UDC 611:004.94

Vasyl Martsenyuk[1], *doctor of technical sciences, prof.,*
Iryna Tsikorska[2]

AN APPROACH FOR DEVELOPING COMPUTER MODELS OF ANATOMICAL STRUCTURES

[1] University of Bielsko-Biala, Willowa Str. 2, 43-309, Bielsko-Biala, Poland,
[2] Ternopil National Medical University, Voli Square 1, 46001, Ternopil, Ukraine

Introduction

The best “model” for studying human anatomy has always been a dead physical body [1; 2]. Since then in most cases all parts of the body are correctly positioned, you can touch all elements, all structures (soft, hard, smooth, rough, dry and wet) as really as possible. Since ancient times, due to curiosity, people have examined the wounds and organs of their dead relatives. However, today access to a dead physical body is tightly regulated by legal, financial and social restrictions. In addition, even in the presence of a dead physical body, problems arise with the demonstration of the thalus cavity and fascia. And finally, the problems of donation of dead physical bodies, their storage, the use of chemical harmful substances, the proper burial of cadaveric samples create certain difficulties for some educational institutions.

To solve such issues, professional anatomical models are used. Ancient and modern models are very different due to the details and materials used. Once upon a time, ordinary wood and ivory, papier-mâché or more detailed plaster models were used to represent anatomical structures, then there were realistic wax models Susini, Towne, Ziegler and finally modern professional plastic models [3–5]. Anatomical models also differ in terms of application — the use by doctors to advise female patients, the skills of surgeons, and, of course, the study of anatomy [6; 7].

Physical anatomical models have inherent limitations in their use, storage, and maintenance. The cost of professional models can reach thousands of dollars, depending on material, size, details, accuracy and interactivity. In addition, with large contingents of students, such physical anatomical models can become damaged and become unusable over time. Modern plastic samples have shown their effectiveness in training [15; 16]. However, most of the above limitations are also inherent in plastic models.

The **objective** of the work is to offer an approach which allows to develop 3D digital models and corresponding interactive applications for medical education on the basis of licensed software.

Materials and Methods

In addition to physical models, modern digital 3D-visualizations of structures and samples have been developed using medical images and digital 3D-modeling. Digital 3D-models of anatomical structures are available on a computer using mobile applications or stand-alone workstations (for example, Anatomage, Touch of Life). The presence of these materials in the anatomical laboratory and on training computers can reduce the need for physical anatomical models or even printed atlases and can even help teaching by bringing training materials to the anatomical table. However, the cost of such a specialized interactive 3D image system is significant (about $ 100,000 for the Anatomage table and additional devices), and the need for constant professional support should be taken into account. Using iPad tablets and mobile devices reduces cost, but requires the development of special applications that also affect learning.

The advantage of digital 3D-models is that they can be constantly manipulated to demonstrate changes in the structure or sample, taking into account age and stage of development, the role of surgery or intervention, the mechanism of functioning. Digital 3D-models can represent the morphogenesis of the heart or inner ear and clarify the embryonic structures and tissues that are important during the successful stages of embryogenesis. In addition, with the help of digital 3D-models, it is possible to trace the focus of cirrhosis of the liver or the progression of Alzheimer’s disease from the ini-

