Haptic User Interface Integration for 3D Game Engines

Gokhan Sengul[1], Nergiz Ercil Çağıltay[2], Erol Özçelik[1],
Emre Tuner[2], and Batuhan Erol[1]

[1] Atilim University Computer Engineering Department, Kızılcaşar Mahallesi,
06836 İncek Gölbaşı - Ankara – Turkey
[2] Atilim University Software Engineering Department, Kızılcaşar Mahallesi,
06836 İncek Gölbaşı - Ankara – Turkey
gsengul@atilim.edu.tr

Abstract. Touch and feel senses of human beings provide important information about the environment. When those senses are integrated with the eyesight, we may get all the necessary information about the environment. In terms of human-computer-interaction, the eyesight information is provided by visual displays. On the other hand, touch and feel senses are provided by means of special devices called "haptic" devices. Haptic devices are used in many fields such as computer-aided design, distance-surgery operations, medical simulation environments, training simulators for both military and medical applications, etc. Besides the touch and sense feelings haptic devices also provide force-feedbacks, which allows designing a realistic environment in virtual reality applications.

Haptic devices can be categorized into three classes: tactile devices, kinesthetic devices and hybrid devices. Tactile devices simulate skin to create contact sensations. Kinesthetic devices apply forces to guide or inhibit body movement, and hybrid devices attempt to combine tactile and kinesthetic feedback. Among these kinesthetic devices exerts controlled forces on the human body, and it is the most suitable type for the applications such as surgical simulations.

The education environments that require skill-based improvements, the touch and feel senses are very important. In some cases providing such educational environment is very expensive, risky and may also consist of some ethical issues. For example, surgical education is one of these fields. The traditional education is provided in operating room on real patients. This type of education is very expensive, requires long time periods, and does not allow any error-and-try type of experiences. It is stressfully for both the educators and the learners. Additionally there are several ethical considerations. Simulation environments supported by such haptic user interfaces provide an alternative and safer educational alternative. There are several studies showing some evidences of educational benefits of this type of education (Tsuda et al 2009; Sutherland et al 2006). Similarly, this technology can also be successfully integrated to the physical rehabilitation process of some diseases requiring motor skill improvements (Kampiopiotis & Theodorakou, 2003).

Hence, today simulation environments are providing several opportunities for creating low cost and more effective training and educational environment. Today, combining three dimensional (3D) simulation environments with these

M. Kurosu (Ed.): Human-Computer Interaction, Part III, HCII 2014, LNCS 8512, pp. 654–662, 2014.

haptic interfaces is an important feature for advancing current human-computer interaction. On the other hand haptic devices do not provide a full simulation environment for the interaction and it is necessary to enhance the environment by software environments. Game engines provide high flexibility to create 3-D simulation environments. Unity3D is one of the tools that provides a game engine and physics engine for creating better 3D simulation environments. In the literature there are many studies combining these two technologies to create several educational and training environments. However, in the literature, there are not many researches showing how these two technologies can be integrated to create simulation environment by providing haptic interfaces as well. There are several issues that need to be handled for creating such integration. First of all the haptic devices control libraries need to be integrated to the game engine. Second, the game engine simulation representations and real-time interaction features need to be coordinately represented by the haptic device degree of freedom and force-feedback speed and features.

In this study, the integration architecture of Unity 3D game engine and the PHANToM Haptic device for creating a surgical education simulation environment is provided. The methods used for building this integration and handling the synchronization problems are also described. The algorithms developed for creating a better synchronization and user feedback such as providing a smooth feeling and force feedback for the haptic interaction are also provided. We believe that, this study will be helpful for the people who are creating simulation environment by using Unity3D technology and PHANToM haptic interfaces.

Keywords: Surgical simulation, haptic devices, game engines, interaction.

1 Introduction

Touch and feel senses of human beings provide important information about the environment. When those senses are integrated with the eyesight, we may get all the necessary information about the environment. In terms of human-computer-interaction, the eyesight information is provided by visual displays. On the other hand, touch and feel senses are provided by means of special devices called “haptic” devices. Haptic devices are used in many fields such as computer-aided design, distance-surgery operations, medical simulation environments, training simulators for both military and medical applications, etc. Besides the touch and sense feelings haptic devices also provide force-feedbacks, which allows designing a realistic environments in virtual reality applications.

Haptic devices can be categorized into three classes: tactile devices, kinesthetic devices and hybrid devices. Tactile devices simulate skin to create contact sensations. Kinesthetic devices apply forces to guide or inhibit body movement, and hybrid devices attempt to combine tactile and kinesthetic feedback. Among these kinesthetic devices exerts controlled forces on the human body, and it is the most suitable type for the applications such as surgical simulations.

