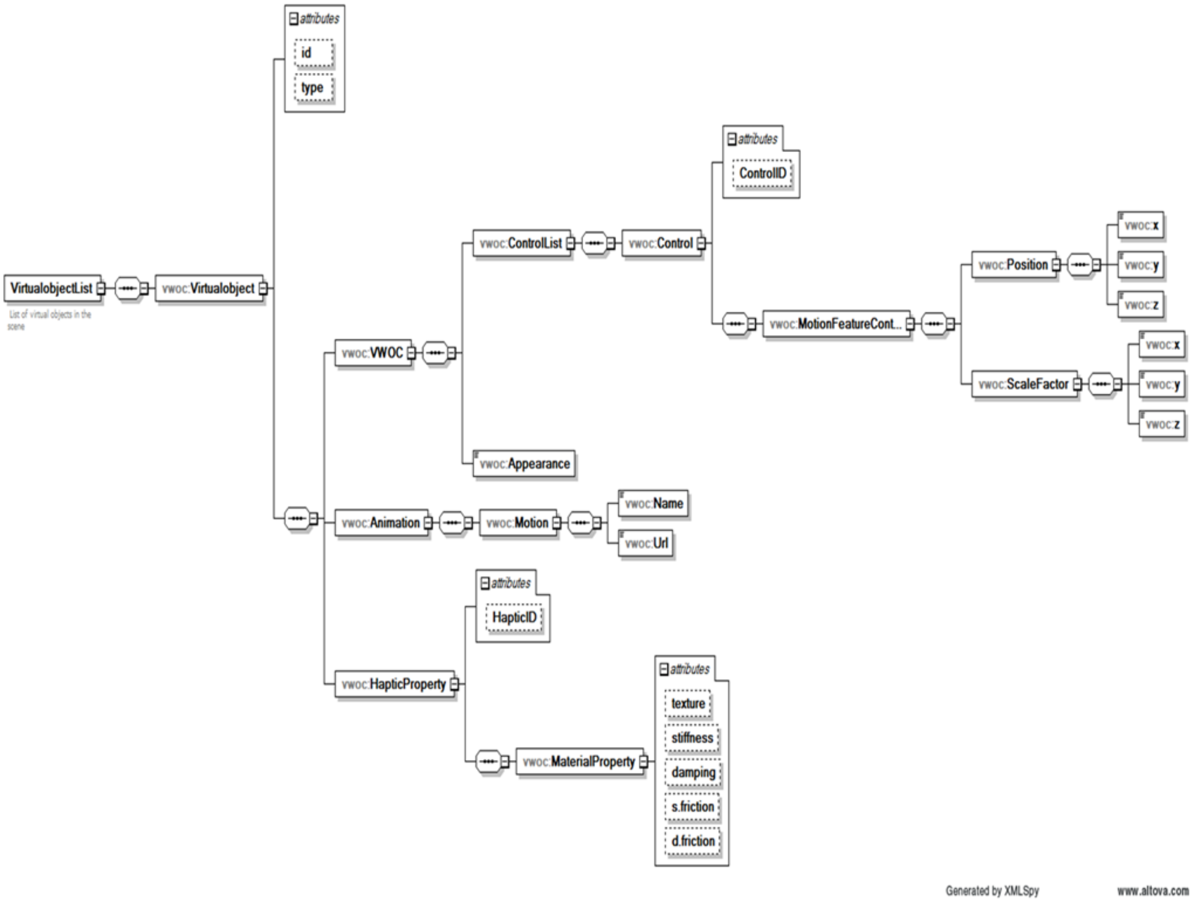


Figure 3.21: Architecture of the virtual world in the authoring tool

in this schema diagram is shown in the following table:

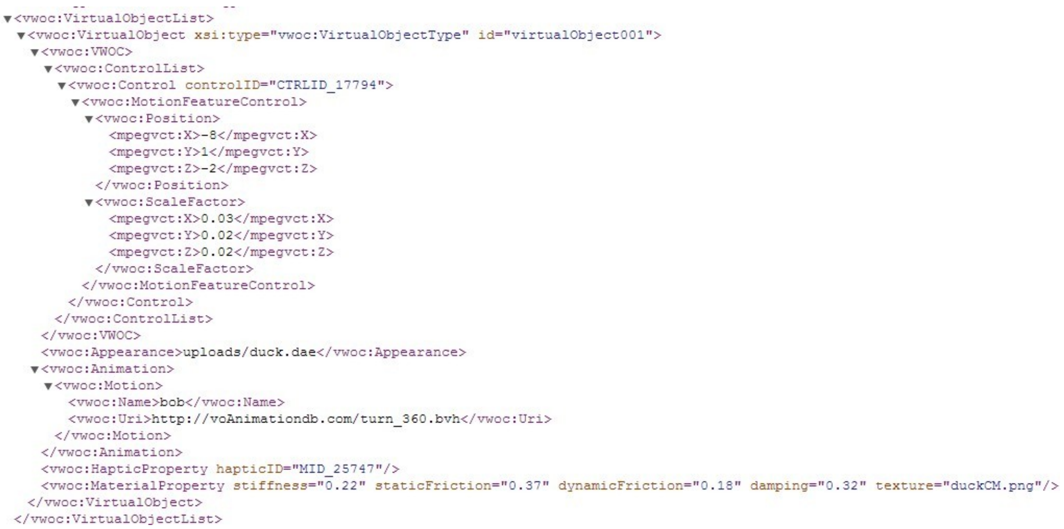
Table 3.1: Sementics of haptic MPEG-V descritpion

Name	Description
VirtualObjectList	A list of virtual objects in the virtual world.

VirtualObject	A virtual object in the virtual world with a unique id and specific type.
VWOC	Characteristics of a virtual world object.
Animation	A set of metadata describes specific animation sequences containing geometric transformations and deformations associated with the object.
HapticProperty	High level description of the haptic property of the virtual world object with a unique HapticID.
HapticID	A unique identifier of the haptic property.
ControlList	Describes a list of the control information of the virtual world object.
Control	Describes a control associated to the virtual world object.
controlID	A unique identifier of the control.
MotionFeatureControl	Containing a set of elements controlling position and scale factors of the virtual object.
Position	The location of the virtual object on x,y,z axis in 3D scene.
ScaleFactor	The scale of the virtual object on x,y,z axis in 3D scene.
Appearance	Contains the link to the source file describing the visual and haptic properties of the object.
Motion	Set of animations represented as a rigid motion.

Name	Name of the animation associated with the virtual object.
Url	Link to the source file of the animation associated with the virtual object.
MaterialProperty	A set of properties describing material characteristics of the virtual object.
stiffness	Stiffness parameter of the virtual object.
damping	Damping parameter of the virtual object.
staticfriction	Coefficient of static friction of the virtual object.
dynamicfriction	Coefficient of dynamic friction of the virtual object.
texture	Contains the link to source file of haptic texture of the virtual object.
mass	The mass of the virtual world object.

In this schema, we chose the namespace of MPEG-V virtual world object characteristics which is “vwoc” as the prefix of every element. The description of all virtual objects in a virtual scene in MPEG-V is included in schema “VirtualObjectList”, in which the prefix vwoc means virtual world object characteristics. Each virtual object has a unique id and a specific type. The schema “VWOC” contains a set of characteristics related to graphic. The control list inside VWOC consists of information related to motion feature control which is composed of three location and scale factors. Followed by control list is the appearance node of the object, it uses the model name of source file. After that, the animation information is described in Motion schema, which contains the name and web link of the pre-defined animation. Then HapticProperty schema containing a specific hapticID describes the haptic properties including stiffness, damping, static friction and dynamic friction of the object in the MaterialProperty node. The texture file mapping to the object is contained in MaterialProperty as well. Figure 3.22 shows an excerpt of a virtual object description in the MPEG-V document generated by Web Haptic Authoring Tool. For the virtual object list shown in the snapshot, a virtual object inside the environment whose id is “virtualobject001” and attribute type is “vwoc:VirtualObjectType” according to “vwoc:virtualobject” is defined.

