MyHealthAvatar

A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information

Project acronym: MyHealthAvatar

Deliverable No. 8.1
Display Suite for Avatars and Evaluation Report

Grant agreement no: 600929

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	MyHealthAvatar
Project Full Name:	A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information
Deliverable No.:	D8.1
Document name:	Display Suite for Avatars and Evaluation Report
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	Final
Actual Submission Date:	04/09/2014
Editor: Institution: E-Mail:	Feng Dong BED feng.dong@beds.ac.uk

ABSTRACT:
This deliverable reports deliverable 8.1 - display suite for avatars and evaluation report in this project.

KEYWORD LIST:
avatar, 3d, human body, anatomy, WebGL

[1] **R**=Report, **P**=implementation, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o} 600929.

The author is solely responsible for its content, it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

MODIFICATION CONTROL			
Version	**Date**	**Status**	**Author**
1.0	**28/08/2014**	**Draft**	Youbing Zhao
2.0	**29/08/2014**	**Draft**	Youbing Zhao
3.0	**31/08/2014**	**Final**	Youbing Zhao

Contents

1 Executive Summary

This document reports deliverable 8.1 - the display suite for avatars and evaluation report - which is the achievement of task 8.1 between month 1-18. Here is a summary of the major points:

1) The web-based 3d avatar rendering suite has been implemented and tested.
2) The online version of the web-based 3d avatar rendering suite has been integrated into MyHealthAvatar platform and can be accessed via http://myhealthavatar.org
3) The evaluation of the avatar rendering suite has also been carried out. The responses show that the avatar rendering suite is an effective and efficient tool for visual analytics of 3d human anatomy.
4) We are continuing our effort to improve the integration of the web-based 3d avatar rendering suite with future MyHealthAvatar functions.

1.1 Project background

Owing to the highly fragmented health systems in European countries, gaining access to a consistent record of individual citizens that involves cross-border activities is very difficult. MyHealthAvatar is an attempt at a proof of concept for the digital representation of patient health status. It is designed as a lifetime companion for individual citizens that will facilitate the collection of, and access to, long-term health-status information. This will be extremely valuable for clinical decisions and offer a promising approach to acquire population data to support clinical research, leading to strengthened multidisciplinary research excellence in supporting innovative medical care.

MyHealthAvatar will be built on the latest ICT technology with an aim of engaging public interest to achieve its targeted outcomes. In addition to data access, it is also an interface to access integrative models and analysis tools, utilizing resources already created by the VPH community. Overall, it will contribute to individualized disease prediction and prevention and support healthy lifestyles and independent living. It is expected to exert a major influence on the reshaping of future healthcare in the handling of increased life expectancy and the ageing population in Europe. This complies with the priority and strategy of FP7 ICT for healthcare, and constitutes a preparatory action aiming at the grand challenge on a "Digital Patient", which is currently the subject of a roadmap in the VPH community.[3]

The MyHealthAvatar project focuses on research and demonstration actions, through which the achievability of an innovative representation of the health status of citizens, named 4D MyHealthAvatar, is explored. The 4D Avatar is anticipated as an interface that will allow data access, collection, sharing and analysis by utilizing modern ICT technology. It is expected to become the citizen's lifelong companion, providing long-term and consistent health status information of the individual citizen along a timeline representing the citizen's life, starting from birth. Data sharing will be encouraged, which will potentially provide to an extensive collection of population data to offer extremely valuable support to clinical research. The avatar will be equipped with a toolbox to facilitate clinical data analysis and knowledge discovery.

MyHealthAvatar can be described as a personal bag carried by individual citizens throughout their lifetime. It is a companion that will continually follow the citizen and will empower them to look after their own health records. This fits very well into the recent trend of developing patient-centred healthcare systems.

[3] MyHealthAvatar project, Description of Work (DoW) document.

1.2 Avatar Modelling and Rendering Suite

This deliverable D8.1 - display suite for avatars and evaluation report - belongs to task 8.1 Avatar modelling and rendering suite (PM3=>PM18) in WP8 Avatar centred visual analytics (PM3-PM33) . The object of the task is to provide a set of display functionality for avatar rendering as one of the visualisation tools for visual model and data analysis. This report describes the work, achievements and evaluation report of the avatar modelling and rendering suite.