According to Robles-De-La-Torre (2006), somesthetic information is critically important for fast, accurate interaction with our environment and without adequate somesthetic feedback, achieving normal and top performance in tasks that require high levels of dexterity is extremely difficult, if not impossible [1]. Today to improve learning and skill development in several fields virtual reality and environments created in computerized environment are providing several benefits. The virtual reality environments today are being developed for education, training, entertaining, planning and future estimations as well as decision making processes. The current virtual reality based technologies are mostly designed based on the visual and auditory senses of human being. This way the user can better understand the virtual environment and get interactions with the objects designed in this computerized environment. On the other hand touch senses are another important feature of human beings for getting information from their environment. Accordingly force-feedback based haptic systems are being developed to provide touch sense for the users and researchers. It is believed that these haptic systems will greatly increase the effectiveness of simulating real-world situations [2]. Several different designs of these force-feedback haptic systems are being implemented for rehabilitation, educational and training purposes in several different areas [2]. Most common ones are surgical simulations and medical training [3-4], military applications, museum displays [5], painting, sculpting, and CAD [6], visualization [7].

Usually these haptic devices could also be customized according to the requirements of the field that the haptic interface is being provided. An example of the haptic usage in the field of surgical training environment is given in Figure 1.

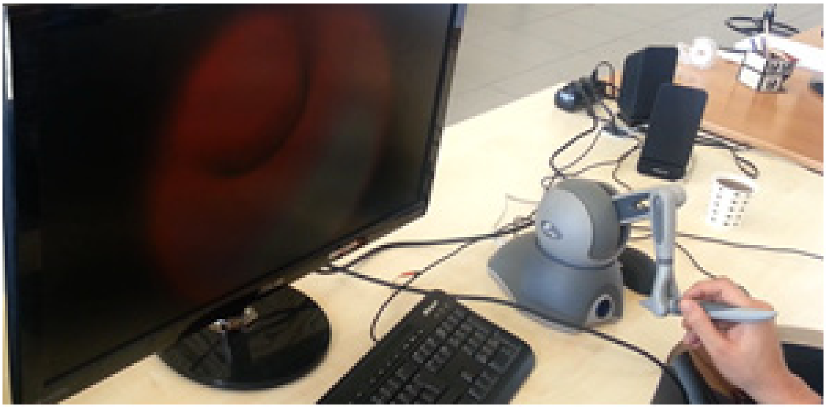

Fig. 1. Haptic Usage for Sugical Simulations

Hence several different designs of these devices with different price and performance ranges are being available. The game engines are one of the environments in which the integration of these haptic devices could provide easy development of software tools for different purposes. However, currently the haptic interfaces are mostly integrated with low-level programming environments and libraries. There are not many examples of integration models for these haptic devices with the game engines.

2 Game Engines and Their Use in Surgical Training Simulations

Creating a virtual surgical training environment requires high man-power and cost since all the parts of the environment requires individually invented and created for every time and for every application. This is the most important problem that has to be solved. All the surgical training environments need at least graphical output, capable of displaying 3D model of the human organs/tissues that needs to be visualized. Besides those, organ/tissue models need to be within a high level of realism, and they need to be able to be controlled easily with the input devices. Besides the simulation environment has to allow the physical simulations such as blood flow etc. The games engines provide high level of solutions to the aforementioned problems, and they allow us to easily generate surgical simulation environments.

A game engine is defined as a system that is used to create and develop video games. The game engines provide a software framework that allows a means to the creators to integrate the software and hardware of a system. The developed games can be run on video game consoles, desktop computers, notebooks and mobile devices.

A game engine consists of functional blocks such as 3D graphics, a physics engine or collision detection (and collision response), event handling, memory management, sound control, scripting, animation, networking, and a scene graph.

The features and details of the functional blocks of a game engine can be found elsewhere and the details are not given here. But among the functional blocks the physics engine has an important position for the surgical training simulations and it is needed to mention about it by a few words.

Distinct from a video game, a surgical training environment requires realistic behavior of the objects in the environment. The physics engine of the game engine implements mathematical models for rigid body simulations of the organs/tissues. Besides by using the physics engine simulation of soft bodies, cloths, fluids such as blood, and smoke caused by the burning of the vessels can be easily provided. So in order to effectively use the game engines in surgical simulations, it is necessary to choose a game engine that has a powerful physics engine that allows the simulations of the aforementioned requirements of the surgical training environment.