---

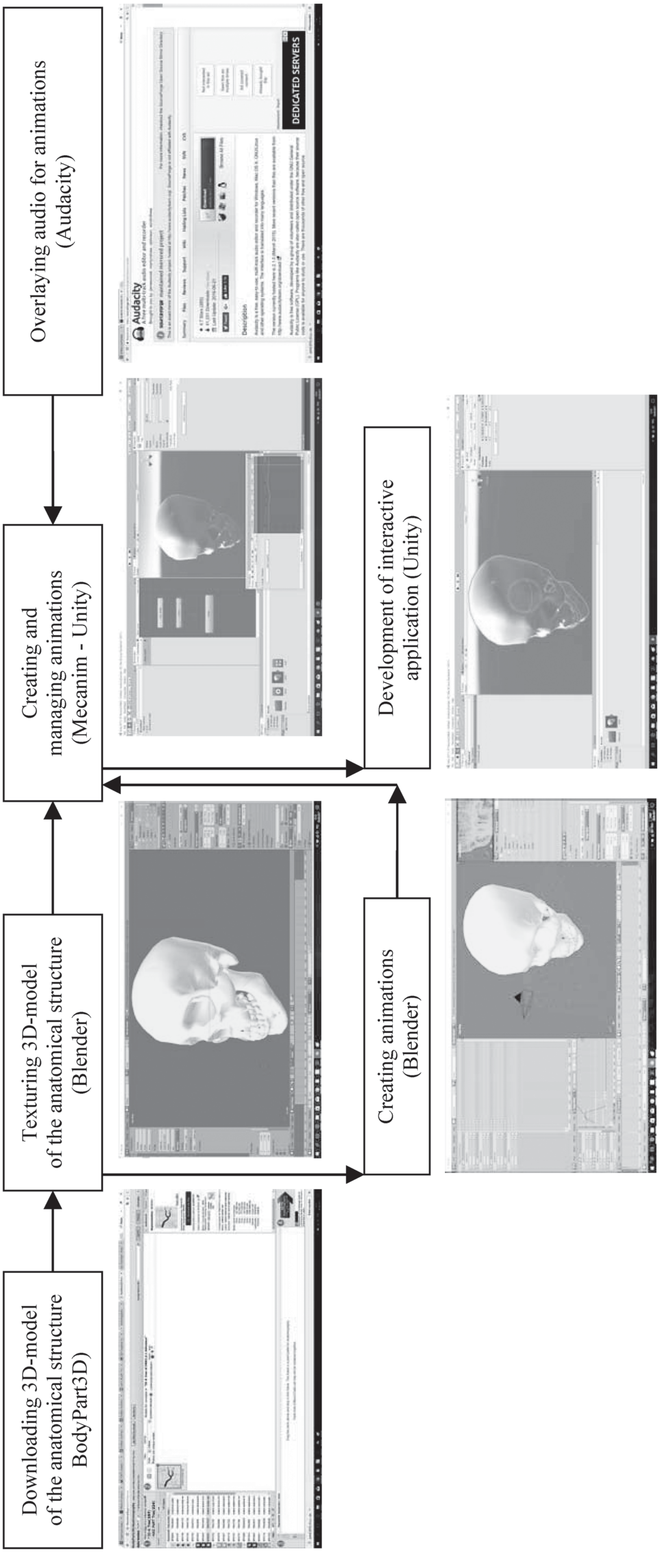


Fig. 1. Flowchart of the development of interactive training tutorials on the basis of 3D-models

tial formation to the stages of complete tissue destruction. And finally, these models can demonstrate the mechanics of the functioning of joints in different positions and under loads, or the physiology of the functioning of the heart muscle. On the contrary, a number of physical models are required in order to obtain any of these representations.

The obvious advantage of digital anatomical 3D-models is that a teacher or educational institution can create them in a relatively short time (requires an experienced biomedical illustrator). The cost may be relatively small (a workstation with open-source software). However, more efficient software can be expensive ($ 1,000 – $ 5,000 / year), even with an academic license. The price of a license depends on the method of use (personal or institutional), negotiations, type of use (clinical or educational). Creating libraries or model repositories can reduce the need for using original models and reduce their cost [10–12].

Main Results

The proposed procedure for developing applications for 3D modeling of anatomical structures includes the following stages of work (Fig. 1).

Using 3D-models of anatomical structures developed in the Anatomography project. The Anatomography project was launched in 2009 at Tokyo University (the founder of the project is Professor Kousaku Okubo). The database of mesh models of anatomical structures is called BodyParts3D. The project address on the Internet is http://lifesciencedb.jp/bp3d/?lng=en. Mesh data for BodyParts3D was obtained from MRI images. The process of building models for BodyParts3D consisted of 3 stages.

1st stage. Anatomical segmentation was performed on the basis of MRI images in a special TARO format.

2nd stage. With the help of medical illustrators using 3D editing programs, missing details were added and edges were sharpened.

3rd stage. Segmentation and modification of the data was carried out in collaboration with clinicians until a conceptual similarity was achieved.

BodyParts3D mesh models are distributed as OBJ files. For version 3.0, the total data size was 127 MB (simplified mesh) and 521 MB (high quality). Furthermore, the number of anatomical structures is — 1,523. Today the current version is 5.0.