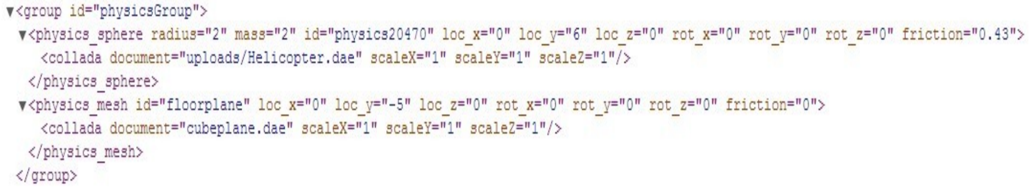


```
▼<vwoc:VirtualObjectList>
  ▼<vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType" id="virtualObject001">
    ▼<vwoc:VWOC>
      ▼<vwoc:ControlList>
        ▼<vwoc:Control controlID="CTRLID_17794">
          ▼<vwoc:MotionFeatureControl>
            ▼<vwoc:Position>
                <mpegvct:X>-8</mpegvct:X>
                <mpegvct:Y>1</mpegvct:Y>
                <mpegvct:Z>-2</mpegvct:Z>
              </vwoc:Position>
            ▼<vwoc:ScaleFactor>
                <mpegvct:X>0.03</mpegvct:X>
                <mpegvct:Y>0.02</mpegvct:Y>
                <mpegvct:Z>0.02</mpegvct:Z>
              </vwoc:ScaleFactor>
            </vwoc:MotionFeatureControl>
          </vwoc:Control>
        </vwoc:ControlList>
      </vwoc:VWOC>
      <vwoc:Appearance>uploads/duck.dae</vwoc:Appearance>
    ▼<vwoc:Animation>
      ▼<vwoc:Motion>
          <vwoc:Name>bob</vwoc:Name>
          <vwoc:Uri>http://voAnimationdb.com/turn_360.bvh</vwoc:Uri>
        </vwoc:Motion>
      </vwoc:Animation>
      <vwoc:HapticProperty hapticID="MID_25747"/>
      <vwoc:MaterialProperty stiffness="0.22" staticFriction="0.37" dynamicFriction="0.18" damping="0.32" texture="duckCM.png"/>
    </vwoc:VirtualObject>
  </vwoc:VirtualObjectList>
```

Figure 3.22: MPEG-V description for a virtual object

3.6.2 HTML5 Graphic elements in MPEG-V

In the previous section we defined an MPEG-V haptic format to describe a haptic enabled virtual world, however the description of 3D scene requires some extra elements including camera, background color and fog. Four kinds of physics objects for enabling the physics effects in haptic application are defined which are not specified in MPEG-V. Thus, for the standardization of the description file, new xml schema, which is according to the GLGE API of HTML5 WebGL standard, is added to the MPEG-V description file. We use it to describe graphic characteristics of the physics objects in the scene. An excerpt of the xml schema which is responsible for representing a virtual object model encapsulated in a sphere in the reconstruction of the 3D virtual world in web is shown in figure 3.23. As we can see in the figure , the xml descriptions of the physics objects are defined in such a way: Firstly, The type of an object in the schema (physics mesh, physics plane, physics box, physics sphere) is defined. Then the properties of them (locations, rotations, friction etc.) are added. Inside the schema of physics object, the model related to the object (a collada file) is needed. Lastly, the properties are assigned into the element (scale and animations).



```
▼<group id="physicsGroup">
 ▼<physics_sphere radius="2" mass="2" id="physics20470" loc_x="0" loc_y="6" loc_z="0" rot_x="0" rot_y="0" rot_z="0" friction="0.43">
   <collada document="uploads/Helicopter.dae" scaleX="1" scaleY="1" scaleZ="1"/>
  </physics_sphere>
 ▼<physics_mesh id="floorplane" loc_x="0" loc_y="-5" loc_z="0" rot_x="0" rot_y="0" rot_z="0" friction="0">
   <collada document="cubeplane.dae" scaleX="1" scaleY="1" scaleZ="1"/>
  </physics_mesh>
 </group>
```

Figure 3.23: Description of a physics mesh in virtual world

3.7 Conclusion

Thus far, we have designed a web-based authoring tool system which performs the proposed haptic application development methodology. Now we can conclude the whole process of how a user design a web application with rich haptic interactions in details. With the help of the proposed system, it takes five steps: **1. Choose the type of application** The user can choose basic mode and physcis-enabled mode based on their application requirements.

2. Upload 3D models into web After the type of hapto-visual application is chosen, users should upload the texture and model files of virtual objects to the MPEG-V engine server. The virtual objects can be created by 3D modeling tools such as Blender by professional artists. The authoring tool currently supports

COLLADA file format.

3. Apply graphic and haptic properties for virtual objects After uploading the model resources to the server, the virtual object will be added into the scene. Users can then select the model by mouse to modify its graphical attributes, including location, rotation, scale and animation. Users can choose from three predefined animations that are provided. If the user selects the physics-enabled mode, the properties of the physics types should be defined for each virtual object. User should choose one shape among four types of shapes: sphere, plane, mesh and box. The physics type will add an imaginary shape that envelops the object.

After the modification of the graphic properties, the user shall assign haptic properties, including stiffness, damping, static friction and dynamic friction to the object using the haptic editor. Additionally, the designer can also define general information about compatible haptic devices for the application in the editor. If the user does so, an MPEG-V description of the haptic device will be added into the application description in the later step of exporting the application.

4. Export MPEG-V description file Once the designer finishes creating the haptic application, the user can then export it into an MPEG-V description file. The MPEG-V web service engine generates a description file based on the attributes the user has assigned to the various objects in the application. The user can download the description file into the local machine in order to be used in the haptic player.

5. Playback in the web haptic player: In order to test the application, the user needs to review it in the web haptic player. Once the user has specified an MPEG-V description file as an input to the player, the scene is rendered in a browser. The user can move the proxy, representing the haptic device, as it is attached to the application in the scene, allowing the user to touch and interact with the virtual objects. For basic mode applications, the virtual environment will be static allowing the user to probe the different objects without disturbing their position. Conversely, for physics mode applications, the user can move objects around and feel them collide with the proxy. Lastly, if the developers want to add new functionality to the application, they can edit the code in the coding area of the graphic and haptic rendering loop. After they modify the code, they can click on the "generate" button to produce the updated application in the browser.

An example of the above five steps will be shown as a proof of concept project next chapter.

Chapter 4

Implementation and Evaluation

In this chapter, we discuss the implementation details of the proposed system. Then we introduce a proof of concept project we developed with the help of the system. After that, we give details of the evaluation process in which experiments and surveys were conducted to evaluate the performance of the system.

4.1 Implementation

– Haptic Device: During the implementation and evaluation process, we used Falcon haptic device from Novint technology. As we can see in the Figure 4.1, the device comes with a 3D Touch controller which is able to track the user's movements in three dimensional real world and send force feedback which is up to 2 lbs to their hands. Its 3D Touch Workspace is 4" x 4" x 4" and the position resolution is up to 400 dpi(Dots per inch).

– Web Server Configuration: The web server we used has an Quad core 2.5 GHz Intel Xeon Processor with 8 MB cache. It also comes with an 8 GB RAM memory and RAID1 hard drive. The server is running with Windows 2003 sp2.

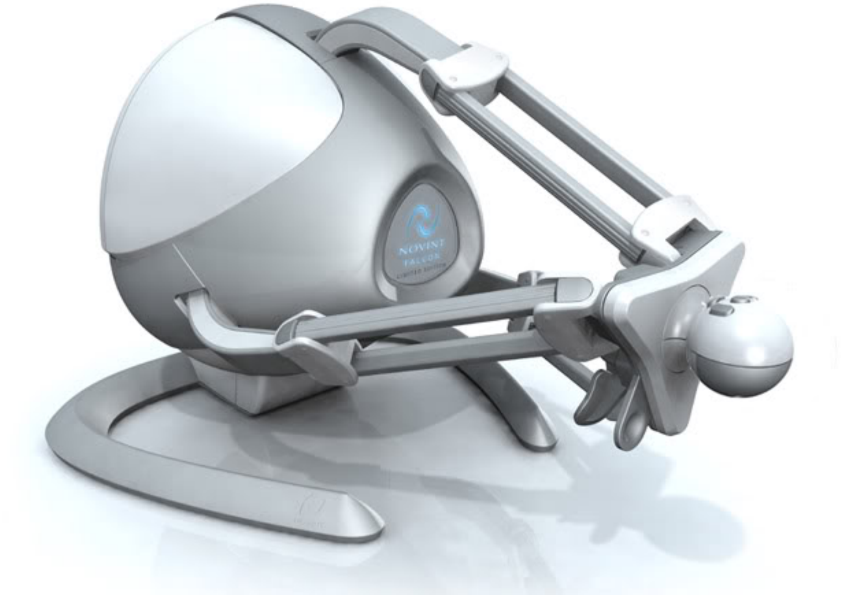


Figure 4.1: Novint Falcon Haptic device

- Programming environment and libraries: The functionality of the system is implemented with javascript and php in Google Chrome. We mainly used GLGE [7] which is a javascript library of WebGL for the graphic rendering and processing of the physics effects. The haptic rendering loop is running in the background by the H5HPlugin developed by Hussein et.al in MCRLab [38]. The MPEG-V engine is written in PHP. The GUI of both the authoring and haptic player have been developed using jQuery UI javascript library and CSS.

