

A novel intelligent object-oriented three-dimensional simulation system for physics experimentation

Object-oriented
simulation
system

241

Rania M. Ghoniem

*Faculty of Specific Education, Department of Computer, Mansoura University,
Mansoura, Egypt and*

*Department of Information Technology,
College of Computer and Information Sciences,
Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia*

H.A. Abas

*Faculty of Specific Education, Department of Psychological and Educational Science,
Mansoura University, Mansoura, Egypt, and*

H.A. Bdair

*Faculty of Specific Education, Department of Computer, Mansoura University,
Mansoura, Egypt*

Received 3 June 2018
Revised 21 September 2018
Accepted 29 October 2018

Abstract

Despite the fact that there being a large literature on simulation, there is as yet no generic paradigm or architecture to develop a three-dimensional (3-D) simulator which depends on autonomous intelligent objects. This has motivated us to introduce a 3-D simulation system based on intelligent objects for Physics Experimentation. We formulated the system's components as an object-orientation model. So, the entities in every experiment's work cell are modeled by characterizing their properties and functions into classes and objects of the system hierarchy. Intelligent objects are realized by developing a knowledge base (KB) that captures a set of rules/algorithms that operate on 3-D objects. Rules fall into two categories: action and property rules. In the simulation layer, the student is allowed, by using the virtual system, to stroll throughout the Physics laboratory in light of a walking model. Student gets to a simulation region to do an experiment through the detection of mathematical collision. From software engineering perspective, the proposed system facilitates the Physics experiment through making the specification of its applicable parts more modular and reusable. Moreover, a major pedagogical objective is achieved by permitting the student tuning parameters, fixing

© Rania M. Ghoniem, H.A. Abas and H.A. Bdair. Published in *Applied Computing and Informatics*. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) license. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this license may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

Publishers note: The publisher wishes to inform readers that the article "A novel intelligent object-oriented three-dimensional simulation system for Physics experimentation" was originally published by the previous publisher of *Applied Computing and Informatics* and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Ghoniem, R.M., Abas, H.A., Bdair, H.A. (2020), "A novel intelligent object-oriented three-dimensional simulation system for Physics experimentation", *Applied Computing and Informatics*. Vol. 16 No. 1/2, pp. 241-258. The original publication date for this paper was 01/11/2018.



Applied Computing and
Informatics
Vol. 16 No. 1/2, 2020
pp. 241-258
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI [10.1016/j.aci.2018.10.003](https://doi.org/10.1016/j.aci.2018.10.003)

component of a device then visualizing outputs. This provides student well interpretation by viewing how distinct parameters affect the outcomes of the experiment. With the objective of student performance measuring, we utilized an exploratory group relying upon pre- and post-testing. The application results demonstrate that the simulator contributes positively to student performance in regard to practical Physics.

Keywords Three-dimensional (3-D) simulation, Object-orientation, Intelligent objects, Rule-based reasoning

Paper type Original Article

1. Introduction

3-D simulation is a challenging innovation to initiate conditions allowing accurate management of the state experienced by knowledgeable users. The simulator is a computer-based program through which individual behaves in a virtual world building up synthetic locations throughout the user interaction layer which imitates senses [1]. Currently, an increasing emphasis is given on virtual environments-based education [2–4]. This brings capability to make use of the constructivist 3-D simulators that can widen the classroom through modern penetrations into diverse realities. Through illustrating the world as actual, the simulator offer the participants the chance to try to discover different alternatives with absence of risks to health, bear the training costs, and eliminate time-consuming that could occur if the user performs the “real thing” [5].

The “behavior” of a simulation system ordinarily includes all responses of the environment to the user physical intervention. This, in turn, refers to the procedures triggered by user, during the time that for instance s/he grasps, and then drops the object. In general, this behavior is encoded procedurally. In addition, the underlying behavior (i.e., movement pattern, etc.) is not defined explicitly other than throughout a few variables set inside equations or scripts [6,7]. In a like manner, the utilization of a specified Artificial Intelligence (AI)-based layer for defining the objects behavior within the environment of a 3-D simulation carries out the thought of the intelligent object-orientation modeling.

The challenge to develop the 3-D simulation systems is the general motivation for this paper. Thus, an object-oriented paradigm is intended to give direction to outline 3-D simulators from perspective of software engineering. Accordingly, the object-orientation approach [8–10] is used to develop a novel Physics 3-D simulation system putting into consideration the ultimate objective to model every experiment’s objects through making their applicable parts extra modular and with a reusable specification. Afterwards, a specified AI-based layer is built for conveying several benefits for the object-oriented 3-D simulation system: supporting redefinition of the alternative and nonrealistic behavior from first standards, allowing the fast modeling, and experimentation besides.

2. Related work

Multiple simulators in various branches have emerged in importance in the last decade [11–24]; an overview of certain studies is shown in Table 1. On the one hand, object oriented modeling has come to be an integral element of the development procedure of software architectures owing to progresses in Computer-Aided-Design. As shown in Table 2, a small number of studies have focused on the object-oriented paradigm for dynamic building systems [25–29].

Moreover, in the published literature and to the best of our belief almost all of preceding researches on simulation have concentrated on simulation visualization instead of the paradigm of building the system dynamically. Thus, there are no other works presenting object-orientation paradigm to direct outlining of 3-D simulation systems with intelligent objects, from the view of software engineering. On the contrary, a key goal of this work is to introduce a 3-D simulation system based intelligent objects for doing Physics experiments thru making their applicable parts more modular and with a reusable specification.

References	Year	Domain application	Contribution
[11]	2017	Geophysical laboratory system	A virtual geophysical laboratory system was presented which relies on Viustools, C#, and database technology. The system assists the end users to grasp and master the laboratory apparatus and experimental processes
[12]	2017	Structural Analysis (SA)	Pedagogy was introduced for teaching SA course that amalgamates augmented reality with 3-D visualizations. The aim was to upgrade the textbook or worksheets-based content of SA throughout visualizing the discrete structural members that use augmented reality conjointly with 3-D interactive models to show the impact of different loading conditions on the structure
[13]	2017	Networking	A simulation model was suggested for the VHDL-based Network Interface Card. This gave chances to simulate the network-to-memory data path at a network node and generate a Value Change Dump (VCD)
[14]	2017	Particle physics collider experiments	The authors highlighted the effect of simulating detector on particle physics collider tests
[15]	2017	Multi-physics simulation of nuclear reactor core	The reactor thermal-hydraulic core analysis is got by the CUPID which has been aimed to analyze transient two-phase flows in components of nuclear reactor
[16]	2016	Metrology activities	As a means to do real hands-on experiments, the authors designed a virtual laboratory framework. It was combined into Moodle (Learning Management System) as a novel action to set up a connection with any other Moodle learning activities (i.e. courses, tests, etc.) and to make certain student tracking
[17]	2016	Mechatronic systems	Virtual reality (VR) Lab for Mechatronic Systems (VLMS) was presented that includes two parts - hydraulics module and robotics module and. Dynamics of each specific lab apparatus was mathematically modeled and then simulated
[18, 19]	2012 & 2013	Programming	In [18], a visualization tool was introduced to improve student's acquisition of object-oriented programming concepts. In [19], two methods have been embraced for the instructional design: one relied upon visualizing animations, while the second upon instructions without effective use of animations
[20]	2011	Control systems	A virtual lab was proposed to simulate the control systems used for technological plants. Therefore, the subsequent plants have been simulated: tube heat exchanger, continuous stirred-tank reactor, besides fluid-stockpiling tank system
[21]	2010	Physics	A tool was presented for simulating the nature dynamics of electromagnetism
[22, 23]	2009 & 2013	Mechanical engineering	A game-based laboratory system was presented as a part of the course, entitled, "Mechanisms & Machine Dynamics" that gives the basics of dynamics and kinematics and connects them to linkages, camera systems, gear trains, couplings, vibrations, along with belt and train drives
[24]	2009	Science	Simulating the Boyle-Mariott's law, heat transportation, AC and DC electrical circuits, Gay-Lussac's law, and Charles's law

Table 1.
An overview of some
simulators presented in
the literature.