Archived data can be downloaded from http://dbarchive.biosciencedbc.jp/en/bodyparts3d/download.html. Images generated by Anatomography and grid data in BodyParts3D are licensed under a Creative Commons license. This is to ensure the widespread use and accessibility for medical education.

Note, that the anatomical structures of BodyParts3D can be integrated into a more complex model created, for example, in MakeHuman software (Fig. 2).

Import and process a 3D model in Blender. The Blender program offers a wide range of objects for creating and further editing: meshes, NURBS surfaces, Bezier curves, vector fonts (TrueType, PostScript, OpenType).

There are tools for cutting the mesh. Boolean functions for grids are implemented. Editing grids is possible using vertices, edges and faces. There are many functions for editing objects that allow you to get an object of almost any kind.

Using Python scripts, you can create new user editing tools.

Blender includes the BMesh system, which allows you to create and edit faces with a very complex structure — for example, consisting of dozens of edges.

The main actions when using the program when working with anatomical structures from the BodyParts3D database are:

— importing the model in the format of an OBJ file (File-Import-Wavefront (. obj) commands);

— representation of the object in different projections (views are obtained using the keys of the numeric keypad: to switch between orthogonal and perspective projections, use the key 5, front view — 1, right view — 3, top view — 7, bottom view — 9, rotate the view window to the specified the angle is obtained by pressing the keys 2, 8, 4, 6, opposite views are obtained when the Ctrl key is pressed);

— geometric transformations of the active object (movement — key g, revolutions — key r, scaling — key s);

— obtaining a grid representation of the model (g);

— transition to editing mode and vice versa (Tab key);

— editing in the modes of vertices, edges, faces (bottom menu keys);

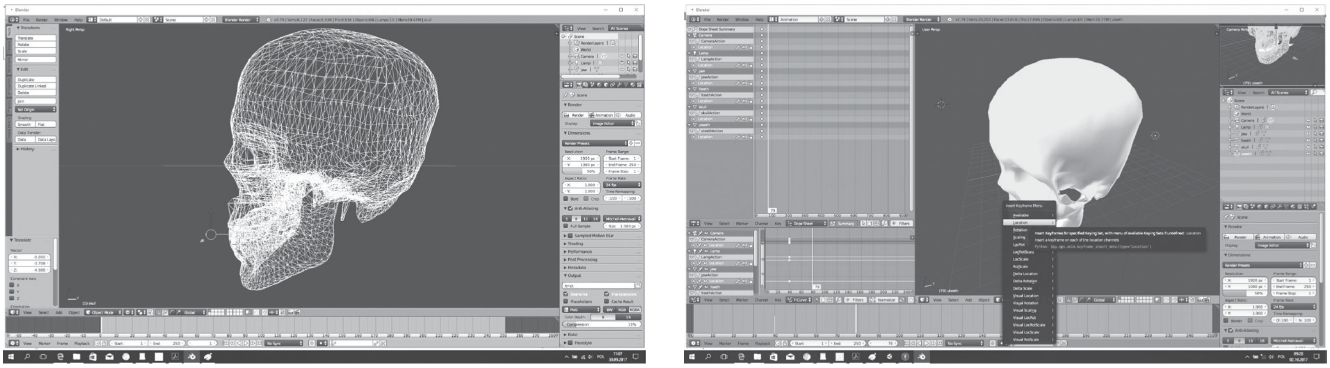
— overlaying material (created on the basis of texture) on the corresponding faces).

The animation system in Blender is based on the use of forward and reverse kinematics. It supports automatic skinning, interactive drawing of the distribution of weights directly on the object, manual editing of the skeleton processing method for each vertex, the envelopes system.

A mixer for non-linear animation with a cycle of movement along the curve is implemented. At the same time, it is possible to animate even individual vertices, which allows you to work even with very complex objects from the point of view of construction (such as anatomical structures).

The construction of animation is based on the use of a system of animation curves, the so-called IPO curves. The principle of “driven-keys” is supported, in which a change in the value of one parameter (for example, width) leads to an automatic change (proportional or not) to another (for example, height). It is possible to control parameter values using mathematical expressions (written in Python). The use of sound files and their editing for the purpose of appropriate synchronization is ensured.

