

© TheAuthor (s) 2013;

This article is published with open access at Licensee Open Journal Systems of Radom University in Radom, Poland

**Open Access**

This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

This is an open access article licensed under the terms of the Creative Commons Attribution Non Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non commercial use, distribution and reproduction in any medium, provided the work is properly cited.

Conflict of interest: None declared. Received: 05.01.2013. Revised: 10.02.2013. Accepted: 16.02.2013.

**UDC 519.685.1**

## **USE OF OPEN PACKAGE BLENDER TO CREATE THE TRAINING COMPLEX VIRTUAL ENVIRONMENT**

**<sup>1</sup>Dolgova E.V., <sup>1</sup>Kurushin D.S., <sup>1</sup>Faizrakhmanov R.A., <sup>1</sup>Shilov V.S.,  
<sup>1</sup>Chomenok V.R., <sup>2</sup>Prokhorov G.V., <sup>2</sup>Gozhenko S.A.**

**<sup>1</sup>Perm National Research Polytechnic University**

**<sup>2</sup>Ukrainian Scientific Research Institute of Transport Medicine, Odessa, Ukraine**

### **Abstract**

The article examines the creation of a virtual reality for usage in training complex, which should provide high-quality rendering of 3D-scenes, simulated physical effects such as gravity, friction, and high quality lighting effects. The proposed solution has no licensing restrictions and it can be used on variety of platforms.

A development of soft-hardware solutions related to automation of operative decision-making on the man-triggered objects requires from a developer to pay more attention to the question of reducing the number of erroneous decisions of a duty executive.

While managing complicated processes such as the management of larger man-triggered vehicles (MV), like locomotives, aircraft and vessels, there may be

"abnormal" situation in which the correctness of decision resulting on the security of many people. [1]

Special part of the research for solutions to this problem is the development of automated intelligent systems, expert systems, training systems, the development of one of the components of which is addressed in this research.

The present stage of development of automated learning systems is characterized by the use of advanced information technology and communications, the increasing number of intelligent components, using powerful graphics systems, personal computers, and some others [2, 3, 4, 5].

Russian and Ukrainian simulator-building skills shows high effectiveness of training programs, training and supervision of knowledge, pledged by applying three-dimensional computer graphics models, methods and algorithms for simulating the working conditions of the operator [7].

At the present time it is well known enough simulators for civil and military applications that are based on these technologies and allows you to simulate the action teams and individual operators in real time with a high degree of certainty.

### **1. A visualization of virtual environment**

One of the most important problems of modern digital simulators is the synthesis of photo-realistic images. This task can be divided into physically based modeling and physically accurate visualization [7, 8, 9].

At the stage of physically based simulation model of the object is determined by the components and the scene, and the equations characterizing the developments taking place is written. At the stage of visualization, these equations are solved, and the image is constructed. In this section we consider the existing development in the field of interactive visualization systems that solve only the second part of the problem - physically accurate visualization.

#### ***1.1 The visualization system Blender Game Engine***

Blender - is a free package for creating three-dimensional computer graphics [10], which includes tools for modeling, animation, rendering, post processing, as well as the creation of interactive games.

Blender Game Engine (BGE) - subsystem Blender, allows you to create interactive applications. BGE supports a variety of geometric primitives, including polygonal models, a system of rapid prototyping mode subdivision surface (SubSurf), Bezier curves, surfaces NURBS, metaballs (metasphere), sculptural modeling, and vector fonts.

Blender's animation tools such as inverse kinematics, skeletal animation, mesh deformation, animation keyframe, nonlinear animation, editing vertex weights, limiter, soft-body dynamics (including collision detection of objects in interaction), the dynamics of solids based on physics engine Bullet, hair system on the basis of the particles and the particle system with the support of collisions.

To automate a number of tasks modeling and describing the behavior of systems used language Python. It allows to describe the same logic of interactive applications. In BGE object (which is the entity that interacts with the outside world) and its data (some form / function of the object) are shared. The ratio of the object data is the ratio  $m:n$ , and dynamically linked, allowing you to use some rapid prototyping processes that are unique to Blender.

Blender Game Engine allows you to simulate gravity (as a permanent force in the direction "down", the force of attraction / repulsion between objects, the friction force (linear and torsional), atmospheric effects. Such as fog, smoke, snow. All this adds realism to the objects, and also allows you to "feel" the depth of the scene.