Table 2.

An overview of some works that adopted the object-oriented paradigm for dynamically building systems.

References	Year	Domain application	Contribution
[25]	2018	Modeling cardiopulmonary interactions	An object-oriented model to investigate cardiopulmonary interactions in humans
[26]	2017	Large-scale Smart Grids	The equation-based/object-oriented modeling method and the modelling language of Modelica to control, design, and manage large-scale Smart Grids
[27]	2015	eHealth	A functional architecture and context-aware model intended to steer an intelligent, adaptive message routing, and personalized policy. The relevance of the approach was illustrated by a use-case in the eHealth field
[28]	2014	Modelling of nuclear reactor dynamics	An object-oriented model and simulation tool for the investigation of the Advanced Lead-cooled Fast Reactor European Demonstrator plant dynamics
[29]	2013	Modelling of physiological systems	An Object-oriented model and simulation of the closed-loop cardiovascular system through using SIMSCAPE

Subsequent contributions of the paper advance the state-of-the-art on the 3-D simulation as follows:

- Presenting a prototype to develop 3-D simulation systems with intelligent objects.
- With the aim to outline the software architecture of the 3-D simulators, an object-oriented paradigm is introduced. Consequently, a 3-D object can be reused in numerous parts of the system. The object may represent a magnet, bulb, avatar, battery, motor, or anything may encounter the user thru the experiment. The model is developed through incorporation of 3-D objects which denote the component parts of the virtual Physics experiment. A KB determines the behavior of 3-D objects through two categories of object rules, that is: action, and property rules to represent knowledge as well as make reasoning. Consequently, the 3-D object may act based upon the user commands to show its unpredictable state.

3. System prototype

In this section, we introduce the prototype of the suggested system which is illustrated in [Figure 1](#). Secondary school pupils can engage with the practical Physics laboratory utilizing the usage of the synthetic character. It is essential to create a real behaviour if you want to avoid boredom at some point of the gaining knowledge during learning. S/he can choose the synthetic character and at once use it to stroll thru the virtual environment (see [Figure 2](#)) primarily based upon walking model, in addition to collision detection with the other 3-D objects. The background of walking and collision detection is detailed in the previous works [\[30,31\]](#).

3.1 Walking model

Animation of human walking is sort of a key framing example. That is as a result of the traits of walking which is a non-stop transition amongst a number of statuses. The synthetic character changes all through walking among the subsequent statuses:

- Status 1: Place left foot on the ground and forwardly move body.
- Status 2: Place right foot on the ground and move body.

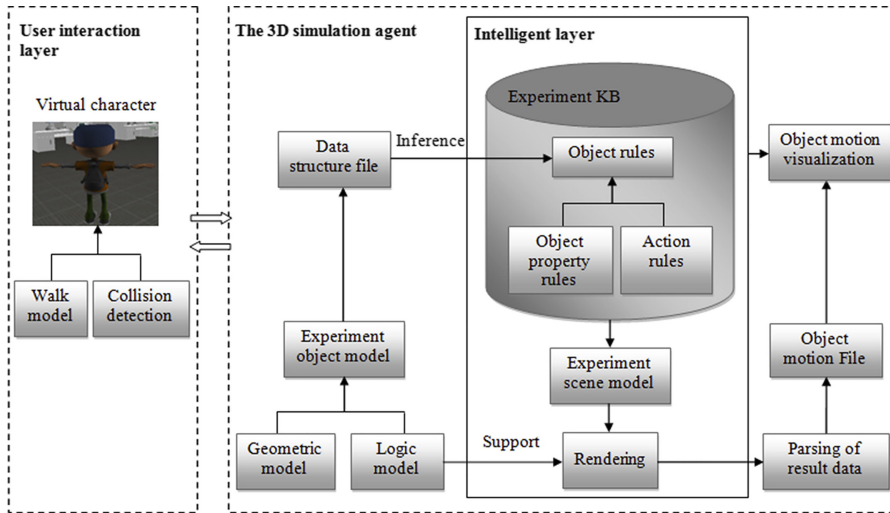


Figure 1. Prototyping of the system.

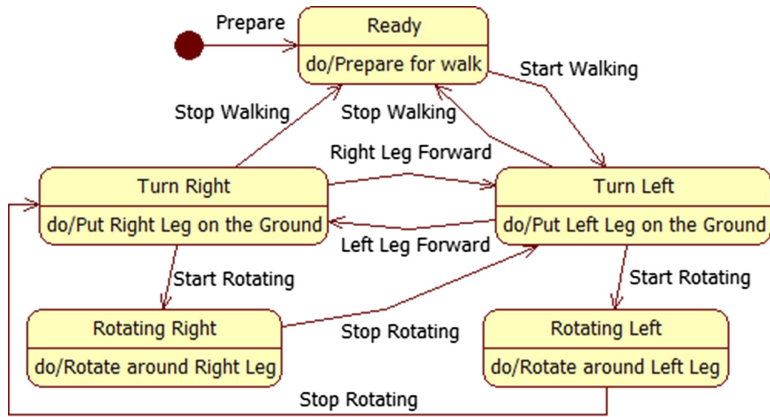


Figure 2. Main screen of the Physics 3-D simulation system.

- Status 3: Transform from current location to *Status 1* to start walking.
- Status 4: Transform from any present status to any position in order to stop walking.

However, the virtual character may additionally need for turning its course throughout strolling. Accordingly, the rotation of body is a motion incorporating two different statuses, as demonstrated in Figure 3. If the rotation procedure starts thru any arbitrary timing, there might be a state wherein no foot touches the ground of lab, and the motion will appear artificial. For that reason, the rotation must coordinate the character's stroll to attain superior visual outcomes. If the status of the object is "in walking", and must turn his body, the virtual

Figure 3.
State diagram of
walking model.



scene is expected to wait until certainly one leg touches the ground and body rises on. Later, the character turns his body around this leg; as a consequence the procedure is easily simulated and appears in a real manner. The state diagram that clarifies the walking model is depicted in Figure 3.