4.2 Proof of Concept Project: Development of a HTML5 Haptic Game

Taking into account that the concept of our web haptic application development procedure is an innovative and preliminary idea. A proof of concept project is needed to show the completeness of our methodology. To achieve this goal, we have developed an HTML5 game in a browser by creating a 3D haptic scene in the web haptic authoring tool and adding the code of game logic for virtual objects into the Web Haptic player. Figure 4.2 shows the game in action. The goal when developing this game was to provide a rich haptic experience in a 3D shooting game on the web which can be played directly in the browser using Falcon haptic device. In the game, the objective of the player is to navigate a maze while collecting coins and killing all of the enemies. The virtual world is augmented with haptic effects. Real kinesthetic feedback is applied when specific event in the game happens. The diagram illustrating the logic flow of the game is shown in Figure 4.2. Under normal

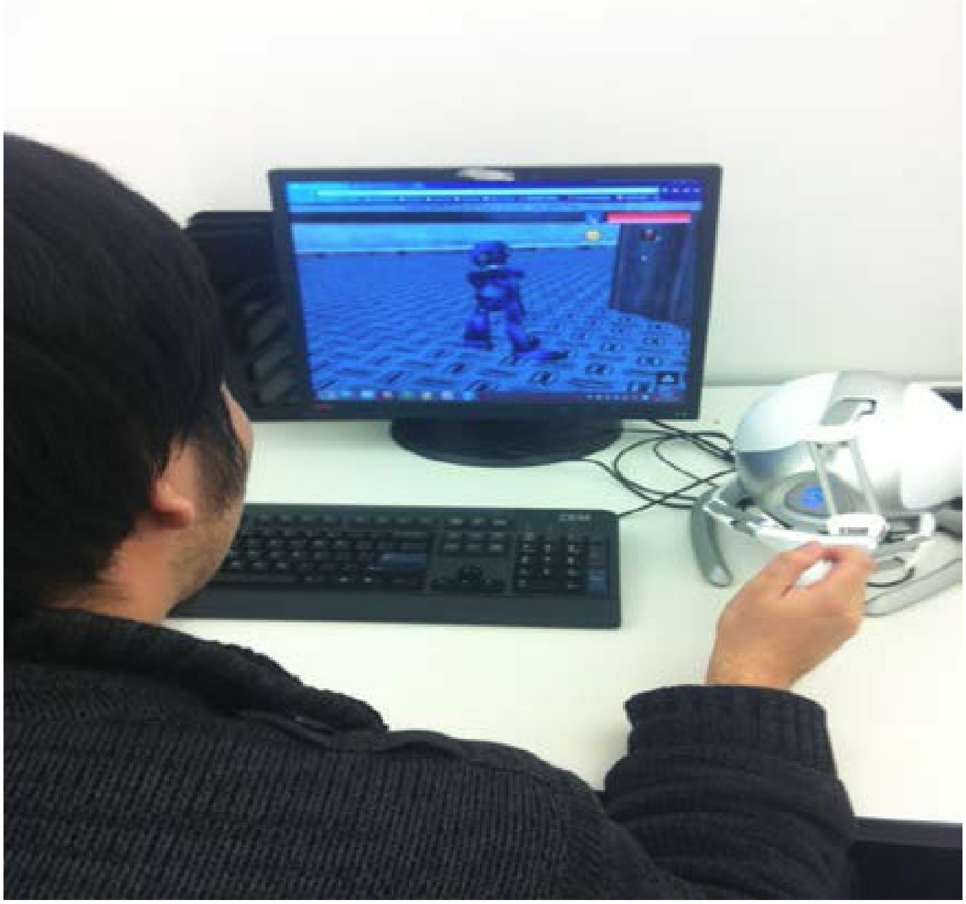


Figure 4.2: HTML5 haptic game developed by authoring tool

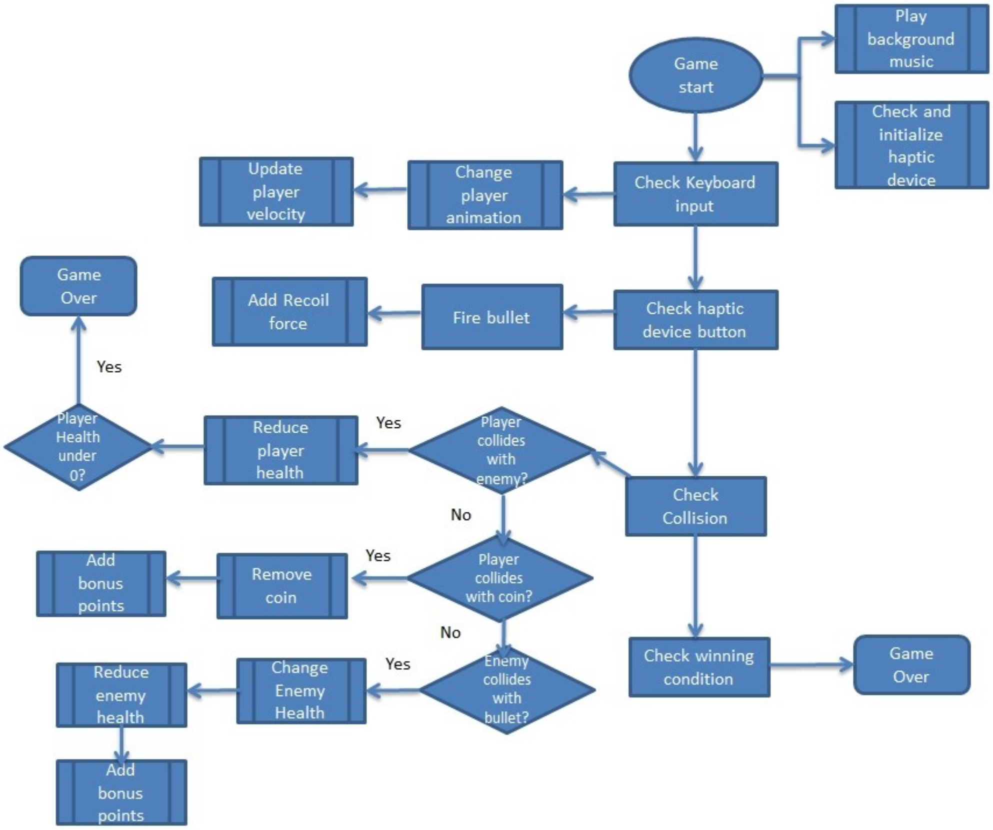
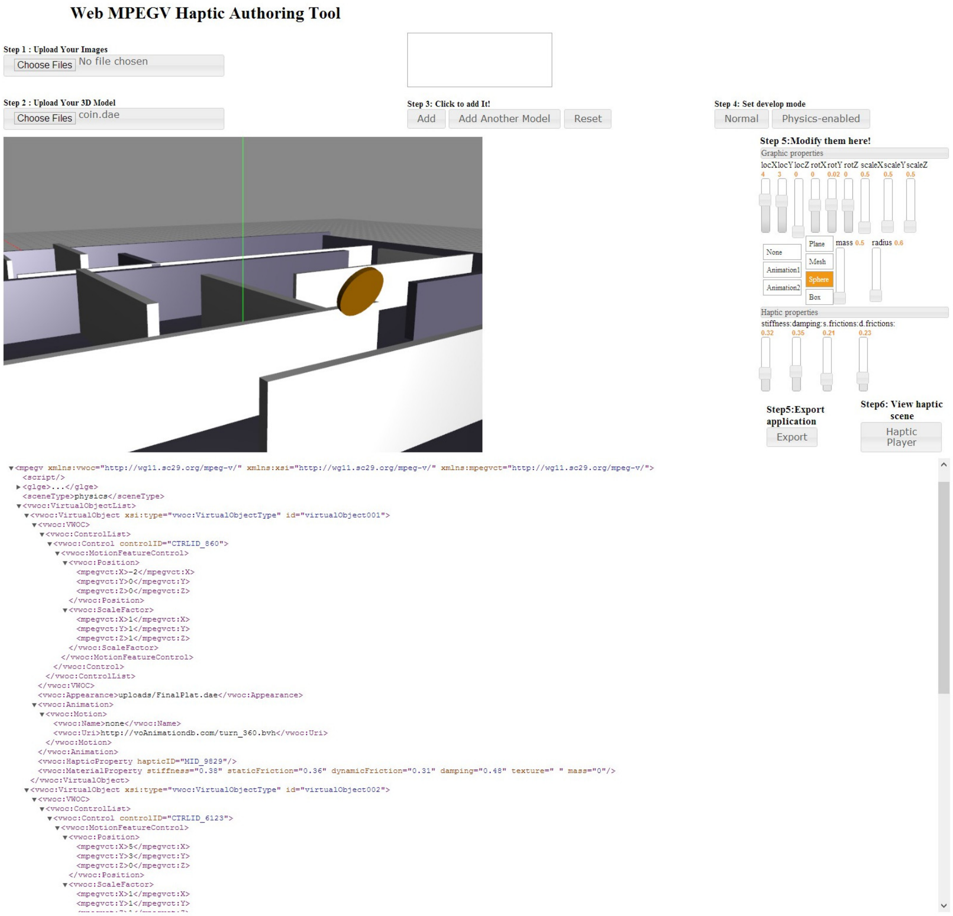


Figure 4.3: Flowchart of game logic

circumstances, the development of such a game requires numerous coding work of creating 3D virtual world, defining collision detection method, adding haptic effects etc. With our system, the work is simplified into five steps: Firstly, considering the goal of this application is a 3D haptic game, we chose the physics-enabled mode in the authoring tool. Then we added 3D models and their corresponding textures. The models include twenty coins, a platform model (including a floor and a wall), and several enemies. Those models were modeled and exported by the 3D modeling platform Blender. We then edited their graphic and haptic properties in the editor interface as is shown in Figure 4.4, set the virtual objects in the locations desired and enabled the animations of enemies. We enveloped the platform with an imaginary mesh shape and other objects with imaginary spheres or boxes for



Web MPEGV Haptic Authoring Tool

Step 1 : Upload Your Images

Choose Files No file chosen

Step 2 : Upload Your 3D Model

Choose Files coin.dae

Step 3: Click to add It!