Because of the aforementioned reasons and their user friendly options, easy prototyping features, high integrity feature with the other programs, capability of programming with high-level programming languages and their visualization features we decided to use game engines in surgical training simulation environment.

There are a number of game engines available today. Some of them are free, but most of them are commercial products. In order to decide the game engine to use, we evaluated the game engines with the following criteria:

- Soft body simulation support
- Performance in real-time applications
- Visualization capacity
- Integration opportunity with the haptic devices
- License and cost
- Application capability in medical simulations

Since it is a simple and powerful game engine with high performance capability in real-time applications we preferred to use the Unity3D (www.unity3d.com). Unity3D has powerful support and it is used by many users around the world. Besides it provides programming capability by Java Script and C#, and it has high visualization capability. In Figure 2 sample visualization for integration of two haptic devices and Unity3D is given.

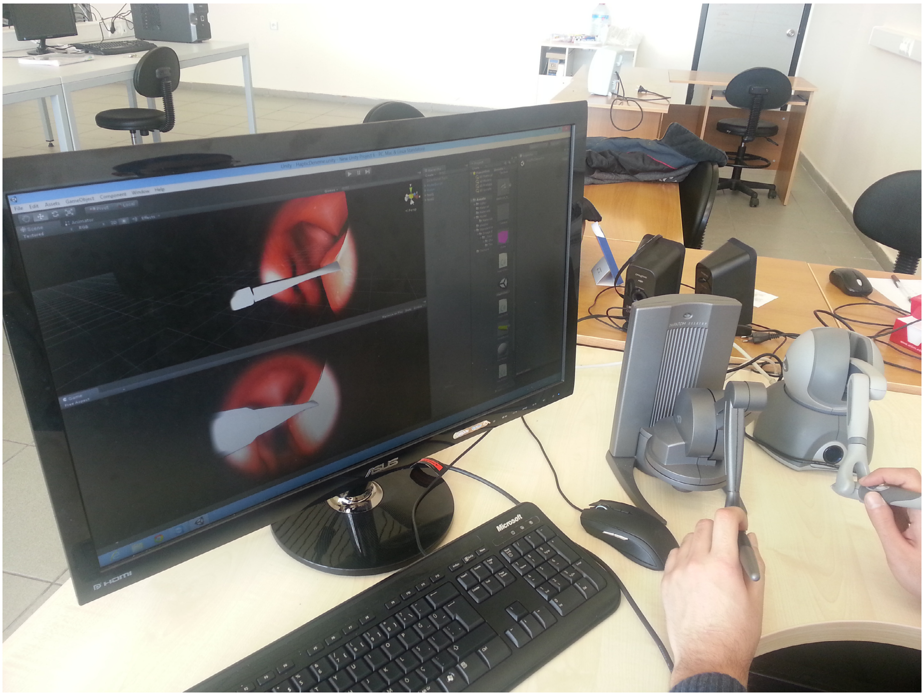

Fig. 2. Sample visualization for integration of two haptic devices and Unity3D

In the following sections we will explain the integration of Unity3D game engine and PHANToM Haptic device for creating a surgical education simulation environment. Also we will give the key points about the integration of haptic device and game engine.

3 Integration of Haptic Devices with Unity3D Game Engine

PHANToM Haptic devices provide a SDK called OpenHaptics library to users to allow easy use of the devices. OpenHaptics library provides two main APIs to programmers, Haptic Device API (HDAPI) and Haptic Library API (HLAPI). Both API can be used with C and C++. HDAPI is a low-level API that allows programmers to control device directly and it does not provide any easy haptic rendering mechanisms. So the programmer has to code the haptic rendering by calculating the force output manually. HLAPI is a high-level API which leverages OpenGL API for graphics rendering. HLAPI provides automatic haptic rendering for OpenGL geometric primitives. Since OpenGL and Unity3D integration is hard, HDAPI is preferred

HDAPI consists of two main components: the device and the scheduler. The device is the basically which haptic device(s) that is planned to use. The scheduler is basically a controller which runs on a different thread, queues and executes the functions that is added to the scheduler list. The general framework that we used for the integration of Unity3D and PHANToM haptic devices is given in figure 3. The details of the each item of the framework are given in the next paragraphs.

