

# Using Unity and Shield UI for Displaying 3D Medical Education Objects in the Web Browser

Vladimir Georgiev<sup>1</sup>, Alexandra Nikolova<sup>2</sup>, Emanuela Mitreva<sup>3</sup>

<sup>1</sup> Computer Science Department, American University in Bulgaria

<sup>2</sup> Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

<sup>3</sup> Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”  
vgeorgiev@aubg.bg, alxnikolova@gmail.com, emitreva@gmail.com

**Abstract.** In this paper we describe how Unity and Shield UI frameworks are used for building the three-dimensional object rendering section of an online knowledge management system for human anatomy. 3D models of humans are organized in multiple levels of hierarchy and each component contains different types of related information. To allow seamless navigation, like searching for and selecting a specific organ, and viewing its medical description in several languages, developers would need to rely on an optimized and stable framework that can target web browsers as well as other platform builds. The specific requirements of this project called for trying the Unity package for building our 3D rendering scene and exporting it to the WebGL platform to make it available to its viewers on all modern web browsers. To build the rich web user interface and bring high-level user experience to the viewers, the Shield UI framework is integrated on top of the Unity WebGL build.

**Keywords:** Unity, Shield UI, WebGL, 3D, e-Learning, Medical Education

## 1 Introduction

In our previous article (Georgiev, 2013) we presented a web application for showing the human body in an interactive 3D format, which allows viewing separate organs and whole systems from different perspectives. That platform would be used by students and other groups like medical personnel and patients. In this paper we describe some specific challenges of the platform and the approaches and technologies we chose for solving them.

One of the major requirements for that system is being easily accessible and constantly available to all its users. As latest developments in the field allow for seamless integration and rendering of 3D content in web browsers (Georgiev, 2013), and Internet has proved as the ultimate media for providing remote content, we chose to implement the platform as a web application. To use it, end viewers only need a recent web browser on their device – a desktop computer, a mobile phone or tablet, and an active Internet connection. The platform itself is a Content Management System (CMS) providing all the necessary functionality for content management, authentication, user management,

billing processing, etc, which provides the highest degree of flexibility to administrators, moderators and end-users.

The other important decision for the project is choosing the right technology to render the three-dimensional models of human anatomy in the web browser. One of the alternatives is using an open-source rendering library or tools that are built on top of and extend such frameworks (Georgiev, 2015). That is the most flexible approach, as it gives the greatest freedom for implementing any requirements using the fundamental APIs and resources. Another option is to use an authoring tool, which allows the creation of a 3D scene along with the required assets like camera, lights and input handling, and then exporting the code to be rendered in a web browser. To avoid the raw implementation of the 3D code, we picked Unity (Unity, n.d.) – one of the popular free tools for building two and three-dimensional applications.

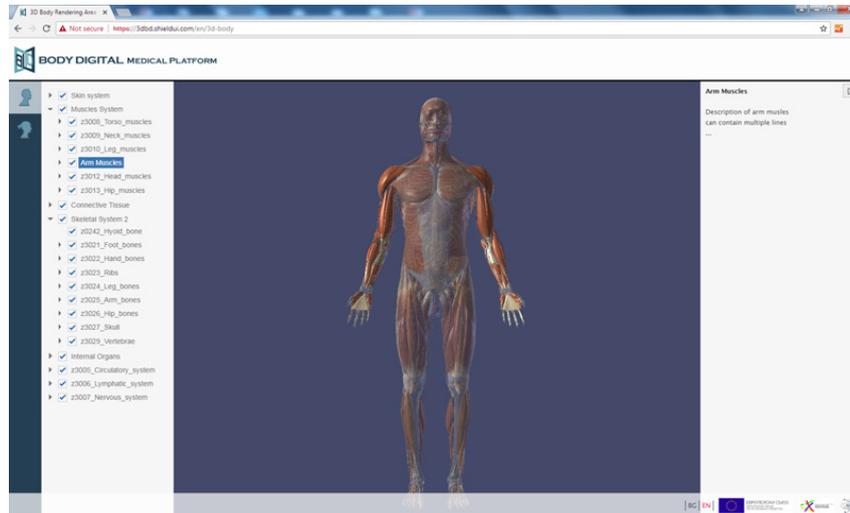
Unity is a powerful tool that provides the general assets out-of-the-box, but it supports very few UI components. The 3D body digital platform makes use of several specific user interface elements like a TreeView for displaying the different systems and contained parts of the human anatomy, allowing the users to toggle their visibility, a Rich Text Editor for managing the related content, various pop-up Windows and other layout containers and etc. Such complex UI widgets are not supported by Unity’s standard libraries and need to be implemented separately. To enable these rich components in the project, we used the Shield UI library – a modern JavaScript and HTML5 framework for building rich user interfaces for web applications (jQuery Plugins and JavaScript UI Framework | Shield UI, n.d.). By leveraging the Shield UI’s highly-customizable controls, we saved a lot of implementation time and effort.

In the next section we are going to reveal the specific features bought by Unity and Shield UI, which we harnessed in our project.