Supported synchronization based on Motion capture technology.



a *b*

Fig. 2. Processing models imported from BodyParts3D in Blender:
a — mesh model; *b* — creating animation

In addition, scripts written in Python create new animated features.

Animation creation is controlled by the channel key mechanism (Location, Rotation, Scaling, and various channel combinations (LocRot, LocScale, etc.) are implemented. Based on the set keys, the model parameters change in all other frames based on automatically generated interpolation curves (animation curves) that have graphical visualization (Fig. 3). To create a Blender animation when the animation recording button is on, the frame pointer is set on the definition You can create a channel key by pressing the i key and selecting the appropriate context menu command.

Animations created in this way can then be used to create an interactive application.

Creating an interactive application based on 3D-models of anatomical structures of BodyParts3D is most expediently presented today using the Unity game engine. Unity, combining all the features of a modern 3D development environment, is used under the Proprietary license, which when used to create computer games imposes restrictions on the game budget and the number of competing players. Moreover, the Personal license with a game budget of up to $ 100,000 and the number of competing players up to 20 is free. The development of interactive training applications is in full compliance with such limitations. Interactive applications using anatomical structures can actively rely on the use of the following Unity components.

The Unity physics engine is one of the most important components for creating virtual reality effects. In order for the 3D object (in this case, the anatomical structure) to be processed by the physical engine, it must contain the Rigidbody component. In the future, this allows you to associate with this component such properties as mass (in relative units), acceleration of displacement, angular acceleration, the use of gravity, kinematics, interpolate displacements, indicate the accuracy of collision detection (collisions).

The collision detection method involves the use of the corresponding components of the physical engine, namely, BoxCollider should be used when tracking collisions of cubic objects (or close to it), Sphere Collider — spherical, Capsule Collider — as a capsule, Mesh Collider — as a grid, Terrain Collider — in the form of a complex surface shape. It should be borne in mind that the use of the last two components significantly increases the load on the GPU, which requires careful use.

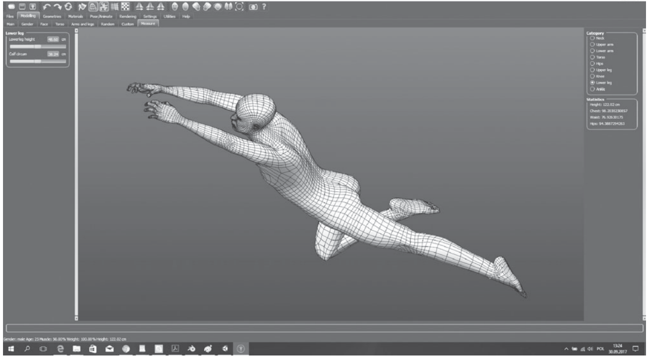


Fig. 3. Software MakeHuman allows to develop mesh models in the different positions

The next group of components of the physical engine is associated with the use of compounds of physical bodies. At the same time, such important types of joints for anatomical structures are supported: mobile (Hinge Joint), fixed (Fixed Joint), elastic joints (Spring Joint) and joints with an arbitrary configuration (Configurable Joint).

External applied force should be modeled using the Constant Force component.

Using the animation system Mecanim. Unity has the sophisticated Mecanim animation system. It provides: simplicity of the control scheme and animation settings for all Unity elements, including objects, their parts and properties; support for imported animation clips and animations created inside Unity; redirecting humanoid animations — the ability to apply animations from one character model to another; simplified process of aligning animation clips; convenient preview of animated clips, transitions and interactions between them. This allows animators to work more independently of programmers, debug animations even before applying program code; management of complex interactions between animations using a visual software tool; animation of various parts of the body with different logic; multi-level.

As a rule, a series of animations created earlier is associated with each 3D object. Control over the execution of such animations is carried out using a specially created component of the Animator Controller. Switching between animations occurs when certain events occur in the program. For example, you can start an anatomical structure wrapping animation by pressing the r key. Even if there is only one animation clip, it should still be placed in the animator controller component for use in the Game Object.

The controller controls the states of different animations and the transitions between them by the so-called state machine, which can be imagined as a block diagram or a simple program written in a visual programming language inside Unity. The structure of the Animator Controller is created and presented and changed in the Animator Window.