3.2 Collision detection and reaction

It is a critical step to compute not only accurate *collision detection* but also a precise response for a virtual character's body mesh. This paper uses a method founded on bounding primitives that surround the character's body segments and does the collision detection tests with them and accordingly achieves lower computational time. The cylinders are the primitives selected in this work, attributable to their symmetric geometry. On the contrary to bounding boxes, the cylinders are properly tailored enclosing the character's body:

- Computing of bounding boxes: The system examines every mesh vertex then defines the minimum and maximum values taken by x , y and z . Therefore, two resulting vectors can be get (C_{max} and C_{min}) which determine the corners of bounding box. The size I and center T of the box are computed using:

$$I = C_{max} - C_{min} \tag{1}$$

$$T = \frac{C_{max} + C_{min}}{2} \tag{2}$$

- Cylinder type: A cylinder can be 3-D aligned on x , y or z , the axis on which an aligned cylinder achieves the maximal of the vector of bounding box size.
- Center, radius in addition to height: The cylinder's center is in line with the bounding box's center. Moreover, the cylinder's height is estimating the axis size which it aligns on; the cylinder's diameter indicates the average between the size values for the two other axes. If x -axis is an axis on which an aligned cylinder and the bounding box's size vector is $[Ux, Uy, Uz]$, then the length will be Gx , the diameter, and radius are computed using:

$$D = \frac{Uy + Uz}{2} \tag{3}$$

$$R = \frac{Uy + Uz}{4} \quad (4)$$

- On this step, checking out collision with various 3-D objects within the virtual scene is acted: Assume a cylinder W which is x-aligned and check whether or not a point is inside it. Suppose that $[Nx, Ny, Nz]$ refers a center of the cylinder, D represent its radius, in addition H is its length. Suppose also $[Ox, Oy, Oz]$ refers to the point. As a consequence, the indicated point is in the cylinder only if:

$$Nx - \frac{1}{2} \leq Ox \leq Nx + \frac{1}{2}, \text{ and} \quad (5)$$

$$\sqrt{(Ny - Ox)^2 + (Nz - Oz)^2} \leq D \quad (6)$$

where the distance of point from the cylinder's axis is less than D .

3.3 The data structure file

A logical object model which is responsible for documenting methods that operate on the data entities is encompassed in this file; how the methods are assembled within interfaces and classes; besides how the distinct object types can interact amongst themselves. We can divide the logical object model into:

- *Geometric model*: The 3-D object geometry information (i. e., *Armature*) comprises the knowledge required for motion planning, i.e. velocity, shape, scale, texture, or size. Each 3-D object owns additionally position and orientation in the 3-D space (denoted by $[Rx, Ry, Rz]$, and $[Sx, Sy, Sz]$, respectively).
- *Logic model*: Defines the entities which will be rendered by the system, along with the methods that operate on the entities.

3.4 KB of 3-D experiment

The rules/algorithms are documented in the KB to operate on the entities and define the 3-D object behavior based on two categories of object rules, in particular: action and property rules, that tend to represent knowledge and make reasoning. The rule is in particular constituted of a name as well as a group of properties and their values besides. Whilst triggering the rule, the group of properties it comprises are immediately implemented. The rule properties determine the entire actions executed by the rule. In [Section 4](#), the object rules defined in this work are described in more detail.

3.5 Rendering process

Rendering of graphic [\[32\]](#) is referred to as the procedure of converting a 3-D scene into a 2-D image frame, with the objective of human vision. This procedure incorporates multiple procedures i.e., command processing via the front-end processor, termed "FEP"; model-view transformation for vertices and vertex normals; using vertex normals; shading clamping outcomes, and vertex shading; plane clipping; primitive assembly; frustum clipping; projection transform; rasterization (scan-conversion); back-face culling; fragment operations; and pixel shading/fogging.

3.6 Visualizing object motion

Subsequent to data processing, the path file of objects' movement begins to be available. Based upon that file, the simulator creates a model-load-list for loading the geometric object

models. Accordingly, the arrangement of models loading corresponds to the arrangement of key points of the output. Therefore, the system drives the 3-D objects' movement models in the simulation region thru loading the data file of object movement. With the goal of object visualization, the data of geometrical movement of each moving object is outputted after rendering; the data is in a PLT form (a format used to render complicated 2D and 3-D scenes).

4. Modeling the Physics simulator using proposed object-oriented paradigm

The procedure of outlining any object-orientated layout model is the transformation of the phenomena from actual world domain to entities of conceptual domain representing actual world. The mapping is done through extraction and theoretical generalization. Entities of conceptual domain are mapped into the object model classes. On this basis, three basic functional blocks are usually essential to develop a 3-D simulation system:

- User interaction (UI) layer: Receives the user input thru input devices in addition to the output visualization to show the change done within the simulation area.
- Information processing (IP) layer: It is a subsystem that targets to control the information which the user input involves. It also enacts the operations as well as tasks which the system's orders particularize.
- The layer of 3-D simulation engines: Such engines apprehend the processes of object redrawing. They also are not significant for developers because they are commonly integrated in the platform of development, or even the run-time environment.

In this context, we describe our object-oriented paradigm intended to develop the Physics simulator as in the following steps.

4.1 Object-oriented model

The object-orientated design of [Figure 4](#) is proposed to introduce a general model to develop the software architecture for Physics 3-D simulation systems. The given model offers mechanisms as modularity, composition, inheritance, encapsulation, data abstraction, and suits the bottom-up design concurrently with rapid implementation of modular and hierarchical architectures of 3-D simulators. The model is firstly outlined thru the refinement of classes and packages of the design that belong to the two top-level layers (i.e.: UI and IP layers). The process of refinement models the entities in the work cell of every experiment via the abstraction of their properties and methods inside classes or objects of the system hierarchy. For each class and package, a number of attributes and methods are defined to indicate a specific functionality in the system. [Table 3](#) demonstrates classes, attributes as well as methods of the user interaction in addition to information processing layers demonstrated in [Figure 4](#). As an example, the object class retains the 3-D object's geometric properties such as shape, size, color or position that are absolutely customized. Furthermore, for characterizing the type of light source, the 3-D lighting class encapsulates two properties (i.e., *Lighting_Kind* and *Position*).