### ***1.2 Application BGE for modeling physical processes***

In the real world objects made of different materials, and naturally have different properties. BGE to simulate the behavior of physical objects uses a number of features and technologies.

Type of the selected object (eg, a cube) is defined by Object Type menu and can take the following values: Occluder, No collision, Static, Dynamic, Rigid body, Soft body.

Objects are managed in the BGE via so-called sensors, controllers and actuators. Sensors allow them to perceive the object various external "stimuli", such as keystroke, mouse movement, touch another object, etc. The controllers are mostly designed to bind a certain way of sensors and actuators. They also allow you to call scripts in Python. Activators can perform various actions on the object or the other.

The whole logic is determined by the visual editing of connections between sensors, controllers and actuators (Fig. 1).

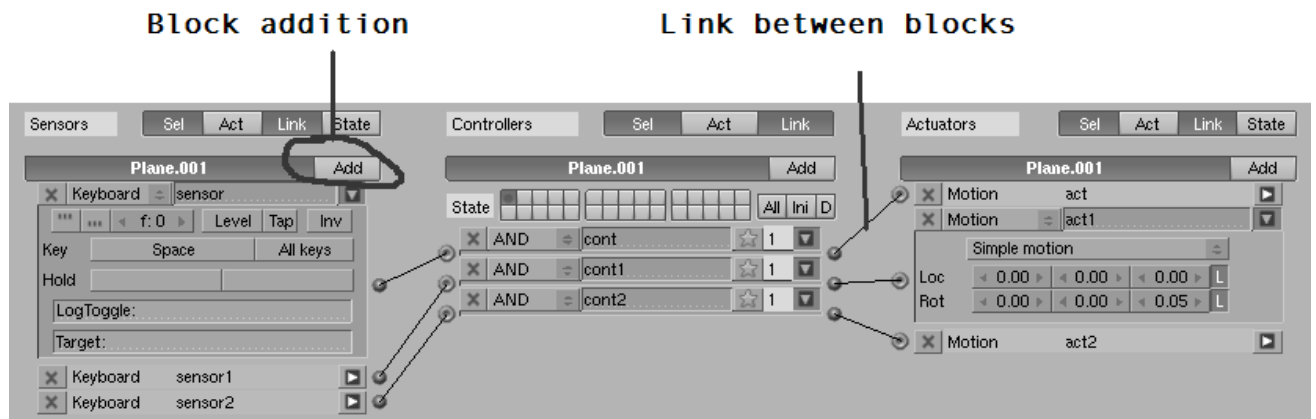


Fig. 1: Logic panel

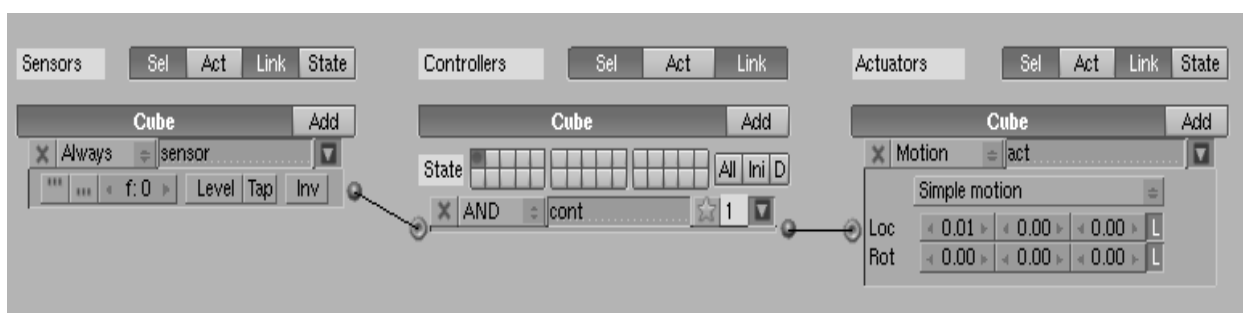


Fig. 2: Such a sensor, controller and actuator force to move forward

Thus, by combining the properties of objects and the logic of their behavior (see Fig. 2) can be quite accurately imitate the behavior of real-world objects. BGE works in real time with a minimum response time of 1/60 seconds.