Each Animator Controller defines an input state called Entry, and an output state Exit. Next, each clip associated with the object (Animation components) has its own state (correspondingly, the block in the block diagram in the block diagram of the Animator window). Transitions between states (blocks, animations) are created using the Mecanim visual toolkit in a flowchart using the Make transition context menu.

Each transition between animations is represented as an object, which is configured in the object inspector window. Moreover, to control the transitions, parameters specially created in the Parameters tab of the Animator window are used. Such parameters can be of four types: Float, Int, Bool, Trigger. Moreover, when using a Boolean parameter of the Trigger type, its value switches each time when using it for a transition.

Having created the parameters and using them to determine the conditions for making transitions between animations in the object inspector for transitions (Conditions panel), the created animation management mechanism can be used in C # language code. For example, this is how the keystrokes of the keyboard are processed to call the animations r — wrapping, s — scaling:

```
using UnityEngine;
using System. Collections;
public class Skull3D : MonoBehaviour {

Animator animator;

// Use this for initialization
void Start () {
   animator = GetComponent<Animator>();
}
// Update is called once per frame
void Update () {
   if(Input. GetKeyUp(KeyCode. r))
         {
         animator. SetInteger(“state”,1); // rotation
   }
   if(Input. GetKeyUp(KeyCode. s))
      {
         animator. SetInteger(“state”,2); // scaling
       }
    }
}
```

Animation can also be triggered when colliding with another specific object (collision):

```
void OnCollisionEnter(Collision col) {
   if (col. gameObject. CompareTag("wall"))
   {
         animator. SetTrigger("broken_down");
   }
}
```

a specific wall tag. By creating and using block diagrams of the Mecanim system in this way, very complex mechanisms for managing animations associated with 3D-models are achieved (Fig. 4).

Create Interactivity with Triggers

Under triggers, as a rule, they mean very general entities (even functions, program code, etc. are possible) that help to control program execution through some kind of interaction (interactivity). In the case of Unity 3D graphics, the concept of triggers is used for the property of components of the Collider type of the physical engine. Setting this property for components allows you to further process events associated with contacts with this object: when you enter (collide) with this object, when you are in this object, when you exit the object. Such actions for ensuring interactivity can be processed in the appropriate methods:

```
void OnTriggerEnter(Collider collider)
{
// actions when entering trigger
}
void OnTriggerStay(Collider collider)
{
// actions when staying within trigger
}
void OnTriggerExit(Collider collider)
{
// actions when waying out trigger
}
```

Conclusions. Thus, we investigate the problems that arise when teaching medical students on physical models and proposes an approach for developing interactive software that uses 3D computer graphics in medical education.

We point out that this development procedure is focused on the use of pre-prepared models of anatomical structures, created for example on the basis of MRI images and presented in different repositories — both commercial and freely distributed (for example, BodyParts3D). Further, the process consists in using free software — which, according to the authors, would contribute to the wide development of interactive training programs in medical universities.

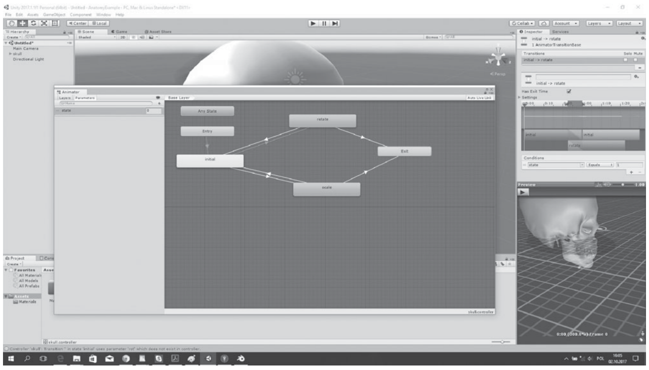


Fig. 4. Creating and managing animations with help of the system Mecanim

Table 1

The Software Offered for Usage when Developing Computer Models of Anatomical Structures

Activity	Software used	Company	License
Using 3D-models of anatomical structures developed in the Anatomography project	BodyParts3D	Anatomography source	Free open-
Obtaining human network models with the possibility of modification	MakeHuman	The MakeHuman team	AGPL
Importing and processing 3D-models, creating animations	Blender	Blender Foundation	GNU GPL
Creating an interactive application with controlled animations	Unity Technologies	Unity Technologies	Proprietary

We believe that the use of modern licensed development tools (Table 1), which have gained popularity today, primarily in the development of entertainment software products (computer games) has several advantages when using computer graphics for development of medical education applications: firstly, focus on processing complex grid models, 3D visualization, creating effects, interactive animation control; secondly, the presence of a sufficient number of specialist programmers who own these technologies; thirdly, sufficient debugging of these software environments (when developing Blender, Unity projects, a lot of high-class programmers were involved, projects were supported for quite a long time, today a large number of software versions have been developed to improve and fix errors).