[Figure 5](#) depicts an instance of a portion of the system modeled utilizing the proposed object-orientated paradigm. The primary experiment, specifically, "Multi-induction", mimics electromagnetic induction between two coils whilst the plug-in of one into the second. With respect to design, the model involves two primary classes i.e., *Electromagnetic* and *Lamp_Device*, primarily based upon the reality that electromagnetic induction generates a force thru an electric conductor which is as variant magnetic fields. The model displays the generalization notion via arrow lines to reveal estimating the mutual features from the referred subclasses, and amalgamation of them into a general superclass, namely,

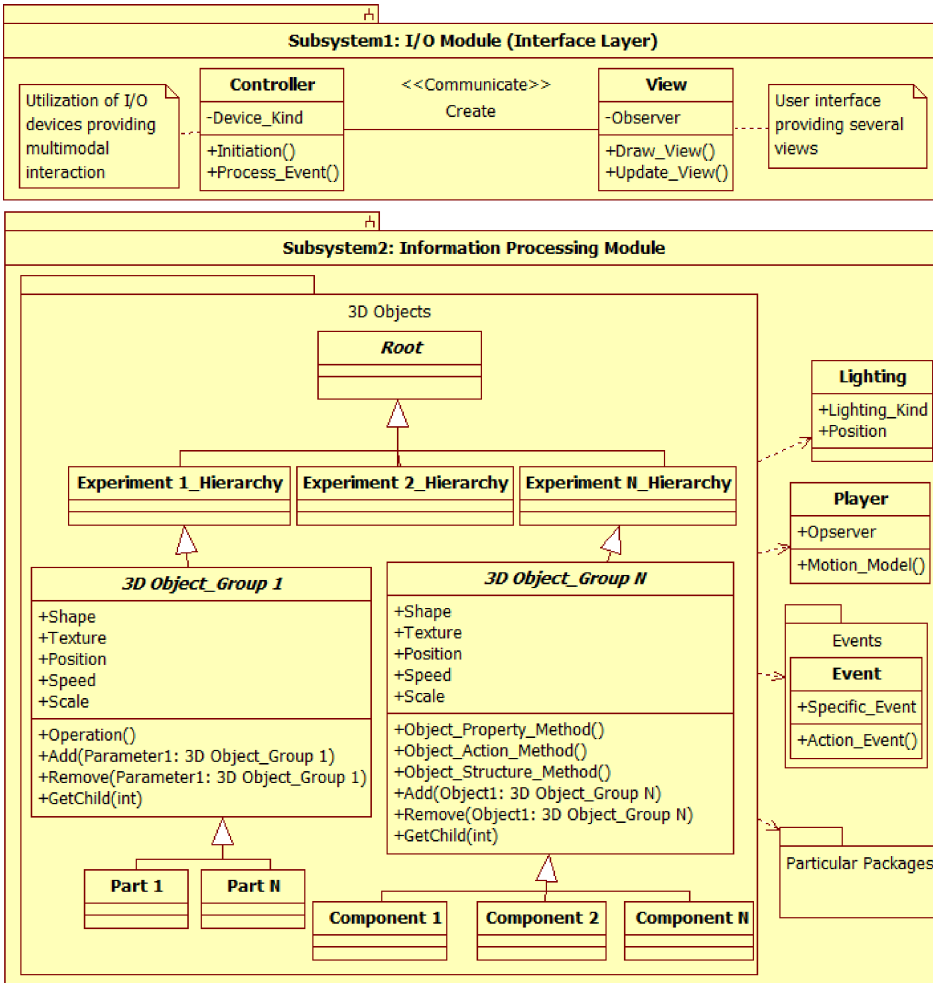


Figure 4. Object-orientation model to develop software architecture for the 3-D simulators with autonomous objects.

Object_Group. Furthermore, the diamond-shaped arrow constitutes the composition notion that represents a format of aggregation indicates the whole object (i.e., *Electromagnetic*) is in charge of disposing its components (i.e., *Cylinder1*, *Cylinder2*, *Bulb*, and *Coil_of_Copper*). The properties of the class denote the protected or private variables for doing modularity. Moreover, the second experiment, namely, “Generator”, mimics the mechanism of action of the Generator and in what manner to convert it into a motor. Therefore, the model indicates the subsequent primary components of the device: *Split Cylinder*, *two Slip Rings*, *Armature*, *DC-Motor*, and so on.

4.2 Characterizing autonomous 3-D objects behavior

To control the behavior of each experiment’s 3-D objects, we used two categories of reasoning based on object rules:

Layer	Package	Class	Properties	Methods	Description
UI	Subsystem. UI	Controller	Device_Kind	Initiation () Process_Event ()	Receiving the input via input units and processing of the events generated by them. Then, the obtained information goes thru IP layer, thus the system begins to be conscious of what to do whilst a new event happens.
UI	Subsystem. UI	View	Observer	Draw_View () Update_View()	Draws or updates the views via the data generated by the model.
IP	3-D Objects	Root (object group of 3-D scene)	Shape, size, color, or position	Add(Object 1: 3-D Object_Group N) Remove(Object 1: 3-D Object_Group N) Stop () Play () Lighting ()	Involves the 3-D object hierarchy for processing the 3-D model attained in the file of data structure.
IP	-	Lighting	Lighting_Kind Position	Lighting ()	At the least one illumination point emits flashes of lightning over the 3-D scene or specific 3-D objects.
IP	-	Player	Observer	Motion_Model ()	A player is incorporated to an observer which has a specific movement model to control the animation of the virtual characters and the 3-D objects within the scene.
IP	Events	Event	Specific_Event	Action_Event ()	Information or events obtained from input devices (as an example, collision detection) are handled within the simulation scene.

Table 3.
Classes, properties as well as methods of the UI and IP layers.

- Property rules-based reasoning: These rules are triggered on condition that the alias/ name of a specified property is indicated in the command of the user. It entails some of actions that may be acted at the properties of object. A command instance is: 'Set the resistor value to high'. For that reason, 'resistor' is the object, 'value' shows the property, 'set' indicates the action, and 'low' refers to the given value. [Algorithm 1](#) illustrates an instance of reasoning intended to adjust the electric circuit behavior based on property rules, where student can change the resistance between any two existing points on the circuit and the total resistance for the electrical devices which exist inbetween those pair of points, to see how those parameters have an effect on the electric potential difference among the two points. [Figure 6](#) depicts the circuit simulation visualization.
- Action rules-based reasoning: Each rule includes a set of actions to be performed on 3-D objects and decides what objects will do employing Methods. An instance is