Step 4: Set develop mode

Step 5:Modify them here!

Graphic properties

locX 4, locY 3, locZ 0, rotX 0, rotY 0.02, rotZ 0, scaleX 0.5, scaleY 0.5, scaleZ 0.5

mass 0.5, radius 0.6

Haptic properties

stiffness: 0.32, damping: 0.35, s.frictions: 0.21, d.frictions: 0.23

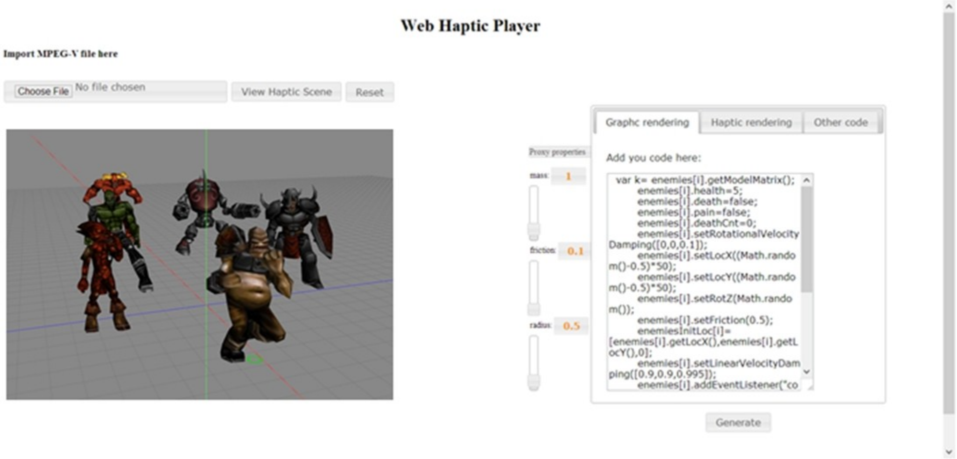
Step5:Export application

Step6: View haptic scene

```
▼<mpegv xmlns:vwoc="http://wg11.sc29.org/mpeg-v/" xmlns:xsi="http://wg11.sc29.org/mpeg-v/" xmlns:mpegvct="http://wg11.sc29.org/mpeg-v/">
   <script/>
  ▶<glge>...</glge>
   <sceneType>physics</sceneType>
  ▼<vwoc:VirtualObjectList>
    ▼<vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType" id="virtualObject001">
      ▼<vwoc:VWOC>
        ▼<vwoc:ControlList>
          ▼<vwoc:Control controlID="CTRLID_860">
            ▼<vwoc:MotionFeatureControl>
              ▼<vwoc:Position>
                 <mpegvct:X>-2</mpegvct:X>
                 <mpegvct:Y>0</mpegvct:Y>
                 <mpegvct:Z>0</mpegvct:Z>
               </vwoc:Position>
              ▼<vwoc:ScaleFactor>
                 <mpegvct:X>1</mpegvct:X>
                 <mpegvct:Y>1</mpegvct:Y>
                 <mpegvct:Z>1</mpegvct:Z>
               </vwoc:ScaleFactor>
             </vwoc:MotionFeatureControl>
           </vwoc:Control>
         </vwoc:ControlList>
       </vwoc:VWOC>
       <vwoc:Appearance>uploads/FinalPlat.dae</vwoc:Appearance>
      ▼<vwoc:Animation>
        ▼<vwoc:Motion>
           <vwoc:Name>none</vwoc:Name>
           <vwoc:Uri>http://voAnimationdb.com/turn_360.bvh</vwoc:Uri>
         </vwoc:Motion>
       </vwoc:Animation>
       <vwoc:HapticProperty hapticID="MID_9829"/>
       <vwoc:MaterialProperty stiffness="0.38" staticFriction="0.36" dynamicFriction="0.31" damping="0.48" texture=" " mass="0"/>
     </vwoc:VirtualObject>
    ▼<vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType" id="virtualObject002">
      ▼<vwoc:VWOC>
        ▼<vwoc:ControlList>
          ▼<vwoc:Control controlID="CTRLID_6123">
            ▼<vwoc:MotionFeatureControl>
              ▼<vwoc:Position>
                 <mpegvct:X>5</mpegvct:X>
                 <mpegvct:Y>3</mpegvct:Y>
                 <mpegvct:Z>0</mpegvct:Z>
               </vwoc:Position>
              ▼<vwoc:ScaleFactor>
                 <mpegvct:X>1</mpegvct:X>
                 <mpegvct:Y>1</mpegvct:Y>
```

Figure 4.4: Screenshot of editing game scene in authoring tool and the MPEG-V description

collision detection purposes. Thus, we have a fully described haptic scene in the browser. Next we exported the application into a MPEG-V file and imputed it into the haptic player. A 3D haptic game was nearly done for this step, however, for a game it was determined that we needed more game logics, such as coin collection events, running, jumping and shooting movements. We also needed to set win and lose conditions for the game, and so forth. Therefore, we added functionality in the graphic rendering and haptic rendering code areas demonstrated in Figure 4.5. The code is written in javascript and supports the game logic by providing functions to check game condition, user input and create haptic effect. A brief introduction to



```
 var k= enemies[i].getModelMatrix();
       enemies[i].health=5;
       enemies[i].death=false;
       enemies[i].pain=false;
       enemies[i].deathCnt=0;
       enemies[i].setRotationalVelocity
Damping([0,0,0.1]);
       enemies[i].setLocX((Math.rando
m()-0.5)*50);
       enemies[i].setLocY((Math.rando
m()-0.5)*50);
       enemies[i].setRotZ(Math.rando
m());
       enemies[i].setFriction(0.5);
       enemiesInitLoc[i]=
[enemies[i].getLocX(),enemies[i].getL
ocY(),0];
       enemies[i].setLinearVelocityDam
ping([0.9,0.9,0.995]);
       enemies[i].addEventListener("co
```

Figure 4.5: Adding game logic code into the game through web haptic player

part of the functions in the game is listed in the subsequent table: As a result we have now developed a basic HTML5 Haptic Game in the browser. The game scene is exhibited in the Figure 4.6.

4.3 Evaluation of the Haptic Effects

We have done evaluations to examine if the force simulations in the system could satisfy the user's needs and whether they are as natural as the corresponding force effects in the real life. In this evaluation phase, 16 subjects particepated in the experiment. Most of whom are from computer science background, to evaluate different kinds of forces in four scenarios. Figure 4.7 shows when a user is evaluating the force.

4.3.1 Evaluation Scenarios

We designed four scenarios representing different haptic effects in the authoring tool. Users were asked to use a Falcon haptic device exploring the scenarios in web

Method	Description
addRecoilForce(var value)	add recoil force to the haptic device when user is shooting the enemy.
addGravityForce(var value)	add gravity force to the haptic device when player is jumping.
addPenaltyForce(var value)	add penalty force to the haptic device when user collides with the enemy.
addCollisionForce(var velocity, var value)	add force when player collides with the wall or the ground
checkKeyboard()	Detect keyboard input and call functions to control avatar accordingly
checkButton()	Check if the button on haptic device is pressed
checkPlayerCollision()	Check the collision on player's avatar: if it collides with an enemy, then the player's heath is reduced. If it collides with a coin, then player's score increases
checkScore()	check the winning condition. If the condition is met, then calls function to jump to the winning page
setAnimation(var object, var anim)	set the animation of player's avatar based on its movement. Or set the animation of enemy when it is shot or dead
fireBullet()	create a bullet object which flies to the direction player's avatar is facing
playSound(var soundNumber)	play the sound effect or music according to game condition

Table 4.1: Description of functions added to the game

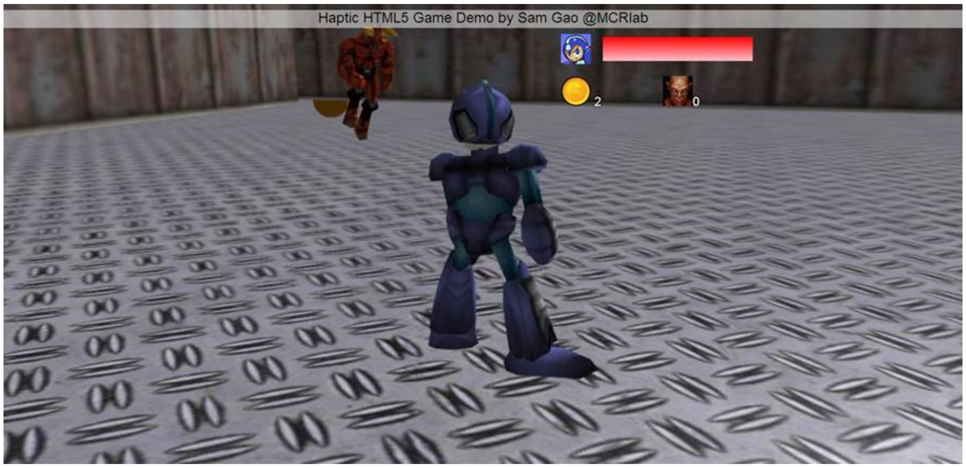


Figure 4.6: The scene of game in browser

haptic player. Figure 4.8 shows the use case diagram of the scenarios. Details of the objectives and contents of the four scenarios are listed as follows:

– Scenario1 evaluates friction force simulation in the system. A user is asked to move the proxy on a plain ground and feel different levels of dynamic friction force when the friction coefficient of the proxy is changing from 0 to 0.5.