2 Deliverable D8.1

This chapter will describe the details of D8.1.

2.1 *Introduction*

Relevant deliverables

D8.1) Display suite for avatars & evaluation report: [month 18].

Relavant Tasks

Task 8.1 Avatar modelling and rendering suite [month 3 - month 18] – achieved

This task is to construct a generic avatar. The objective of this task is to provide a set of display functionality for avatar rendering. The human geometry model should include a number of layers to portray human anatomy. High-resolution anatomy data from the third party is used to build the geometric model. A 3d avatar is planned to be built on the web, allowing for interactive body part selection and highlighting, and interactive part transparency adjusting.

Relevant milestones

MS081 Avatar rendering (initial version) [month 12] – achieved

Status

A web-based avatar rendering suite using three.js has been implemented and released online as a stable component of the MyHealthAvatar platform at http://myhealthavatar.org. The current work supports rendering, interactive viewing, part selection and model picking. A variety of browsers have been tested to ensure that it works properly. Domain experts from the MyHealthAvatar partners have evaluated the avatar suite.

Publications

1. Youbing Zhao, Xia Zhao, Feng Dong, Gordon Clapworthy, Nikolaos Ersotelos, Enjie Liu: WebGL-based interactive rendering of whole body anatomy for web-oriented visualisation of avatar-centered digital health data. BIBE 2013: 1-4

2. Nikolaos Ersotelos, Xia Zhao, Youbing Zhao, Hui Wei, Enjie Liu, Gordon Clapworthy, Feng Dong: A user interface design for a patient oriented digital patient. BIBE 2013: 1-4

2.2 *Background*

2.2.1 WebGL, Three.js and SceneJS

With the evolution of OpenGL [1] and the advent of HTML5 and WebGL (Web Graphics Library) [2], the 3D graphics capability of web browsers has increased considerably.

WebGL is a JavaScript API for rendering interactive 3D graphics in web browsers. WebGL elements can be mixed with other HTML elements and composited with other parts of the page. The latest versions of Firefox, Chrome, Safari support WebGL, and Internet Explorer also started the support of WebGL from version 11.

With WebGL one can write OpenGL programs that render in a web page. However, since version 3.1, OpenGL no longer supports backward compatibility, which implies that the old programming paradigms in OpenGL 1.x and 2.x are no longer favoured, and WebGL users need to write shaders by themselves. This is inconvenient for some programmers who know little about OpenGL shaders.

Three.js [3] is an open-source javascript library, based on WebGL, that is designed to fill the gap by simulating the old OpenGL style programming paradigms such as lighting, material, camera, etc. It is easier to use and does not require knowledge about shaders.

SceneJS [4], another open-source javascript library based on WebGL, provides a JSON-based scene graph API and supports scene graph management. It was created for the efficient rendering of large numbers of objects.

2.2.2 Web-based Human Body Visualisation

There are two major 3D human body websites, one is Zygote Body [5] and the other is BioDigital Human [6]. Both provide interactive visualisation of the whole-body male and female anatomy. The model quality of Zygote Body is better than that of BioDigital Human. Zygote Body shows the whole body after model loading, while BioDigital Human displays a skeleton initially and lets the user select other parts from the anatomy tree to show. The memory management of Zygote Body also outperforms BioDigital Human, and while the latter provides some special visualisations of conditions, none of these aims at health visual analytics. Zygote Body uses WebGL directly and BioDigital Human uses SceneJS as its WebGL engine.

2.3 Technology

In the implementation of the avatar rendering suite, we use three.js for the web-based 3D graphics . In addition, three.js also comes with loaders to load Wavefront OBJ format directly. We also use JQuery [7] for the web user interface.

2.3.1 Model Preparation

We obtained a 3D surface-model set of the whole human body of both the male and the female from an online model vendor. The model set comprises 56 3D Studio Max [8] surface models with textures, each consisting of anatomical sub-object groups identified by their Latin names. The quality of the model is medium but the structure is well organised.