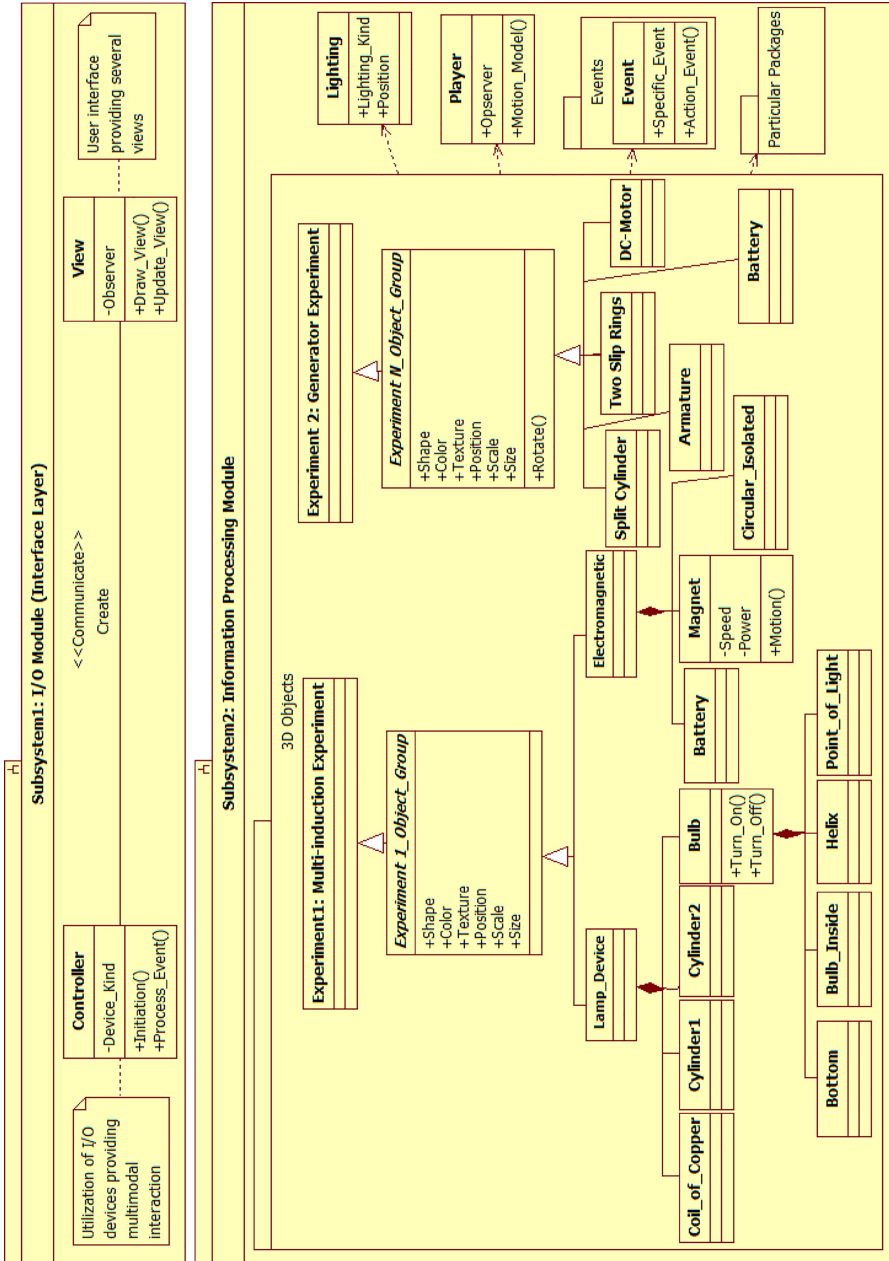


Figure 5. Object-orientation model of a portion of the system.

Algorithm 1.
Control the electric circuit behaviour using the property rules-based reasoning.

Input: Current between two points on a circuit I , and resistor R .

Output: Simulating the electric potential difference ΔV between two points on the circuit.

1. Compute ΔV between any added points to the circuit, where $\Delta V = I \cdot R$;
2. **IF** the input resistor value R is high
3. **AND**
4. The input current value I is low
5. **THEN**
6. | Decrease the *light* of the 3-D object “Lamp” using the method “OnTriggerStay()”;
7. | Reduce the movement speed of the pointer of the 3-D object “Galvanometer” using the method “animate()”;
8. **ELSE**
9. | Increase the *light* of the 3-D object “Lamp” using the method “OnTriggerExit()”;
10. | Momentarily direct the pointer of the 3-D object “Galvanometer” using the method “animate()”;
11. **END IF**
12. Vec3 X = new Vec3(0,1,0); // x-axis
13. float velocity = 8; // low value of velocity
14. float dt; // dt indicates the time change Δt
15. void animate() { // Rotation method
16. | Object.Rotate(X, velocity * dt);}
17. void OnTriggerExit() { // Method to turn off the lamp
18. | lamp.enabled = false;}

provided in [Algorithm 2](#) to control the Generator mechanism of action if the student starts to fix components such as: DC-Motor, Two Slip Rings, or Split Cylinder. Moreover, creating of the sine wave plot representing the relation among induced current C as well as rotation angle θ . Consequently, the Generator mechanism of action and the way to convert it into a motor are visually simulated as provided in [Figure 7](#).

Algorithm 2.
Control the Generator mechanism of action using the action rules.

Input: Adding/removing component parts from the device of Generator

Output: (1) Simulating the Generator mechanism of action and in what manner to transform it into a motor.
(2) Creating plot of sine wave which demonstrates the relation between C and θ .

1. **IF** the user fixes a *Split Cylinder*
2. **THEN**
3. | Replace the *Two Slip Rings* object with the *Split Cylinder* employing the twoBrush() method.
4. | Rotate the *Split Cylinder* and *Armature* mechanically in magnetic fields.
5. | Calculate and create plot of a current sine wave changeable from 0 till $+R_{max}$.
6. **END IF**
7. **IF** the end user fixes a *DC-Motor*
8. **THEN**
9. | Substitute the *Battery* object with a *Galvanometer*.
10. | Add a *Cylinder* object employing the Motor2Brush() method.
11. | Rotate the *Slip Ring1*, *Slip Ring2*, and *Armature* in magnetic fields.
12. **END IF**
13. **IF** the user fixes *Two Slip Rings*
14. **THEN**
15. | Put in *Two Slip Rings* object utilizing the HalfBrush() method.
16. | Mechanically rotate the *Two Slip Rings* and *Armature* in magnetic field.
17. | Calculate and create plot of a current sine wave changeable from R_{max} till $-R_{max}$.
18. **END IF**

5. Results of application

This section expounds the application of the proposed system to evaluate its effectiveness. The study was conducted in Abou El Mahasen Snagab High School, Mit Ghamr, Dakahlia Governorate, Egypt in the second semester of 2016/2017 academic year. In an exploratory group, the pre/post application has been used.

5.1 Application of 3-D experiments

As Figure 8 indicates, the sequence diagram of interactions demonstrates how student interacts with the virtual experiment. S/he is free to stroll thru the virtual Physics laboratory after logging into the system. Due to collision with the objects of the virtual environment, student can reach the simulation region to do the experiments. S/he can change the Physics parameters and view how they affect outcomes of the experiment. A number of experiments intend the student to put apparatus to visualize the effect.

Object-oriented
simulation
system

253

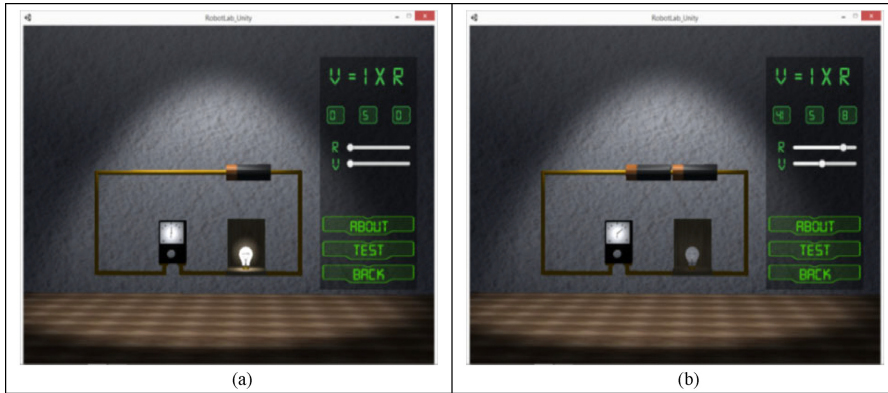


Figure 6. The electric circuit simulator (a) low resistor value, high current value, increasing lamp light, and increasing movement of the Galvanometer pointer, on the contrary in (b).