– Scenario2 aims to evaluate simulation of spring force and damping force in the system, in this scenario, a user is asked to move the proxy into a sphere in the basic scene and feel different levels of spring and damping force when proxy moves into different depth of the sphere.

– Scenario3 evaluates impulse force in the system. In the scenario, a user is asked to move the proxy on a pool table in the physics-enabled scene and hit the balls on it. The user could feel different levels of impulse force when proxy is moving with different velocities or its mass is changed by user.

– Scenario4 evaluates the physics shape in the system. A user is asked to move the proxy in a maze where there are several objects with enclosed with different physics shapes(sphere, box and mesh) and compare the shape of them through touch perception. Screenshots of the above mentioned scenarios are shown in

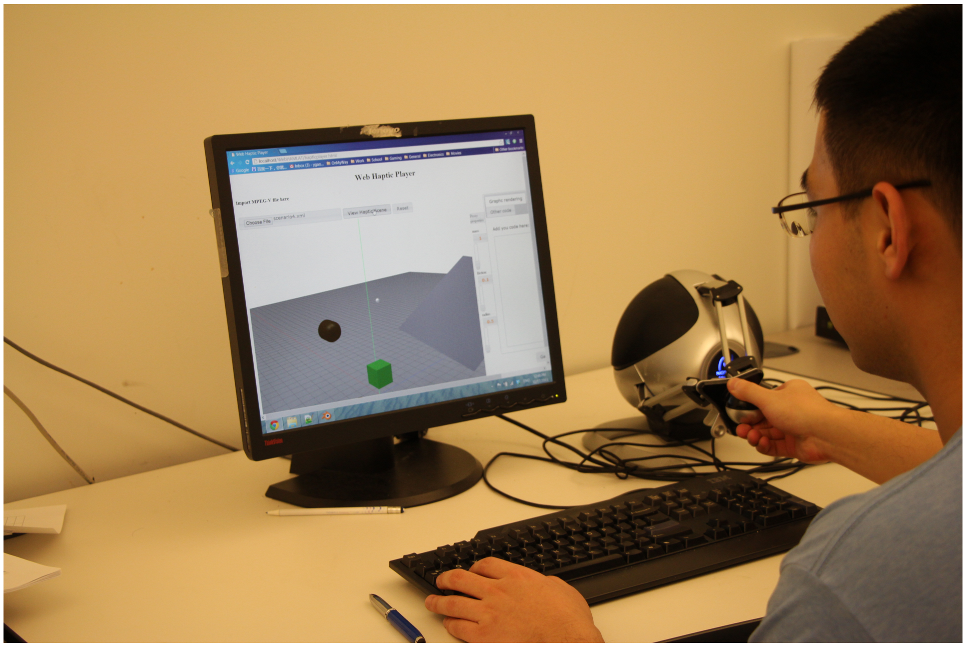


Figure 4.7: A user is evaluating force simulation in scenario 2

Figure 4.9.

Once the user finished feeling the forces in each scenario, they were asked to fill the survey which contains two questions: 1, How natural the force feel? 2, The force correctly reflects the haptic property of the object? For each question, the user should give a score from 1 to 5, where score means the degree of their satisfaction with the scenario.

4.3.2 Results and Summary

The results of the conducted evaluation questionnaires are shown in the following two tables. For each scenario, an average of the results of two questions showing the general ratings from the 16 users and the variance is given. The results of user's answers to the first question are as follows:

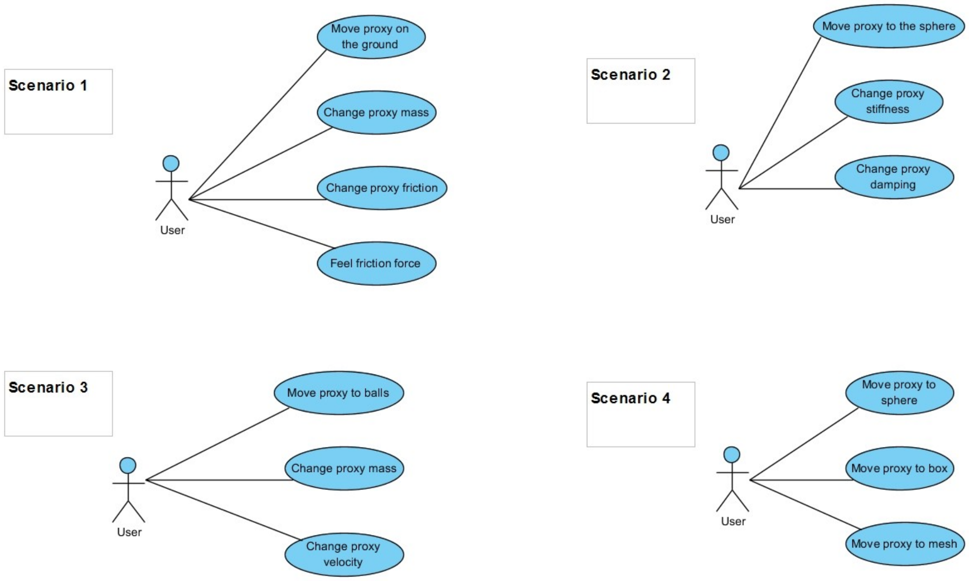


Figure 4.8: Use case diagram of force evaluation scenarios

Scenario	Type of force evaluated	Average	Variance
Scenario1	Friction force	4.065	0.4625
Scenario2	Spring force and damping force	4.0625	0.5586
Scenario3	impulse force	4.25	0.4375
Scenario4	combination of forces	4.125	0.6094

The results of user's answers to the second question are as follows:

Scenario	Properties of object evaluated	Average	Variance
Scenario1	Friction, mass	4.6875	0.2148
Scenario2	Stiffness, damping	4.375	0.3594
Scenario3	Velocity, mass	4.4375	0.3711
Scenario4	Shape, mass	4.3125	0.7148

In order to show more details of the results. We also analyzed the above results using box plots. For each question of each scenario, the range of ratings from the 16 users is shown as a box constructed according to distribution of scores. The bottom

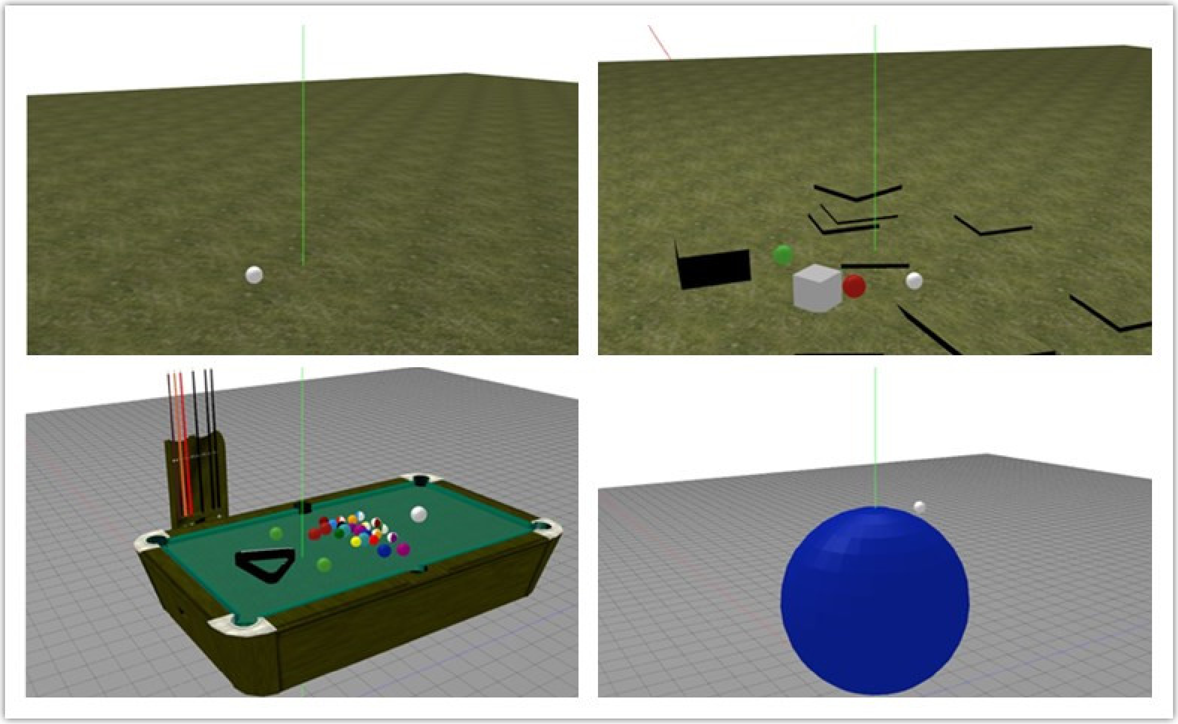


Figure 4.9: Screenshots of four evaluation scenarios

of each box is the 25th percentile, the top is the 75th percentile, and the white box in the middle is the mean value of all ratings. Minimum and maximum ratings are shown in the upper and lower bounds as Crosses. The results of the users' answers to the first question are shown in Figure 4.10, and the results of users' answers to the second question are shown in Figure 4.15: The above evaluation results illustrate the fact that the force simulations in my system are natural to users and allow user to get the touch feeling that similar as real object in the real world.