The resolution of the model is too high for direct use in web applications. The loading of the unsimplified models will consume a great deal of memory and will result in the browser freezing or crashing. Consequently, the models have been simplified by processing with the ProOptimizer in 3D Studio Max; sometimes, multi-pass mesh reduction or interactive reduction is needed as automatic processing may not fulfil the task. After simplification, most of the models are reduced to 5-10% of their original size, some even lower. For example, the number of vertices in the skull model is reduced from 222k to 8k.

2.3.2 Model Loading

We use Wavefront OBJ as the model format since plain text OBJ files are easy to read and it has many importers and exporters. Materials and textures are desirable for realistic anatomy rendering, and it supports these well. Three.js provides an OBJ loader with ObjMTLLoader.js and MTLLoader.js. In addition to texture mapping, the code can be easily adapted to support bump mapping, but to save memory we decided not to enable bump mapping as it almost doubles the texture memory consumption.

In the current implementation, all parts are preloaded. The model names and storage location are retrieved from a database dynamically. There are multiple model files to be loaded and all of them are downloaded in parallel by the OBJMTLLoader. The loading time of the whole model set is dependent on the internet connection and the local cache - this usually ranges between seconds and several minutes. A progress bar is displayed to show the loading progress. Figure 1 shows the 3d avatar skeleton and the user interface of the avatar rendering suite.