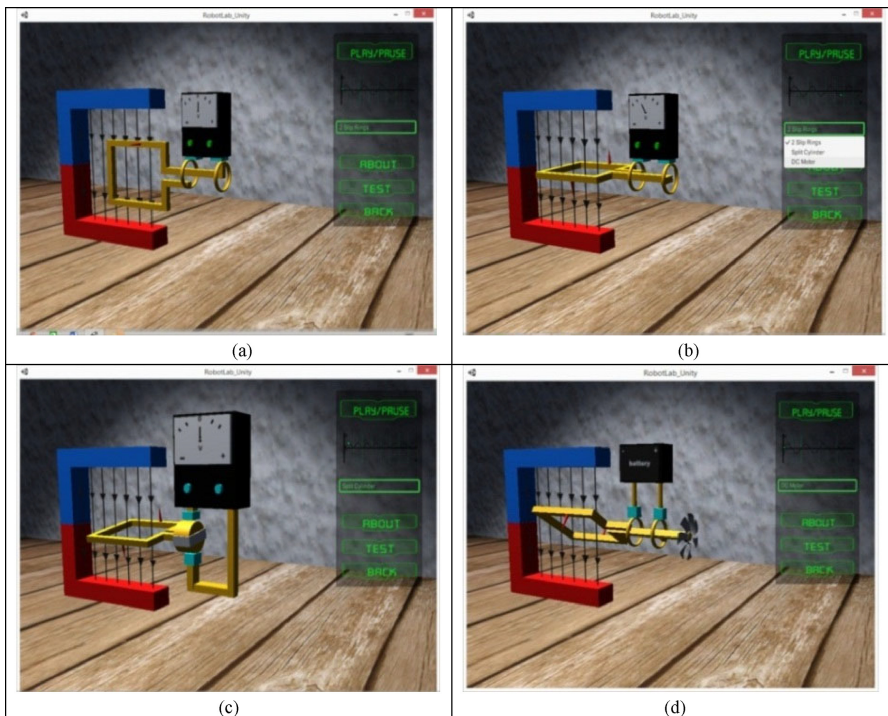


Figure 7. Generator simulator (a) Putting “Two Slip Rings” in the device, (b) Rotation of “Two Slip Rings” and “Armature” in the magnetic fields, (c) Add “Split Cylinder” to the device, and (d) Conversion of Generator into DC-Motor.

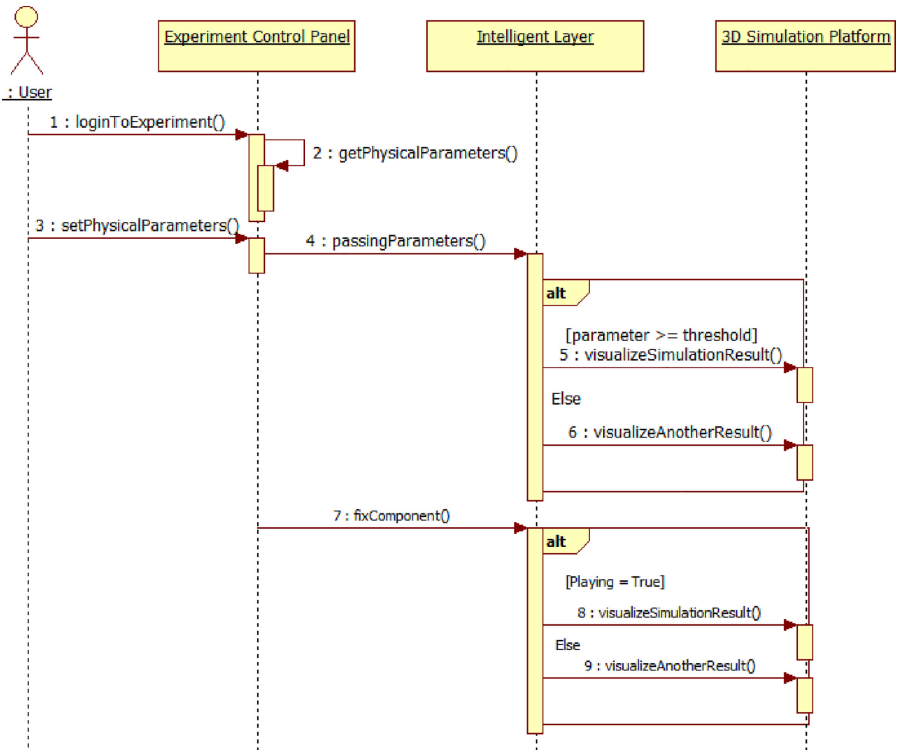


Figure 8.
Learner interactions
sequence diagram.

5.2 Performance evaluation

The performance measurement tool (Physics achievement test) implemented tribal to the students (60 students), after and before subjecting them to the 3-D simulation experiments. Figure 9 depicts the students' scores mean with respect to pre/post-application of the Physics

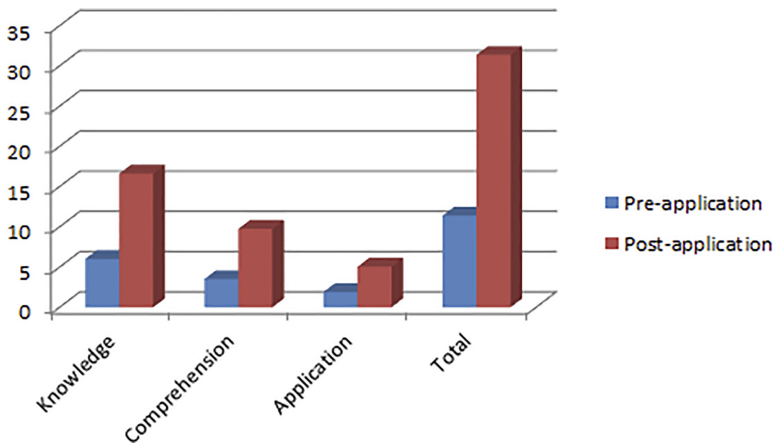


Figure 9.
Average of students'
scores in pre and post-
application.

achievement test ordered underneath the levels of Bloom's taxonomy of the behavioral goals: knowledge, comprehension, and application besides. The results reveal the significant improvement of the students' scores.

These results are due to the application of the proposed system which provides 3-D Physics laboratory enabling pupils to understand the Ohm's laws, Resistors in series/parallel, and electromagnetic induction. Actually, the pupils actually envision across Generators worlds and electric motors, which are challenging. They also imagine the "Electric Circuit" which represent a critical subject in Physics, because it is a foundation for supplementary learning about electricity. Accordingly, juniors cannot find difficulties when trying to conceive a scenario like "what a magnetic field produces as a result to a current in a straight wire or a circular loop".