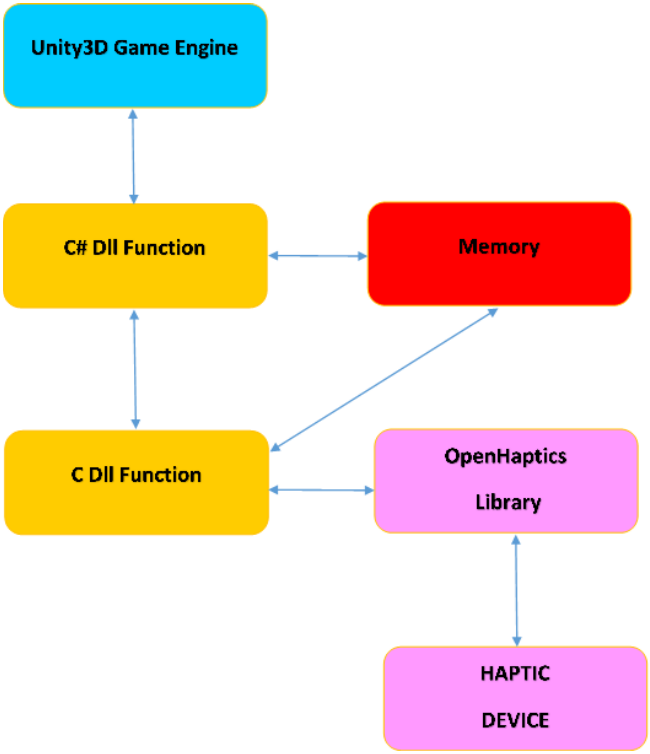

Fig. 3. The general framework for the integration of Unity3D and PAHNToM haptic devices

Since Unity3D uses C# and JavaScript for game scripts, we developed a dll in C which uses HDAPI and import the functions. We preferred to code a dll in C for HDAPI because it is easier to control the device in C than in C#. The requirements for the dll were; the position of the device, the angles of the stylus, the status of the each button on the device, and other functions to control the device and the scheduler. This dll has a structure which contains the necessary information about the device; its position, angles and the status of the buttons. It also has the functions to initialize and disable device, get the current data of the device, apply force, and scheduler functions.

After the development of the dll is completed, we tested it with another C program to see if the data from device is correct and it can apply the force correctly. After the test, we wanted to try it on Unity3D. We have created a C# script in Unity3D and imported my dll functions to that script and we were able to move the objects in game with the haptic device and apply some arbitrary forces to device. But since there will be more than one game, we developed an interface which uses dll and provides easy functions to Unity3D developer.

For this interface we followed the same structure that we did for the dll in C#. Then we created two instances of this structure in C#: one for current data and one for the previous data of the device. Then a function is coded that allocates a memory location as the size of DeviceData structure and assigns the current data instance to that memory location and marshals the structure as pointer, then it sends this pointer to one of the functions on dll. That dll function fills the structure. When the operations about the structure are done, we move forward to complete the class. We used singleton pattern for the class, because there will not be more than one instance of one haptic device. We have also hidden the initialization and starting of the structure from the Unity3D developer, because they need to be done in a specific order. For example, if scheduler is started before initializing, the device scheduler will not be able to find the device. Same applies for the shutting down the device and scheduler also. If the device is closed before stopping the scheduler, it will cause errors.

4 The Problems Encountered during the Integration

During the integration of Unity3D and PHANToM haptic devices, we have encountered many problems and we developed solutions for those problems. In this section we listed the encountered problems and our solution suggestions. We believe that this will help the developers who need to integrate the game engines and haptic devices.

4.1 The Methods Given in the Haptic Library Cannot Be Directly Used in Unity3D Game Engine

All the haptic devices provide their libraries that can be used by the users to be easily use the haptic devices. But those libraries and their methods/functions cannot be directly accessed by the game engines. In order to provide a solution to this problem, we develop two different dll functions (one is written in C and the other is written in C#) as given in the figure 3. Those dll functions include all the library functions of the haptic device. Besides those dlls have C++ library functions as well. By this way a programming environment is develop to integrate the haptic device and game engine. The functions written in C# access this dll functions first, then by using the library functions in this dlls the program is able to access the both the haptic device and game engine.

4.2 Two Haptic Devices Cannot Be Controlled and Used by the Game Engine at the Same Time

In some simulations and training environments it is necessary to use two haptic devices at the same time. Haptic device library (OpenHaptics) provides necessary integration options. Unfortunately it is not possible to connect the two haptic devices at the same with the Unity3D. In order to solve this problem we created a class in C# library and then every object defined in the class is directed to a haptic device. By this way controlling of two haptic devices is provided by the Unity3D.

4.3 Vibration Problem of the Haptic Device in Integration

There was a vibration problem of the device when it is controlled by the game engine. It is found that this was due to two reasons: the differences between the force vectors and low transfer rate of the data to the haptic. In order to solve this problem, we rescaled the force vectors and we limited the magnitude difference of the force vectors and we increased the transfer rate of the data.