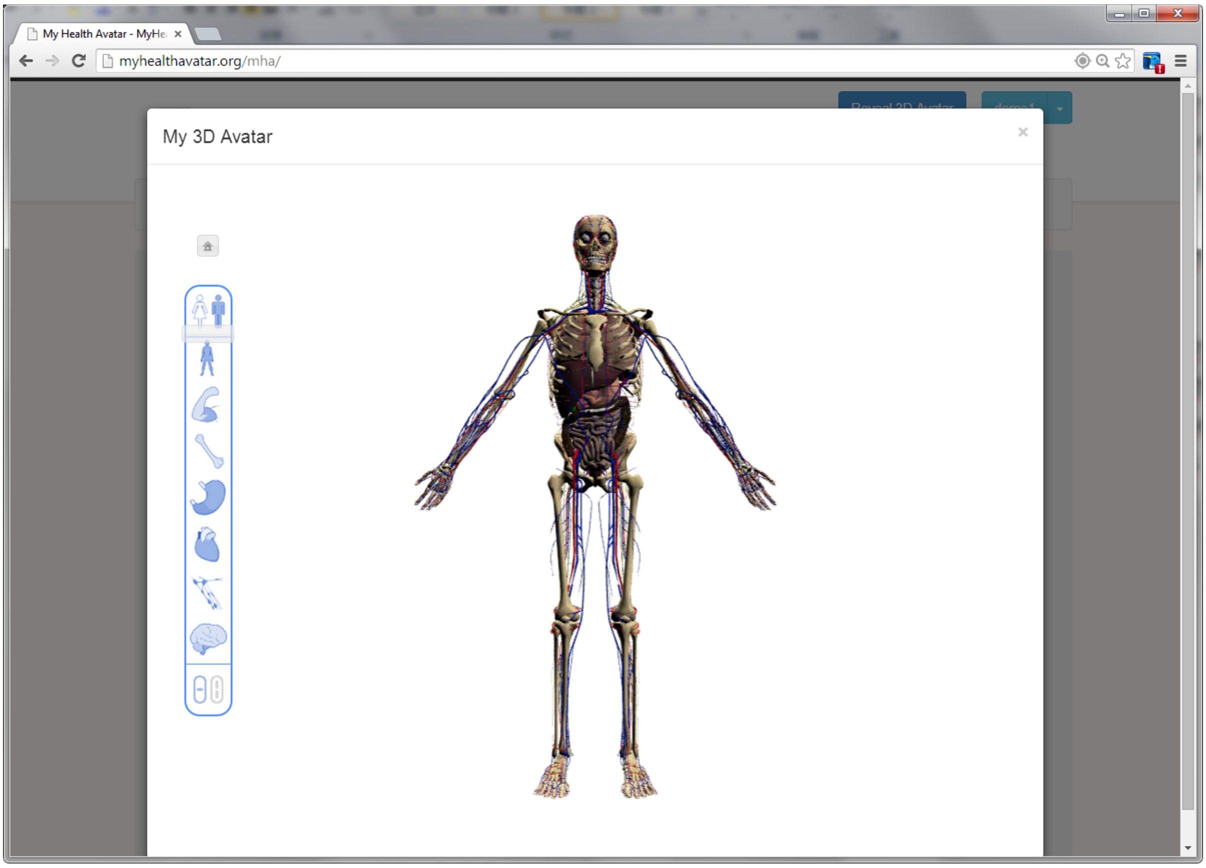

Figure 1. A skeleton after all model parts have been loaded

2.3.3 Interactive Model Rendering

The core components in three.js for rendering are the scene, renderer, camera and lights. All the models that are shown in the browser are stored in the scene; when an object is to be hidden, it is removed from the scene. The renderer is of type THREE.WebGLRenderer and accepts a scene and a camera as parameters to perform the rendering. Lights and camera settings are very similar to those in the old-style OpenGL programming. Trackball controls are attached to the camera to support interactive model exploration with rotating, zooming and panning.

2.3.4 Part Selection

To explore the human anatomy model set efficiently, selective rendering is desired as there are too many body parts and many of them occlude each other.

We use a toolbar similar to Zygote Body to adjust part selection and to set part transparency, as shown in Figure 1. There are two modes for the toolbar, the single vertical slider mode and the multiple horizontal slider mode. Under the single vertical slider mode, the toolbar acts as a vertical toolbar for de-selecting parts from the outside to the interior. Under the multiple horizontal slider mode, the toolbar acts multiple horizontal sliders which can be adjusted for different parts separately. Consequently the user can select to show any combinations of parts he desires.

Figure 2 shows a model with a semi-transparent skeleton and non-transparent internal organs after user adjust.