We used the Wilcoxon signed-rank test to prove the hypothesis postulating significant statistical differences amongst the mean scores of experimental group with respect to pre and post-testing with its levels (knowledge, comprehension, and application) and overall score in favor to post-application. The outcomes show significant improvement ($p < 0.05$) in all testing levels. Consequently, the differences in grades between the two testing scores support the post-application. These differences demonstrate the positive contribution of the simulator to the pupil performance.

6. Conclusions

In this paper, we report a new 3-D simulation system with object-orientation for Physics experimentation. An object-orientation model is introduced with the intention to develop an intelligent simulator system. First, the entities in every experiment's work cell are modeled by characterizing their properties and functions into a set of hierarchical classes and objects. In that respect, we develop a knowledge base to document the set of rules/algorithms that operate on the 3-D objects and entities employing the inference rules. From pedagogical perspective, secondary school students can engage with the simulator utilizing the usage of the synthetic characters. It is essential to create a real behavior to prevent feeling run down from happening at some point of the gaining knowledge during *learning*. The student can virtually choose the character and at once use it to stroll thru the Physics laboratory primarily on the basis of a walking model, in addition to collision detection with any 3-D object. The system facilitates the Physics experiments through making their applicable parts extra modular and with a specification for reuse. Moreover, a prime instructive objective is achieved by permitting the student to tune parameters, fix component of a device then visualizing outputs. This provides pupil well interpretation by viewing the impact of distinct parameters on the outcomes of experiment. The system also overcomes challenges in Physics conventional training, especially in development countries, like the shortage of labs which are outfitted with devices for conducting this kind of experiments and the lack of professional surveillance. For future work, further important features will be introduced to make the system more convenient, for instance, a Natural Language Processing (NLP) module to promote the two-way communication with student via natural speech.

References

- [1] Innocenti, Virtual reality experiments in economics, SSRN Electr. J. (2015), doi: [10.2139/ssrn.2818550](https://doi.org/10.2139/ssrn.2818550).
- [2] X. Fan, X. Zhang, H. Cheng, Y. Ma, Q. He, A Virtual Experiment Platform for Mechanism Motion Cognitive Learning, Lecture Notes in Computer Science Virtual and Mixed Reality - Systems and Applications (2011) 20–29, doi: [10.1007/978-3-642-22024-1_3](https://doi.org/10.1007/978-3-642-22024-1_3).
- [3] B. Zhang, The building of network virtual laboratory for physics teaching, in: Advances in Intelligent Systems and Computing Proceedings of the 2012 International Conference of Modern Computer Science and Applications, 2013, pp. 241–246, doi: [10.1007/978-3-642-33030-8_39](https://doi.org/10.1007/978-3-642-33030-8_39).

-
- [4] X. Li, F. Ma, S. Zhong, L. Tang, Z. Han, Research on Virtual Experiment Intelligent Tutoring System Based on Multi-agent, Entertainment for Education. Digital Techniques and Systems Lecture Notes in Computer Science (2010) 100–110, doi: [10.1007/978-3-642-14533-9_11](https://doi.org/10.1007/978-3-642-14533-9_11).
- [5] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V.M. Petrović, et al., Virtual laboratories for education in science, technology, and engineering: a review, *Comput. Educ.* 95 (2016) pp. 309–327, doi: [10.1016/j.compedu.2016.02.002](https://doi.org/10.1016/j.compedu.2016.02.002).
- [6] M. Cavazza, J.-L. Lugin, S. Hartley, M.L. Renard, A. Nandi, J. Jacobson, et al., Intelligent virtual environments for virtual reality art, *Comput. Graphics* 29 (2005) 852–861, doi: [10.1016/j.cag.2005.09.002](https://doi.org/10.1016/j.cag.2005.09.002).
- [7] R. Aylett, M. Cavazza. Intelligent virtual environment: a state-of-the-art report. Eurographics 2001 conference. STAR Reports 2001.
- [8] G. Yasuda, An object-oriented multitasking control environment for multirobot system programming and execution with 3-D graphic simulation, *Int. J. Prod. Econ.* 60–61 (1999) 241–250, doi: [10.1016/s0925-5273\(98\)00181-9](https://doi.org/10.1016/s0925-5273(98)00181-9).
- [9] Y. Zhong, C. Liu, A domain-oriented end-user design environment for generating interactive 3-D virtual chemistry experiments, *Multimedia Tools Appl.* 72 (2013) 2895–2924, doi: [10.1007/s11042-013-1554-1](https://doi.org/10.1007/s11042-013-1554-1).
- [10] C.Y. Cardall, R.D. Budiardja, GenASiS Mathematics: object-oriented manifolds, operations, and solvers for large-scale physics simulations, *Comput. Phys. Commun.* 222 (2018) 384–412, doi: [10.1016/j.cpc.2017.10.004](https://doi.org/10.1016/j.cpc.2017.10.004).
- [11] S. Xiang, L.C. Wang, VGLS: A virtual geophysical laboratory system based on C# and viustools and its application for geophysical education, *Comput. Appl. Eng. Educ.* 25 (2017) 335–344, doi: [10.1002/cae.21801](https://doi.org/10.1002/cae.21801).
- [12] Y. Turkan, R. Radkowski, A. Karabulut-Ilgu, A.H. Behzadan, A. Chen, Mobile augmented reality for teaching structural analysis, *Adv. Eng. Inf.* 34 (2017) 90–100, doi: [10.1016/j.aei.2017.09.005](https://doi.org/10.1016/j.aei.2017.09.005).
- [13] G.R. Garay, A. Tchernykh, A.Y. Drozdov, S.N. Garichev, S. Nesmachnow, M. Torres-Martinez, Visualization of VHDL-based simulations as a pedagogical tool for supporting computer science education, *J. Comput. Sci.* (2017), doi: [10.1016/j.jocs.2017.04.004](https://doi.org/10.1016/j.jocs.2017.04.004).
- [14] V.D. Elvira, Impact of detector simulation in particle physics collider experiments, *Phys Reports* 695 (2017) 1–54, doi: [10.1016/j.physrep.2017.06.002](https://doi.org/10.1016/j.physrep.2017.06.002).
- [15] J.R. Lee, H.Y. Yoon, Multi-physics simulation of nuclear reactor core by coupled simulation using CUPID/MASTER, *Int. J. Heat Mass Trans.* 115 (2017) 1020–1032, doi: [10.1016/j.ijheatmasstransfer.2017.07.124](https://doi.org/10.1016/j.ijheatmasstransfer.2017.07.124).
- [16] A. Ballu, X. Yan, A. Blanchard, T. Clet, S. Mouton, H. Niandou, Virtual Metrology Laboratory for e-Learning, *Procedia CIRP* 43 (2016) 148–153, doi: [10.1016/j.procir.2016.02.110](https://doi.org/10.1016/j.procir.2016.02.110).
- [17] V.M. Petrović, B. Nikolić, K. Jovanović, V. Potkonjak, Development of Virtual Laboratory for Mechatronic Systems, *Advances in Intelligent Systems and Computing Advances in Robot Design and Intelligent Control* (2016) 622–630, doi: [10.1007/978-3-319-49058-8_68](https://doi.org/10.1007/978-3-319-49058-8_68).
- [18] I. Cetin, Visualization: a tool for enhancing students' concept images of basic object-oriented concepts, *Comput. Sci Educ.* 23 (2013) 1–23, doi: [10.1080/08993408.2012.760903](https://doi.org/10.1080/08993408.2012.760903).
- [19] J. Urquiza-Fuentes, J.Á. Velázquez-Iturbide, Toward the effective use of educational program animations: the roles of students engagement and topic complexity, *Comput. Educ.* 67 (2013) 178–192, doi: [10.1016/j.compedu.2013.02.013](https://doi.org/10.1016/j.compedu.2013.02.013).
- [20] M. Kaluz, L. Cirka, M. Fikar, Virtual laboratory of process control, *Proceedings of the 18th International Conference on Process Control*, 2011.
- [21] F.R.D. Santos, C. Guetl, P.H. Bailey, V.J. Harward, Dynamic virtual environment for multiple physics experiments in higher education, in: *IEEE EDUCON 2010 Conference*, 2010, doi: [10.1109/educon.2010.5492507](https://doi.org/10.1109/educon.2010.5492507).