Ключові слова: комп’ютерна графіка, мультимедіа, медична освіта, BodyParts3D, Blender, Unity, 3D-моделювання.

REFERENCES

1. Amy Manson, Matthieu Poyade, and Paul Rea. A recommended workflow methodology in the creation of an educational and training application incorporating a digital reconstruction of the cerebral ventricular system and cerebrospinal fluid circulation to aid anatomical understanding. *BMC Med Imaging.* 2015; 15: 44. doi: 10.1186/s12880-015-0088-6

2. John R. Fredieu & Jennifer Kerbo & Mark Herron & Ryan Klatte & Malcolm Cooke. Anatomical Models: a Digital Revolution. *Med. Sci. Educ.* 2015; 25: 183-194. DOI: 10.1007/s40670-015-0115-9

3. Marković D., Marković Ž. B. Development of anatomical models — chronology. *Acta Med Medianae.* 2010; 49: 56-62.

4. Riva A., Conti G., Solinas P., Loy F. The evolution of anatomical illustration and wax modelling in Italy from the 16th to early 19th centuries. *J Anat.* 2010; 216: 209-222.

5. Hopwood N. The art of medicine. *Lancet.* 2008; 372: 1946-1947.

6. Russell K. F. Ivory anatomical manikins. *Med Hist.* 1972; 16: 131-42.

7. Maerker A. Florentine anatomical models and the challenge of medical authority in late-eighteenth-century Vienna. *Stud Hist Phil Biol Biomed Sci.* 2012; 43: 730-740.

8. Douglass C., Glover R. Plastination: preservation technology enhances biology teaching. *Am Biol Teach (Natl Assoc Biol Teach).* 2003; 65: 503.

9. Latorre R. M., García-Sanz M. P., Moreno M., Hernández F., Gil F., Lypez O. et al. How useful is plastination in learning anatomy? *Vet Med Educ.* 2007; 34: 172-176.

10. Becker B. W. Digital learning object repositories. *Behav Soc Sci Libr.* 2010; 29: 86-88.

11. Kerfoot E., Lamata P., Niederer S., Hose R., Spaan J., Smith N. Share and enjoy: anatomical models database—generating and sharing cardiovascular model data using web services. *Med Biol Eng Comput.* 2013; 51: 1181-1190.

12. Richards G., McGreal R., Hatala M., Friesen N. The evolution of learning object repository technologies: portals for on-line objects for learning. *J Dist Educ.* 2002; 17: 67-79.

Submitted 23.12.2019
Reviewer L. S. Godlevsky, MD, prof.,
date of review 24.12.2019

УДК 611:004.94

Василь Марценюк, Ірина Цикорська

ПІДХІД ДО РОЗРОБКИ КОМП’ЮТЕРНИХ МОДЕЛЕЙ АНАТОМІЧНИХ СТРУКТУР

Робота присвячена проблемі проектування 3D-моделей анатомічних структур для медичної освіти. Пропонується підхід до розробки програмного забезпечення з можливістю інтерактивного проектування анатомічних структур. Використовуване програмне забезпечення вільно розповсюджується і може застосовуватися в медичних університетах.

Ключові слова: комп’ютерна графіка, мультимедіа, медична освіта, BodyParts3D, Blender, Unity, 3D-моделювання.

UDC 611:004.94

Vasyl Martsenyuk, Iryna Tsikorska

AN APPROACH FOR DEVELOPING COMPUTER MODELS OF ANATOMICAL STRUCTURES

The work is devoted to the problem of designing 3D-models of anatomical structures for medical education. We offer an approach for software development with possibility of interactive design of anatomic structures. The software used is freely distributed and may be used in medical universities.

Key words: computer graphics, multimedia, medical education, BodyParts3D, Blender, Unity, 3D-modelling.