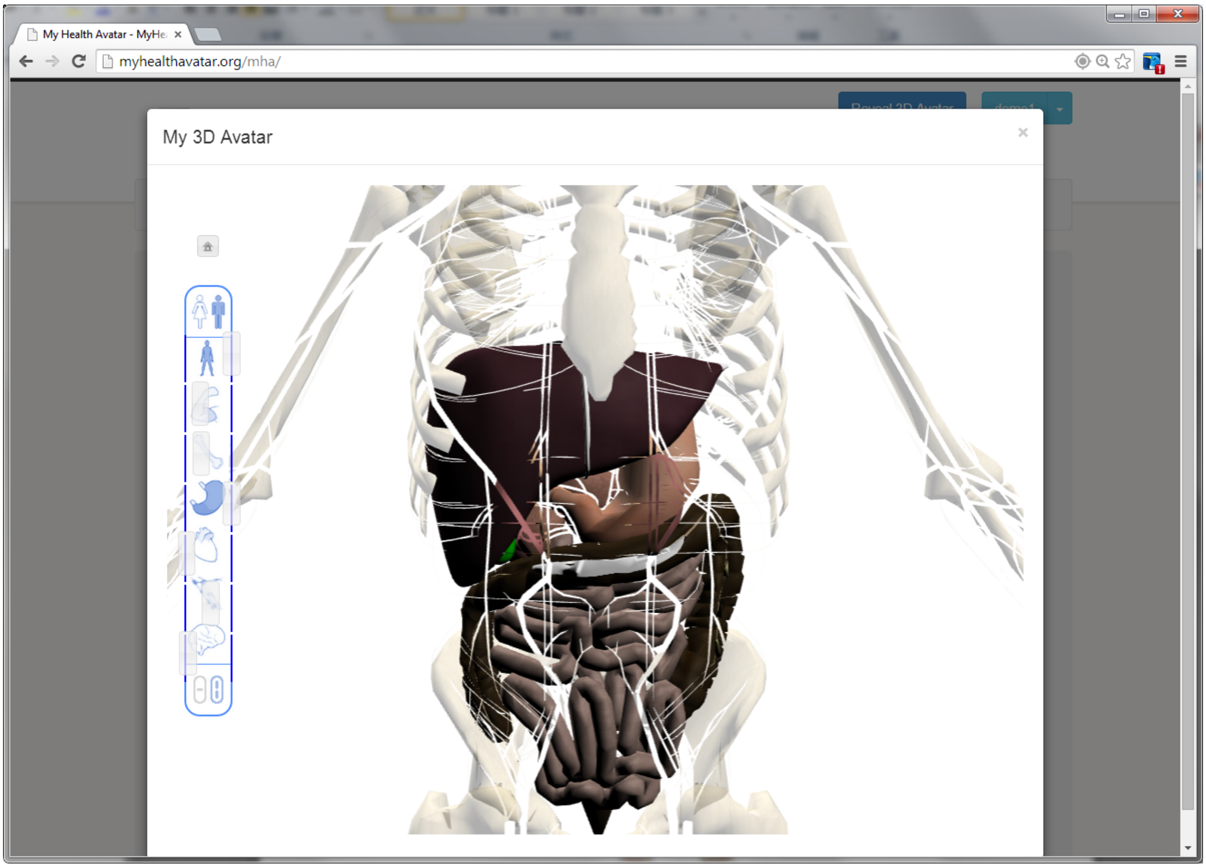

Figure 2. A semi-transparent skeleton and non-transparent internal organs after user adjusts the slider bar.

2.3.5 Picking and Automatic Focusing

For visual analytics of the digital avatar, displaying body-related health data along with the corresponding body parts is an intuitive and efficient way for both patients and doctors. For example, the colour of a dysfunctional organ can be shown in a colour different from its normal colour. When the patient or doctor picks the part, related information automatically displays. Consequently, for interactive health information exploration, visualisation and analysis, picking is a key step in the human-computer loop.

In our implementation, picking is implemented by the raycaster provided by three.js. All the models in the display are stored in a list which is used by the raycaster together with the position of the user clicks. The raycaster calculates all ray intersections and the first one is the foremost one being picked. A popup message box will show up, which can be used to display health related information; in the current implementation it shows the name of the picked part. Figure 5 shows the stomach is picked by the user.

When the user picks one part, the 3d avatar suite will automatically zoom to the selected part to fit to and centers on the canvas. This is implemented by a javascript tweening library to animate the camera position and direction to intepolated positions and directions between the origin and target.

Figure 3 shows a highlighted and auto-focused stomach after user picking.