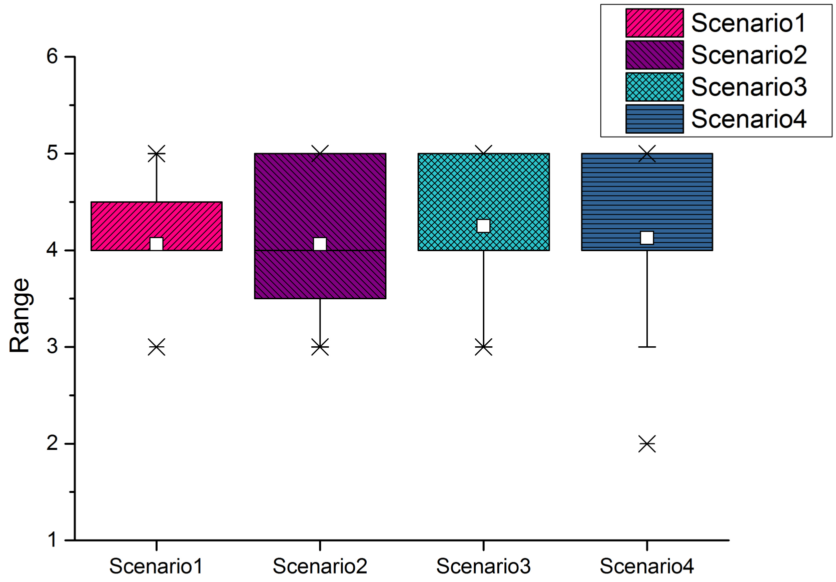


Figure 4.10: Box plot of question 1 in force evaluations

4.4 Evaluation of the Application Development Process

As mentioned in Chapter 3, our goal when developing the authoring tool was to enable users without programming background to build their own haptic application in five simple and standardized steps. A usability study was performed in MCRlab at uOttawa in order to evaluate the application development process. Figure 4.12 shows a female user evaluating the authoring tool.

4.4.1 Evaluation Procedure

During the evaluation, we divided people who participated in the evaluation into several groups according to their level of web programming experience and knowledge of haptics. Then we introduced the whole concept of the system and steps

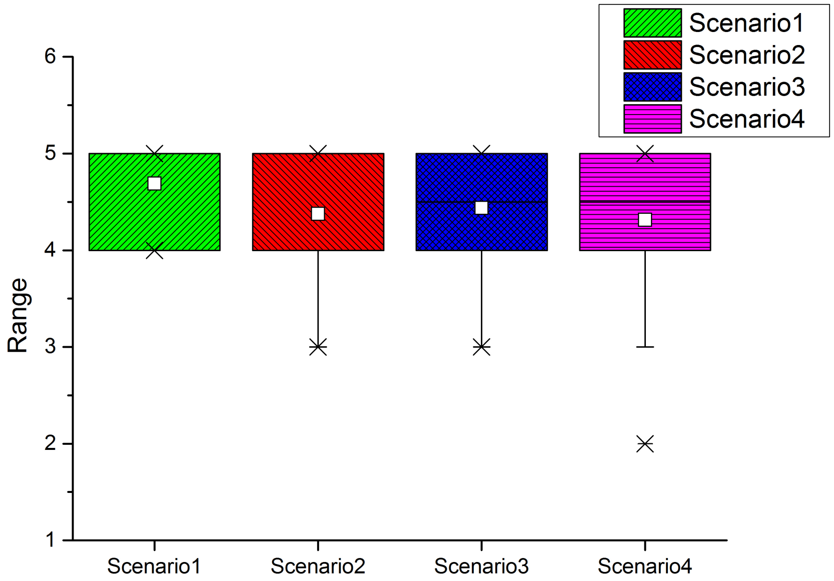


Figure 4.11: Box plot of question 2 in force evaluations

to build a haptic application in web. After that they were asked to develop their own haptic application by the authoring tool. Once a user has finished building the application, she exported the application in MPEG-V file and feels the haptic simulations through haptic player by Falcon haptic device.After each participant completed playing the hapto-visual scene, they were asked to fill a questionnaire composed of six questions. The first four questions are designed for evaluating the application development process in the authoring tool. The last two questions are related to the QoE of whole system. Each participant is asked to rate these statements using the 5-point Likert scale, in which 5 means Strongly Agree and 1 means Strongly Disagree. The statements used in my questionnaire are as follows:

Figure 4.12: A female user is evaluating the authoring tool

Factor	Question
System Functionality	The whole process of building haptic application is complete.
	This authoring tool has all the functions and capabilities we expect it to have.
Quality of Experience	My own-built application satisfies my expectation.
	Ill consider using this system to build haptic application.
Ease of Use	It was easy to learn to use this system.
Overall performance	Overall, I am satisfied with this system.

4.4.2 Results and Comparison

The results of the aforementioned evaluation are shown in the following table. For each question, the average and variance of the ratings from users are calculated. Box plot of the results is given in Figure 4.13.

Question	Average	Variance
The whole process of building haptic application is complete.	4.625	0.234375
This authoring tool has all the functions and capabilities we expect it to have	4.25	0.5625
It was easy to learn to use this system	4.625	0.359375
Ill consider using this system to build haptic application	4.1875	0.277344
My own-built application satisfies my expectation.	4.4375	0.246094
Overall, I am satisfied with this system.	4.5	0.25

From the results of question 5 and 6, we could clearly see that users are satisfied with the proposed methodology of designing haptic application. Average ratings of question 1 and 3 are the highest among all questions and show the ease of use features meet user needs. Results of question 2 and 4 indicates there is much room for improvement of the functionality of our system. Finally, considering the goal of developing the system was to encourage people with no or little programming knowledge to design and feel their own-built haptic application. We divided users into three groups according to their level programming experience, then we compared the ratings of above questions from the three groups. As is shown in Figure4.14, for most questions, users with startup programming knowledge and users with normal programming knowledge gave slighter higher ratings than users who are professional developers: Startup developers gave the highest ratings in the questions related to functionality of the system(question 1,2,4,5). Normal devel-

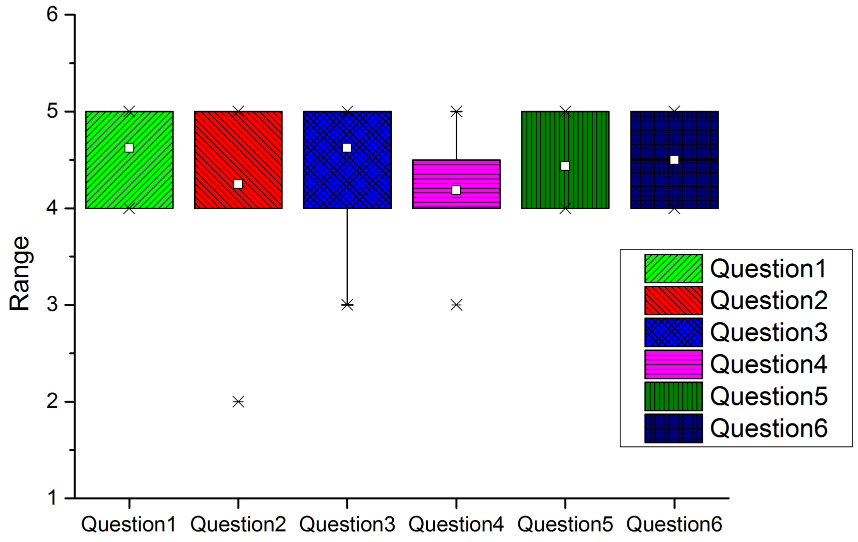


Figure 4.13: Box plot of system evaluation

opers prefer the ease of use and overall performance of the system(question 3,6). This result illustrates the system satisfies the needs of targeted users.