### 1.3 Usage of Python

If features built-in logic is not enough, for example you want to receive data from external sources, such as files, databases or the Internet, you can use the controller type Script can call a subroutine in the language Python [6]. An example of the use of Python for the animation of the object is shown in Listing 1

Listing 1 - Spring pendulum animation

```
from numpy import *

g = array([0,0,9.8])
m = 1
k = 1

def objectByName(name):
    return GameLogic.getCurrentScene().objects['OB' + name]

def Fg():
    return m * g

def Fk(x):
    return -k * array(x)

def Ft(v):
    return v / 10

try:
    o = objectByName('A')
    v = array([o['v_x'],o['v_y'],o['v_z']])
    x = array(o.position)
```

```

F = Fk(x) - Fg() - Ft(v)
a = F/m
v = v + a/10
x = x + v/10
o.position = [x[0],x[1],x[2]]
o['v_x'],o['v_y'],o['v_z'] = v[0],v[1],v[2]

except KeyError:
    pass

```

As you can see, on the subject "A" gravity and elastic response of suspension, all calculations are made in vector form, is transparent to the programmer. The operators of addition, multiplication, subtraction and division are covered in the module numpy.

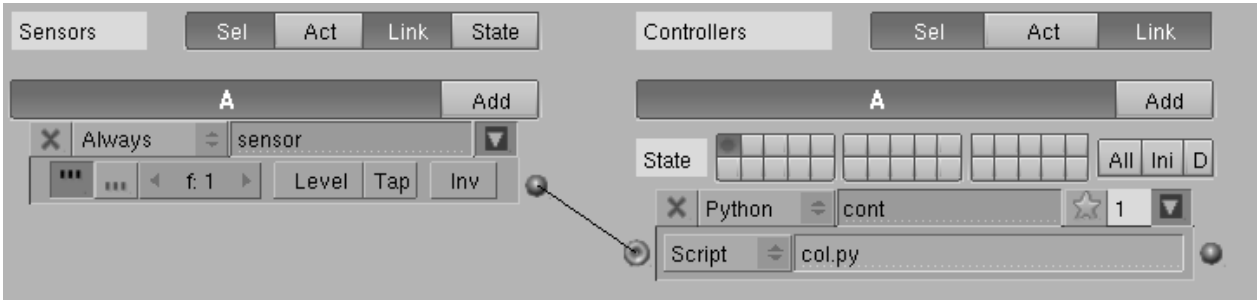


Fig. 3: Description of logic using Python

Description of virtual world's Logic is shown in Fig. 3. As well seen, the activator is not used, since coordinates of the objects are defined directly from Python.

**2. Using Blender to create a virtual space training complex**

In the process of creating a training complex (TC) was set a goal : to create a three-dimensional model of the chassis lifting machine (CLM), where later will be installed a specialized equipment.

In the TC as an imaging system is used game engine Blender Game Engine (BGE), it is the built-in software for three-dimensional modeling Blender.

Since working with a large number of objects and complex physical interactions requires significant computing resources, from the point of view of three-dimensional modeling it is required to create a model containing the fewest number of polygons, but a detailed model should be sufficient to render realistic training scenarios.

### ***2.1 Purpose of the three-dimensional model***

Three-dimensional model to be created is designed for realistic visualization of operators with GMP as part of all training scenarios, training complex.

The three-dimensional models has some functional requirements that should be provided. The three-dimensional model of GMP should implement the following functions:

- realistic visualization of the chassis;
- the possibility of conflict free installation of various models of the equipment;
- low polygon count (approximately 20-30 thousand with all the equipment);
- model should be fully compatible with the environment;
- conflict-free model should interact with the environment and be animated.

The following process of the creation of three-dimensional models is the same for all three-dimensional models that were created for the TC.

Based on the characteristics of TC, was highlighted a number of substantive requirements that must be satisfied by generated three-dimensional models. For example, for the environment were separated the following requirements:

- hills and lowlands must be present in the landscape;
- in the center of the world should be relatively flat area (slope of not more than 3 °);
- in the outside world must be some elements of the natural environment (trees, rocks, etc.).