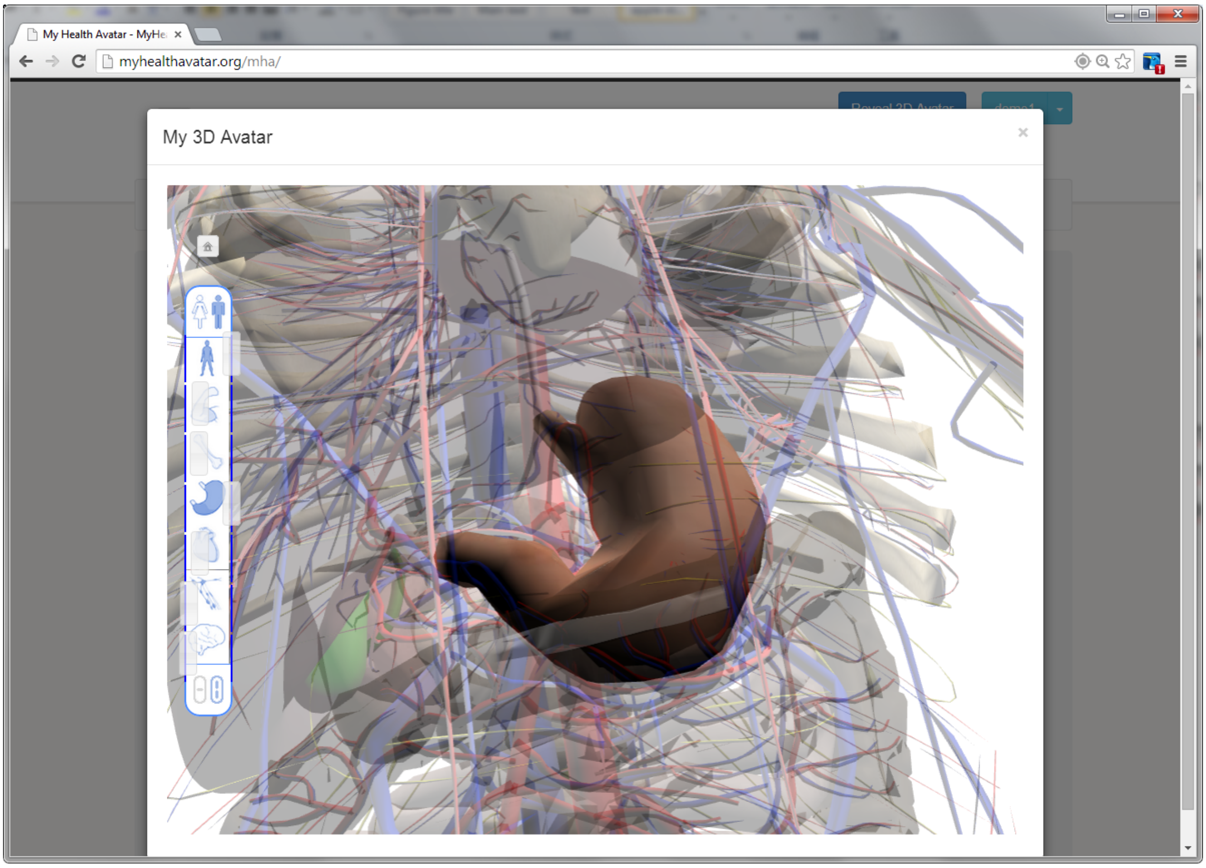

Figure 3. Highlighted and auto-focused stomach after user picking

2.3.6 Memory Management

Most current web browsers do not automatically release all the model memory allocated during reloading and page refreshing, which makes browser memory consumption accumulate and leads to slow-down, freeze or even crash of the browsers.

Memory management in our work comprises two parts: WebGL memory release of geometry, textures and materials by calling the dispose() method in Three.js; model release by setting the relevant model variables to null while page reloading. Test results of an automatically refreshing page in Chrome show that memory consumption accumulation has been removed.

2.4 Evaluation

The web-based 3d avatar suite has been integrated into MyHealthAvatar platform and is available to all the MyHealthAvatar partners at http://www.myhealthavatar.org. Among the partners, there are domain experts on a variety of subjects, including not only computer science but also medical science, biology, law, etc.

The 3d avatar suite is tested and evaluated in mainstream browsers, including Chrome, Firefox, Safari, Opera, Internet Explorer (version 11). It can run smoothly in all these browsers with WebGL support enabled.

Most partners responded that they can learn how to interact with the 3d avatar without a special learning phase. Trackball interactions with a mouse are intuitive. Picking and interactions with the toolbar are also natural. The 3d avatar gives an overall interactive anatomy model for users and can be accessed via the internet.

Responses from our medical partners also pointed out the model itself is not very accurate from medical viewpoints and more information can be displayed when the user click on certain organs.

In addition, despite that it has been integrated into MyHealthAvatar platform, the full integration with the health and lifestyle information can only be achieved when the full platform is ready.

3 Conclusion

Our web-based 3d avatar shows that, with the support of WebGL, the Internet has become a feasible platform to visualise large 3D models such as the complete human anatomy. Embracing this powerful 3D rendering capability in web pages may provide opportunities for a variety of web-based applications, especially the avatar-centered health applications, to present more effective and intuitive visualisation of their data to their end users across the Internet.

In conclusion, here are the major features of the web-based 3d avatar suite:

1. Deliverable 8.1, the web-based 3d avatar suite has been achieved.
2. We have released the avatar suite to MyHealthAvatar platform at http://myhealthavatar.org/
3. Responses from the evaluation show that the 3d avatar suite provides functions needed to interactively display 3d human body anatomy and is easy to access, use and interact with.
4. We are continuing our effort in the integration of the 3d avatar with the whole platform.

4 References

[1] OpenGL, http://www.opengl.org/

[2] WebGL, http://www.khronos.org/webgl/, http://en.wikipedia.org/wiki/WebGL

[3] Three.js, http://threejs.org/

[4] SceneJS, http://scenejs.org/

[5] Zygote Body, http://www.zygotebody.com/

[6] BioDigital Human, https://www.biodigitalhuman.com/

[7] JQuery, http://jquery.com/

[8] Autodesk 3D Studio Max, http://www.autodesk.com/products/autodesk-3ds-max