4.5 Evaluation of the User Interface of Authoring tool

For a menu-based multimedia editing tool such as the proposed system, it is important to ensure the graphic interface of the system to be well designed. Thus, along with the evaluation of application development process, we performed an evaluation on the graphic user interface. Users were asked to fill a questionnaire containing five statements regarding to the efficiency, simplicity, Flexibility, User control, organization of the system's user interface. The statements are listed as follows:

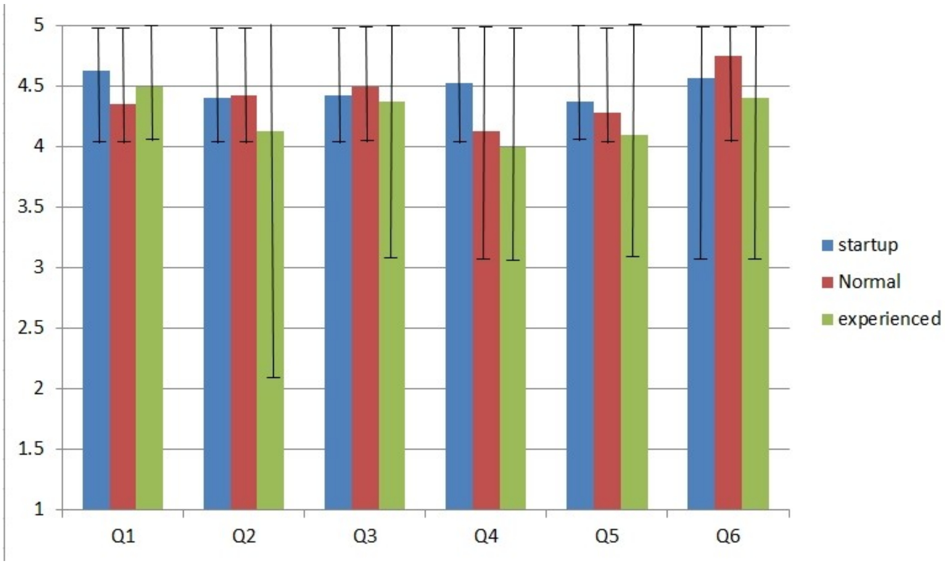


Figure 4.14: Comparison result of different groups

- The interface of this system is easy to use
- The interface of this system looks pleasant for me.
- The organization of different steps in building the application (step1-5) on the authoring tool screens is clear
- The graphic editor matches my needs
- The haptic editor matches my needs

Similar as the evaluation of application development process, we also used 5-point Likert scale for the GUI evaluation questionnaire. The results collected from the same groups of users are shown in the following table.

Question	Average	Variance
The interface of this system is easy to use	4.4375	0.621094
The interface of this system looks pleasant for me.	4.1875	0.652344
The organization of steps in building the application clear	4.375	0.484375
The graphic editor matches my needs	4.0625	0.308594
The haptic editor matches my needs	4.375	0.359375

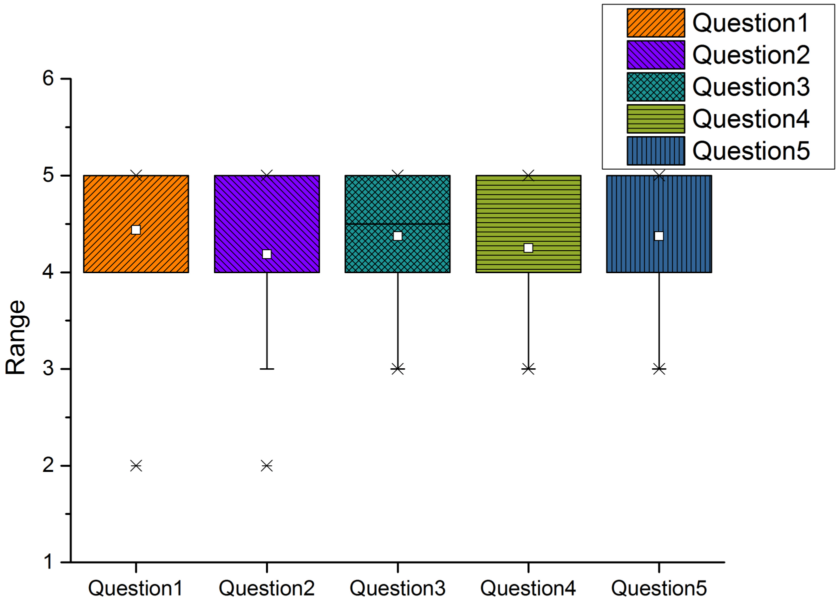


Figure 4.15: Box plot of user interface evaluation

4.6 Discussion

Our system is based on a new concept of web-based haptic content authoring. The proposed method and the system still need to be improved according to users' feedback. While we were conducting the usability study for the whole system, we organized a discussion phase for users to talk about how they feel about the system and give us their recommendations about how my system can be improved. In this phase, a lot of useful advises from different groups of users were collected. The suggestions from users are summarized into the following table. We classified those suggestions into groups according to their purpose.

Table 4.2: User's suggestions in usability study

Purpose	System component	Number of Users proposed	Suggestion
Functionality	Haptic player	Number of Users proposed	User should be able to change the avatar(proxy) when she is exploring the scene.
	Authoring tool	1 user	When adding and editing the objects, user is able to duplicate and remove the same object.
	Authoring tool	1 user	Provide internal 3D models for users, such as UV sphere, cube, plane, cylinder, and animals. Also, provide a list for models added in the scene.
	Authoring tool	2 users	Add preview mode in the authoring tool for allowing users to view and feel the application in authoring tool before exporting it to MPEG-V file.

	Haptic player	1 user	Add more controlling options such as keyboard.
	Haptic player	1 user	Use different textures for the proxy to represent specific material of proxy according to its friction coefficient.
Flexibility	Haptic player	1 user	For users with tactile barrier, add options for them to choose the level of force according to their physical ability.
Portability	the whole system	1 user	Make the system accessible for mobile platforms
Haptic features	Authoring tool	1 user	Provide more haptic content authoring ability such as tactile and temperature
	Haptic player	1 user	Support more haptic devices such as haptic jacket
Interface design	Authoring tool	1 user	Add technical descriptions to sliders and menus in graphic and haptic editor
	Authoring tool	4 user	Improve organization of the menus, use a main menu for application development steps
	Authoring tool	1 user	Add a limit of the 3D scene, the proxy should not move outside of the scene.

Ease of use	Authoring tool	1 user	Add help mode in the menu of authoring tool, a detailed user's manual should be provided under this mode.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

With the fast development of web applications and the new generation of WWW standards, the traditional multimedia contents in the web applications fail to satisfy user's needs for adding touch perception channel besides visual and auditive channel when simulating virtual reality for gaming, educational and other purposes. By integrating haptic contents into the web, users can get a realistic experience of touch in his or her sensory system through interacting with virtual environment enriched by haptic contents such as haptic-enabled games and E-learning systems. However, adding haptic contents into web applications requires designer to posses professional web programming background and deep knowledge of haptics. What's more, it is time-consuming and tedious for users to assign haptic properties to objects and define specific haptic effects inside haptic applications. In order to deal with that problem, in this thesis, we proposed a web-based haptic authoring tool to allow user without programming and haptics background to be able to create haptic applications in a web browser. By using our system, users can add 3D objects into the application and modify their haptic and graphic properties. Once user is done, haptic effects of the objects are generated by the system and user is able to export the application into disk. At last, by importing the description file, end user could explore the scene by haptic device and feel haptic force feedback in the virtual environment designed by themselves. Our work defines a complete method containing five simple steps to ease the development process of web haptic applications. We have designed two types of hapto-visual scenes which enables different types of visual and haptic simulations according to user's purpose of building the application. The collision detection methods are implemented based on Axis-aligned Bounding box algorithm. Besides, we designed haptic simulations of several types of forces which are experienced in real life. For rebuilding and storing application, we have defined a way to describe a web haptic application using MPEG-V xml schema. As a result we have successfully developed a system to facilitate the development of future web haptic applications. Moreover, a proof of concept project was done to test the functionality of our system. We developed a HTML5 game with rich 3D and haptic interactions. Also evaluations were conducted to evaluate the haptic simulations and proposed application development process as well as user interface of the system. Results show that users are satisfied with the design and functionality of our system. At last, we collected feedback and suggestions from users while evaluating the system to improve it in the future.

5.2 Future Work

For the future versions of our system, firstly we would like to improve it based on users' suggestions proposed in the usability study. We will be adding a user's manual to help new users and startup developers know how to use our system. Internal 3D models and other ease to use options such as "duplicate" and "remove" will be added to the authoring tool. In the haptic player, the user will have options to choose the avatar, change force level and choose controlling device. More types of haptic devices, besides devices composed of kinethetic haptic interfaces we are currently supporting, including wearable devices and joysticks will be supported as well. A preview mode will be added to the authoring tool to allow users get the touch feeling of objects they are modifying through haptic device. This work would ensure people assign the preferred haptic properties in their custom haptic application before exporting it. Also, authoring of more haptic contents should be enabled in the tool. The virtual objects in the scene should have properties of tactile information. The system should allow user to choose the feeling of the object on different types of haptic devices. Users will be able to feel the material texture and temperature of an object when they are exploring the scene using a wearable device with heat and vibrotactile actuators such as haptic gloves. As for users who choose to use haptic joystick to play the application, the authoring tool will let designers to set the type and level of vibrations on joystick. Furthermore, a mobile version of the authoring tool and haptic player should be developed and published as a public mobile app. The app will provide a simplified flow to create haptic games in mobile platform for people who use smart phones or tablets. User could develop and test a game on the mobile device and upload it to the server to invite their friends to play. The haptic effects in the game which user creates will be based on vibrations. Finally, we are planning to provide more complex modeling of haptic collaborations and an event listener in the editing interface, as well as enabling sound effects authoring along with the tactile authoring of HTML elements to the authoring tool.