4.4 Exceeding the Movement Range of the Haptic Device in Simulation Model

It is necessary to define the movement range of the haptic device in the simulations. If it is not defined, the haptic device movement commands may cause the simulation environment to move further then the available space. In order to solve this problem, we developed an interface to stop the movements when the movement range is exceeded and it also allows users to restart the simulation when the movement range is exceeded.

4.5 Unable to Measure the Force Feedbacks of the Objects That Is Moved by the Force

Unity's current surface force methods cannot allow us to measure the force feedbacks. We extended these methods to be able to measure the force feedbacks.

5 Discussion and Conclusion

The education environments that require skill-based improvements, the touch and feel senses are very important. In some cases providing such educational environment is very expensive, risky and may also consist of some ethical issues. One example to this kind of educational environments is the surgical training education. The traditional education in surgical training is realized in operating rooms on real patients, which is very risky for patients and expensive. The current trend in this education is the usage of simulators which include haptic devices. Simulator based surgical education provides safer environments education [8-10].

But on the other hand, an efficient surgical simulator design requires realistic modeling of human organs/tissues, high level of visualization capabilities and real time interaction with the haptic devices. Those are the major drawbacks of the simulators that make the design and development of the simulator difficult and it requires too much man-power and time. But efficient game engines help the developer to appropriately and easily design and develop simulators. In this study we proposed a framework to integrate the haptic devices and games engines.

We used PHANToM Haptic device and Unity3D as the game engine. We presented the game engine selection criteria as follows: real-time performance of the game engine, soft body simulation capability, flow control capability and capability of the physics engine of the game engine. We suggest that Unity3D is one of the best alternatives that can be integrated with haptic devices to be used in surgical simulator interfaces.

Besides in this study we also proposed a framework to integrate PHANToM Haptic device and Unity3D game engine. We outlined the problems encountered and we suggested our solution approaches. We believe that this work will help the developers of the simulators who need to integrate the haptic devices with the Unity3D game engine.

Acknowledgement. This study is conducted for improving the scenario designs of the educational materials which are developed for neurosurgery education project (ECE: Tubitak 1001, Project No: 112K287) purposes. The authors would like to thank the support of TÜBİTAK 1001 program for realizing the ECE project.

References

1. Robles-De-La-Torre, G.: The importance of the sense of touch in virtual and real environments. IEEE Multimedia 13(3), 24–30 (2006)
2. Bar-Cohen, Y.: Haptic devices for virtual reality, telepresence, and human-assistive robotics. Biologically Inspired Intelligent Robots 73 (2003)
3. D'Aulignac, D., Balaniuk, R.: Providing reliable virtual, echographic exam of the human thigh. In: Salisbury, J.K., Srinivasan, M.A. (eds.) Proceedings of the Fourth PHANToM User's Group Workshop. AI Lab Technical Report No. 1675 and RLE Technical Report No. 633. MIT, Cambridge (1999)
4. Mor, A.B.: 5 DOF force feedback using the 3 DOF PHANToM and a 2 DOF device. In: Proceedings of the Third PHANToM Users Group, PUG98. AI Technical Report no. 1643 and RLE Technical Report no. 624. MIT, Cambridge (1998)
5. Shulman, S.: Digital antiquities. Computer Graphics World 21(11), 34–38 (1998)
6. Dillon, P., Moody, W., Bartlett, R., Scully, P., Morgan, R., James, C.: Sensing the fabric. Paper presented at the Workshop on Haptic Human-Computer Interaction, Glasgow (2000)
7. Durbeck, L.J.K., Macias, N.J., Weinstein, M., Johnson, C.R., Hollerbach, J.M.: SCIRun haptic display for scientific visualization. In: Salisbury, J.K., Srinivasan, M.A. (eds.) Proceedings of the Third PHANToM Users Group, PUG98. AI Technical Report no. 1643 and RLE Technical Report no. 624. MIT, Cambridge (1998)
8. Tsuda, S., Scott, D., Doyle, J., Jones, D.: Surgical Skills Training and Simulation. Curr. Probl. Surg. 2009 46, 271–370 (2009)
9. Sutherland, L.M., Middleton, P.F., Anthony, A., Hamdorf, J., Cregan, P., Scott, D., Maddern, G.J.: Surgical Simulation: A Systematic Review. Ann. Surg. 243(3), 291–300 (2006)
10. Kampiopiotis, A., Theodorakou, K.: Virtual Reality in the Teaching of Motor Skills: Literature Review. Pakistan Journal of Applied Sciences 3(1), 36–39 (2003)