-
- [22] E.-S.S. Aziz, S.K. Esche, C. Chassapis, Content-rich interactive online laboratory systems, *Comput. Appl. Eng. Educ.* 17 (2009) 61–79, doi: [10.1002/cae.20210](https://doi.org/10.1002/cae.20210).
- [23] E.-S.S. Aziz, Y. Chang, S.K. Esche, C. Chassapis, A multi-user virtual laboratory environment for gear train design, *Comput. Appl. Eng. Educ.* 22 (2013) 788–802, doi: [10.1002/cae.21573](https://doi.org/10.1002/cae.21573).
- [24] W. Tlaczala, M. Zaremba, A. Zagorski, G. Gorghiu, Virtual physics laboratory for distance learning developed in the frame of the VccSSe European Project, *Proceedings of the Fifth International Conference on Multimedia & ICT's in Education*, 2009.
- [25] N. Guizani, J. Fayn, Model-driven architecture for self-adaptive context-aware message routing in pervasive environments, *ICAS 2015 The Eleventh International Conference on Autonomic and Autonomous Systems*, 2015.
- [26] F. Casella, A.G. Bartolini, A. Leva, Equation-based object-oriented modelling and simulation of large-scale smart grids with modelica, *IFAC-PapersOnLine* 50 (2017) 5542–5547, doi: [10.1016/j.ifacol.2017.08.1096](https://doi.org/10.1016/j.ifacol.2017.08.1096).
- [27] J.F.D. Canete, P.D. Saz-Orozco, D. Moreno-Boza, E. Duran-Venegas, Objectoriented modeling and simulation of the closed loop cardiovascular system by using SIMSCAPE, *Comput. Biol. Med.* 43 (2013) 323–333, doi: [10.1016/j.combiomed.2013.01.007](https://doi.org/10.1016/j.combiomed.2013.01.007).
- [28] R. Ponciroli, A. Bigoni, A. Cammi, S. Lorenzi, L. Luzzi, Object-oriented modelling and simulation for the ALFRED dynamics, *Progr Nucl Energy* 71 (2014) 15–29, doi: [10.1016/j.pnucene.2013.10.013](https://doi.org/10.1016/j.pnucene.2013.10.013).
- [29] C. Ngo, S. Dahlmanns, T. Vollmer, B. Misgeld, S. Leonhardt, An object-oriented computational model to study cardiopulmonary hemodynamic interactions in humans, *Comput. Methods Programs Biomed.* 159 (2018) 167–183, doi: [10.1016/j.cmpb.2018.03.008](https://doi.org/10.1016/j.cmpb.2018.03.008).
- [30] T. Maruyama, S. Kanai, H. Date, Tripping risk evaluation system based on human behavior simulation in laser-scanned 3-D as-is environments, *Automation Constr.* 85 (2018) 193–208, doi: [10.1016/j.autcon.2017.10.011](https://doi.org/10.1016/j.autcon.2017.10.011).
- [31] T. Maruyama, S. Kanai, H. Date, M. Tada, Motion-capture-based walking simulation of digital human adapted to laser-scanned 3-D as-is environments for accessibility evaluation, *J. Comput. Des. Eng.* 3 (2016) 250–265, doi: [10.1016/j.jcde.2016.03.001](https://doi.org/10.1016/j.jcde.2016.03.001).
- [32] L. Xing, T. Li, H. Huang, Q. Zhang, J. Han, Efficient modeling and analysis of energy consumption for 3-D graphics rendering *Integration, VLSI J.* 55 (2016) 455–464, doi: [10.1016/j.vlsi.2016.02.009](https://doi.org/10.1016/j.vlsi.2016.02.009).

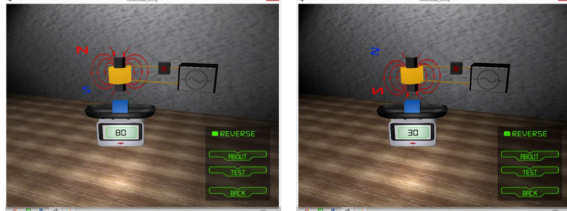
(The Appendix follows overleaf)

Experiment description

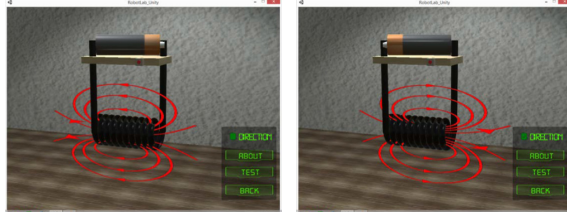
Name: Multi-induction. Simulation: Pupil can change the parameters of speed & power given to the magnet, to interpret their impact on the current inducing and magnetic flux: lamp is lit in (a), and vice versa in (b)



Name: Current Balance. Simulation: When pressing the key (a) The loop coil becomes North pole and the Iron is South pole. Thus, the iron is attracted to the magnet, the reading of sensitive balance is increased, and vice versa in (b)



Name: Circular Loop. Simulation: When pressing the switch (a) the magnetic flux lines are straight parallel lines perpendicular to the plane of the coil, and vice versa in (b)



Name: Current Direction Simulation: current direction and its impact on magnetic flux shape (a), clockwise and counterclockwise in (b)

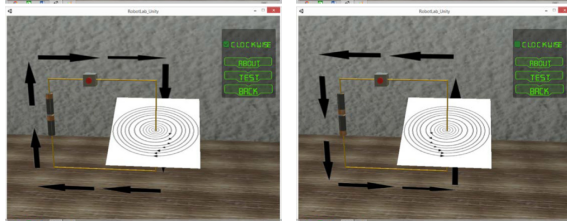


Table 4.
Some simulated experiments.

Corresponding author

Rania M. Ghoniem can be contacted at: RMGhoniem@pnu.edu.sa, prof_rania@mans.edu.sa

For instructions on how to order reprints of this article, please visit our website:

www.emeraldgroupublishing.com/licensing/reprints.htm

Or contact us for further details: permissions@emeraldinsight.com