References

[1] Blender's official site. http://www.blender.org/ Accessed October 30, 2013.

[2] Chai3d: The open source haptics project. http://www.chai3d.org/ Accessed October 30, 2013.

[3] Collada - digital asset and fx exchange schema. https://collada.org/ Accessed October 30, 2013.

[4] extreme virtual reality. http://www.vrmedia.it/en/xvr.html Accessed October 30, 2013.

[5] Geomagic haptic devices. http://geomagic.com/en/products-landing-pages/haptic Accessed October 30, 2013.

[6] Geomagic sensable. http://www.sensable.com Accessed October 30, 2013.

[7] Glge, webgl for lazy. http://www.glge.org/ Accessed October 30, 2013.

[8] Immersion corporation haptics for the digital user experience. www.immersion.com Accessed October 30, 2013.

[9] Impulse(physics), wikipedia. http://en.wikipedia.org/wiki/Impulse_(physics) Accessed October 30, 2013.

[10] Introduction of parser, wikipedia. http://en.wikipedia.org/wiki/Parsing Accessed October 30, 2013.

[11] Introduction to html5, wikipedia. http://en.wikipedia.org/wiki/HTML5 Accessed October 30, 2013.

[12] Jiglib - rigid body physics engine - the rowlhouse. http://www.rowlhouse.co.uk/jiglib/ Accessed October 30, 2013.

[13] Minimum bounding box, wikipedia. http://en.wikipedia.org/wiki/Minimum_bounding_box Accessed October 30, 2013.

[14] Mpeg-4 official specifications. http://mpeg.chiariglione.org/standards/mpeg-4 Accessed October 30, 2013.

[15] Mpeg-7 official specifications. http://mpeg.chiariglione.org/standards/mpeg-7 Accessed October 30, 2013.

[16] Mpeg-v offical site. http://wg11.sc29.org/mpeg-v/ Accessed October 30, 2013.

[17] Multimedia communications research laboratory. http://www.youtube.com Accessed October 30, 2013.

[18] Netscape browser, wikipedia. http://en.wikipedia.org/wiki/Netscape Accessed October 30, 2013.

[19] Novint falcon. http://www.novint.com Accessed October 30, 2013.

[20] Opengl:the industry's foundation for high performance graphics. http://www.opengl.org/ Accessed October 30, 2013.

[21] Threejs editor. http://mrdoob.github.io/three.js/editor/ Accessed October 30, 2013.

[22] Threejs offical site. http://threejs.org/ Accessed October 30, 2013.

[23] Touchcon: Haptic interface for instant messenger. http://hyunjoooh.com/2011/11/16/touchcon/ Accessed October 30, 2013.

[24] Virtual reality modeling language. http://www.w3.org/MarkUp/VRML/ Accessed October 30, 2013.

[25] Webgl official site: Opengl es 2.0 for the web, khronos group,. http://www.khronos.org/webgl/ Accessed October 30, 2013.

[26] Youtube. `http://www.youtube.com` Accessed October 30, 2013.

[27] L. Jochen L. Won A. Sheldon, M. Javier. HaptiCast: A physically based 3d game with haptic feedback. In *Futureplay*, 2006.

[28] B. Sandro C. Carlo B. Massimo C. Marcello, T. Franco. Lowering the development time of multimodal interactive application: The real-life experience of the xvr project. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 270–273. ACM, 2005.

[29] T. Ebrahimi C. Pereira. *The MPEG-4 Book*. Prentice Hall PTR, 2002.

[30] G. Andrea C. Sara, M. Davide. A framework for the development of haptic-enhanced web applications. In Marco Brambilla, Takehiro Tokuda, and Robert Tolksdorf, editors, *Web Engineering*, volume 7387 of *Lecture Notes in Computer Science*, pages 253–267. Springer Berlin Heidelberg, 2012.

[31] E. Saddik E. Mohamad, O. Mauricio. A guided tour in haptic audio visual environments and applications. *International Journal of Advanced Media and Communication*, 1(3):265–297, 2007.

[32] M. Bergamasco C. Gottlieb F. Tecchia E. Ruffaldi, A. Frisoli. A haptic toolkit for the development of immersive and web-enabled games. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 320–323. ACM, 2006.

[33] B.S. Martin E.O.Marc. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.

[34] J. Fleureau P. Guillotel N. Mollet M. Christie A. L'ecuyer F. Danieau, J. Bernon. H-studio: an authoring tool for adding haptic and motion effects to audiovisual content. In *ACM symposium on User interface software and technology*, pages 83 – 84, 2013.

[35] M. Sile C. J. Hasser V. L. Bulatov F. Evan, A. Gardner. Web-based touch display for accessible science education. In *In Haptic HumanComputer Interaction*, pages 52–60, 2000.

[36] M. Adams F. Hamza-Lup. Feel the pressure: E-learning systems with haptic feedback. *Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on*, pages 445–450, 2008.

[37] E. Saddik H. Abdelwahab, E. Mohamad. Effect of kinesthetic and tactile haptic feedback on the quality of experience of edutainment applications. *Multimedia Tools Appl.*, 67(2):455–472, 2013.

[38] A. E. Saddik H. Osman, J. Cha. The html5 haptics plugin. In *IEEE International Workshop on Haptic Audio Visual Environments and Games*, pages 130 – 133, 2012.

[39] Y. Kim J. Ryu J. Cha, Y. Seo. An authoring/editing framework for haptic broadcasting: Passive haptic interactions using mpeg-4 bifs. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 274 – 279. IEEE, 2007.

[40] G. Sukhatm J. Hespanha, M. Mclaughlin. *Haptic collaboration over the Internet.* Prentice Hall, 2002.

[41] M. Konstantinos T. Dimitrios K. Nikolaos, V. Konstantinos. 3d hapticwebbrowser: Towards universal web navigation for the visually impaired. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, pages 25:1–25:2, New York, NY, USA, 2010. ACM.

[42] J. Cha A. E. Saddik M. A. Rahman, A. Alkhaldi. Adding haptic feature to youtube. In *Proceedings of the international conference on Multimedia*, pages 1643–1646. ACM, 2010.

[43] A. E. Saddik M. Eid, A. Alamri. Mpeg-7 description of haptic applications using haml. In *Haptic Audio Visual Environments and their Applications, 2006.*, pages 134 – 139, 2006.

[44] J. Feasel L. Kohli M. Whitton R. Hubbold M. Glencross, C. Jay. Effective cooperative haptic interaction over the internet. *Virtual Reality Conference, IEEE*, (5):115–122, 2007.

[45] S. Hughes M. OMalley. Simplified authoring of 3d haptic content for the world wide web. In *Proceedings of the11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 428 – 429. IEEE, 2003.

[46] Y. Takehiko S. Jackson L. Tonya N. Chang, L. Yueqing. Haptic user interfaces for the visually impaired: Implications for haptically enhanced science learning systems. *International Journal of Human-Computer Interaction*, 28(12):784–798, 2012.

[47] K. Moustakas N. Kaklanis, D. Tzovaras. Haptic navigation in the world wide web. In *Universal Access in Human-Computer Interaction. Applications and Services*, pages 707–715, 2009.

[48] K.Tsuneo U. Kouichi N.Hiroaki, F.Yuki. A tangible information explorer using vibratory touch screen. *Computing*, 95(10-11):1053–1071, 2013.

[49] P. Lammertse R.Q. Linde. Hapticmaster 8208; a generic force controlled robot for human interaction. *Industrial Robot: An International Journal*, 30(6):515–524, 2003.

[50] E. Mohamad C. Jongeun S. Abdulmotaleb El, O. Mauricio. *Haptics Technologies: Bringing Touch to Multimedia.* Springer Publishing Company, Incorporated, 1st edition, 2011.

[51] A. Alamri A. E. Saddik S. Andrews, M. Eid. Extending blender: Development of a haptic authoring tool. In *HAVE'2007 - IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, pages 44 – 49, 2007.

[52] D. Mazza S. Comai. Introducing haptic interactions in web application modeling. In *Web Systems Evolution, 2010 12th IEEE International Symposium on*, pages 43 – 52, 2010.

[53] A. Saleh E. Saddik S. Juan, M. Rahman. Web 3.0: A vision for bridging the gap between real and virtual. In *Proceedings of the 1st ACM International Workshop on Communicability Design and Evaluation in Cultural and Ecological Multimedia System*, pages 9–14. ACM, 2008.

[54] M. Tretter M. S. OModhrain S. Zhu, R. Kuber. Identifying the effectiveness of using three different haptic devices for providing non-visual access to the web. *Interacting with Computers*, 23(6):565 – 581, 2011.

[55] A. E. Saddik. The potential of haptics technologies. *Instrumentation Measurement Magazine, IEEE*, 10(1):10 –17, 2007.

[56] M. Hahn Y. Kim, H. Shin. A bidirectional haptic communication framework and an authoring tool for an instant messenger. In *Advanced Communication Technology, 11th International Conference*, pages 2050 – 2053. IEEE, 2009.

[57] B. Stephen Y. Wai, K. Katri. Web-based haptic applications for blind people to create virtual graphs. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 318–389. IEEE Computer Society, 2003.