There are a number of requirements to the size and scale of the objects that will be present on the stage. Since the standard edit box Blender has a dimension of 60 × 60 standard units, it was decided to equate one standard unit of Blender to 10 meters.

To reduce the time required, it was decided not to create all the models from scratch, and use as the basis of some of them have already completed models, such as model crane arm with lifting devices.

However, the pattern was not suitable for use in TC, because they are made in the format and 3D StudioMAX and possess an excess of polygons. Models have been converted from the format \*.max to format \*.obj and imported into Blender for the subsequent design, based on these models required. Schemes have been used, as well as photographs of models required to improve adherence to the size and scale models of parts [10].

The purpose of the TC is the demonstration of the operation of some machinery and equipment not only visually, but also must be valid from the point of view of physics . This was reached by the three-dimensional model of the manipulator crane and lifting devices was attached to a mathematical model, which simulates the movement of the pistons, as well as swing lifting devices [11]. For correct operation of mathematical models with three-dimensional models, the necessary changes were made and produced by the interaction of configuration objects.

The lighting of the world was made by method of three points. This is a very simple classical scheme to achieve a soft lighting in the scene. Sun type lamps were used. Rays of such light are parallel to each other, like sun ones. However, these lamps can generate only a shadow of ray tracing. Unfortunately, the BGE does not support this method [10]. Therefore, on the stage with shadows Spot type lamps are used.

## **Conclusion**

Implementing of environment Blender and Blender Game Engine helps us :

- to realize quite complicated physical processes occurring in the system of lifting machines - hydraulic - boom - suspension - load - the operator;
- to provide high-quality rendering of objects in real time;



- to reduce development costs through economies on license fees and the high-level programming block in the BGE.

## Literature

1. Коровин К.В. Автоматизированная система ограничения массы поднимаемого груза стреловых грузоподъемных кранов: автореф. дис. канд. техн. наук. – Челябинск, 2009.

2. Орлов А.Н. Общая динамическая модель грузоподъемных кранов // Оптимизация параметров строительных и дорожных машин. – Ярославль: Изд-во Яросл. политех. ин-та, 1992. – С. 13–20.

3. Орлов А.Н., Синицын В.Ю. К расчету частот собственных колебаний грейфера на канатном подвесе // Исследование оптимальных металлоконструкций и деталей подъемно-транспортных машин. Вып. 6. Саратов, 1992. – С. 28–32.

4. Орлов А.Н. Поперечные колебания грейфера с учетом ослабления подъемных канатов // Динамика, прочность и надежность технологических машин. – Труды СПбГТУ, № 478. – 1998. – С. 48–51.

5. Д.С. Анализ и исследование динамического состояния стрелового крана в переходных режимах при подъеме груза / Матер. межвуз. науч. конф. Ч. III // Санкт-Петербургский гос. техн. ун-т, 2001. – С.26–27.

6. Сузи Р.А. Язык программирования Python: Учебное пособие. – М.: ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. – 328 с.

7. Баканов А.С., Обознов А.А. Эргономика пользовательского интерфейса: от проектирования к моделированию человеко-компьютерного взаимодействия. – М.: Изд-во «Институт психологии РАН», 2011. – 175 с.

8. Головач В.В. Дизайн пользовательского интерфейса. – 2009. – С. 47–59.

9. Калинин И.А., Самылкина Н.Н. Требования к современным тестирующим системам. – 2006. – С. 352–356.

10. Обзор Blender Game Engine. [Электронный ресурс]. – URL: [http://wiki.blender.org/index.php/Doc:RU/2.4/Manual/Game\\_Engine](http://wiki.blender.org/index.php/Doc:RU/2.4/Manual/Game_Engine) (дата обращения: 20.04.2012).

11. Изучаем Blender Game Engine. [Электронный ресурс]. – URL: [http://b3d.mezon.ru/index.php/Изучаем\\_Blender\\_Game\\_Engine](http://b3d.mezon.ru/index.php/Изучаем_Blender_Game_Engine) (дата обращения: 20.04.2012).

12. СанПин 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, ПЭВМ и организации работы».

The work was supported by the Russian Ministry of Education (contract № 13.G25.31.0093) as part of the RF Government Decree № 218 "On measures of state support for the development of cooperation of Russian higher education institutions and organizations implementing complex projects on high-tech production"