## **2 Using Unity and Shield UI**

Unity is a cross-platform multipurpose game engine for developing two-dimensional and three-dimensional applications, like games and simulations, for computers, consoles and mobile devices. Unity content is created using the Unity Editor – a feature-rich environment for authoring 2D and 3D content, supporting built-in assets like cameras, lights, particle systems, ability to import and export various image, audio, video and 3D file formats, animations, graphics, and physics, and multi-platform build tools. As previously mentioned, the project requirements were to show the hierarchically-organized groups of human anatomy objects on the screen, and make them interactive and responsive to user input. When certain events are triggered in the 3D scene (part of the Unity project), the scene itself, as well as the user interface (part of the website JavaScript context) should be updated. For example, when the user clicks on part of the body, the other anatomy parts are made semi-transparent and the description pane is updated with the correct text. Additionally, events should be fired in the opposite direction – selecting a part from the left tree view hierarchy would make other visible anatomies semi-transparent, indicating the currently “selected” one. Figure 1 below

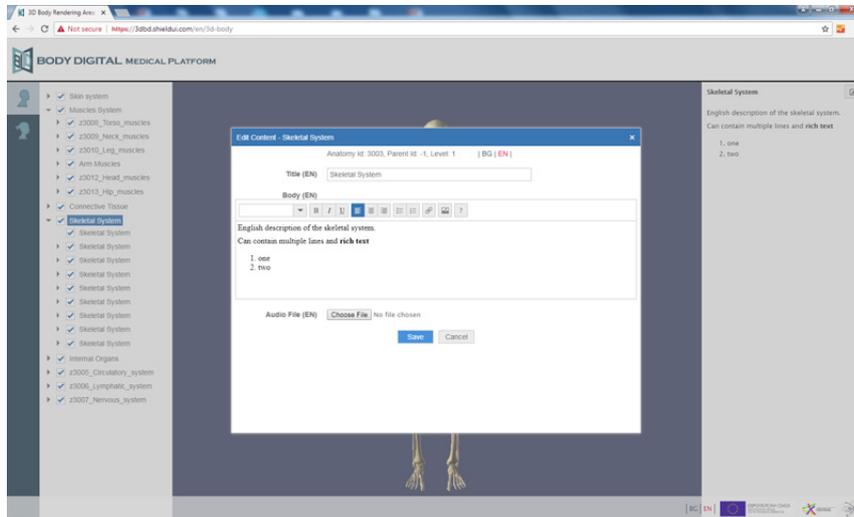
shows this scenario in action, where the arms of a male human model are currently selected from the left tree view and the right side was adequately updated.



**Fig. 1.** Selecting an anatomy in the 3D body viewer

The Unity programming model allows the integration with external JavaScript code, which would be available in the WebGL build (the target code executed in the web browser). This makes it possible to capture events occurring inside the 3D rendering area – like interaction with the object, and propagate them to the JavaScript code, responsible for the rest of the user interface. For sending messages in the other direction – from the web browser’s JavaScript to the Unity scene, there are several alternatives. The most flexible one is to directly invoke methods on the custom script objects defined for the Unity project. Those objects are defined as C# class instances in the Unity project, but since the whole scene is exported as native platform code (in our case JavaScript), it is possible to access them from the other JavaScript code running on the web page.

The ability to send messages between the scene’s and the main page’s JavaScript contexts makes it possible to enhance the 3D content and extend its functionality with rich and impressive UI widgets. The Shield UI framework was selected for no random reason – it provides a complete set of user interface components and each of them is very flexible, highly-customizable and unique in its functionality and support for interactivity. We integrated their TreeView control, which provides a way to present the hierarchical anatomies with the ability to toggle their visibility with a checkbox. Another important UI enhancement was the Editor widget, which enables rich text formatting for the anatomies description and other related data. To facilitate a unique look-and-feel of the platform, we used other widgets like the pop-up Window, TextBox and file Upload components.



**Fig. 2.** Editing the information for an anatomy part

### 3 Conclusion

In this article we presented two key technologies used for displaying 3D content in the web browser – the Unity 3D game engine and the Shield UI JavaScript and HTML5 framework. With their use we created a feature-rich and complex web interface, which meets the requirements of a bigger project – an online e-Learning platform for showing the human body in an interactive three-dimensional format.

### References

- Georgiev, V., Milkov L., Angelov Y., Dimitrov S., (2016). Online 3D Knowledge Management System for Human Anatomy, Proceedings of Sixth International Conference for Digital Presentation and Preservation of Cultural and Scientific Heritage, Veliko Tarnovo, September 27, Bulgaria 2016
- Georgiev, V., (2013). 3D Rendering in the Web Browser, 42nd Spring Conference of the Union of Bulgarian Mathematicians, Borovets, April 2-6, Bulgaria 2013
- Georgiev, V., (2015). Tools for Presenting 3D Objects and Virtual Collections Online, Proceedings of Fifth International Conference on Digital Presentation and Preservation of Cultural and Scientific Heritage, Veliko Tarnovo, September 28-30, Bulgaria 2015
- Unity (n.d.). Retrieved from <https://unity3d.com/>
- jQuery Plugins and JavaScript UI Framework | Shield UI (n.d.). Retrieved from <https://www.shieldui.com/>

Received: June 10, 2018  
 Reviewed: June 30, 2018  
 Finally Accepted: July 10, 2018