Development of Interactive Software for Teaching Three-Dimensional Analytic Geometry

ANCA IORDAN
Technical University of Timisoara, Engineering Faculty of Hunedoara,
Revolutiei 5, 331128 Hunedoara
ROMANIA
anca.iordan@fih.upt.ro

Abstract: - In this work will be presented the elaboration of an interactive system for teaching three-dimensional analytic geometry. Incorporating modern methods and techniques, the presented software will lead the user to obtain experience in understanding and handling the knowledge in geometry field.

Key-Words: - Interactive Software, Three-Dimensional Analytic Geometry, Java, Distance Education.

1 Introduction
The multimedia technologies transformed the computer into a valuable interlocutor and allowed the students, without going out of the class, to assist the lessons of different emeriti scientists and professors, to communicate with persons located in different countries, to have access to different information [1]. By a single click of the mouse, the student can visit an artistic gallery, read the originals for writing a history paper or visualize information for a narrow profile, which couldn’t be found five-ten years ago.

One of the main aspects of using computer for lessons is the development of the student’s creative thinking. An optimal mean in this case is the introduction in the computational training means of the interactivity elements [2]. The „interactivity” term means „to interact, to influence one-to-another”. This property of the computational technologies is absolutely unique compared with television, lectures, books, instructive movies etc. The interactivity principle became a heuristic principle of the computer-assisted training, because is connected with some fundamental characteristics of the training process: trainees and trainer’s mutual interaction and influence. In general, the feed-back’s existence is a characteristic feature to any computational qualitative programs.

Before designing the informatics system designated to the knowledge assimilation process from three-dimensional geometry, we achieved, in co-operation with other didactic activists, diverse educational programs that can be used for studying mathematics. We implemented programs for the study of mathematical functions [3], euclidian geometry [4], plane analytic geometry [5, 6] and for doing geometrical constructions [7, 8].

2 Presentation of the Proposed Informatics Sistem

2.1 Sistem’s analysis
In achievement of the interactive informatics system for studying three-dimensional geometry is aimed to reach the following pedagogical purposes:

- interactive presentation of the theoretical concepts and the main results;
- interactive presentation of applications with different difficulty levels for each required subdomain;
- achievement of exact drawings by replacing the pencil and ruler with the mouse.

The program for testing the knowledge will ensure the computer’s intervention in one or more of the knowledge verification steps.

In elaboration of the assessment instrument will be aimed to reach the following qualities:

- validity: measures what is meant to measure;
- fidelity: results almost constant during its successive application;
- applicability: easy to manage and interpretation;
- objectivity: the concordance degree with the assessment results performed by other means is maximum.

The informatics system will be described in a clear and concise manner by presenting the use-cases, using the UML unified modeling language. Each case describes interactions between the user and the system [9].

Representation of the use cases diagram is shown in figure 1.
The conceptual modeling allows the identification of the most important concepts for the informatics system [10]. Because classes means concepts, will be used the following classes to identify the three-dimensional geometry elements: point, line, plane, vector, ellipsoid, sphere, hyperboloid of one sheet, hyperboloid of two sheets, elliptic paraboloid, hyperbolic paraboloid, pseudosphere, helicoid.

The existing inheritance relationships between the classes previously presented can be represented by means of relationship diagrams between classes (Figure 2). Beside these classes corresponding to the geometric concepts, there are classes that contain algorithms for representation both of the independent geometric elements, and the dependent ones, such as: intersection of two unparallel lines, intersection of two unparallel plans, normal vector of a plan, tangent and normal to a quadric into a point. The class Drawing3D contains algorithms for representation of 3D geometric constructions.

The other two kinds of dynamic diagram fall into a category called “Interaction diagrams”. They both describe the flow of messages between objects. However, sequence diagrams focus on the order in which the messages are sent.

The sequence diagram [11] is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

The sequence diagrams for this software are made with ArgoUML-0.24. The diagram illustrate in figure 3 shows the interactions between objects, which have as purpose the drawing of the plan tangent to the ellipsoid into a point.

One can notice that there are interactions between nine objects, out of which the objects of Desen3D, Vector<Element3D> and Graphics2D type are already created, and the objects of Parametru2, Punct3D, Plan3D, Element3D and Ellipsoid type will instantiate during the interactions.

These objects are represented on Ox axis and on Oy axis are represented the messages ordered increasingly in time. At the beginning, the execution’s control is undertaken by the object of Desen3D type which creates an instance of the Parametru2 class.

Now, the control is undertaken by this newly created instance, that will allow the display of a window where will be introduced data for determining a point of the ellipsoid.
Giving back the control to the object of Desen3D type, further will be instantiated the object of Punct3D type, then will be destroyed the object of Parametru2 type.

Further, the execution’s control is transmitted to the object of Vector<Element3D> type which will lead to instantiate an object of Ellipsoid type, representing the ellipsoid of which tangent plan will be draw-up. Further, will be instantiated the object of Plan3D type, representing the plan tangent to the ellipsoid and then will be destroyed the objects of Punct3D and Ellipsoid type. Further, the execution’s control is transmitted to the object of Vector<Element3D> type, in order to add the tangent plan previously created in the list of 3D elements of the geometric construction, and then will be destroyed the instance of the Plan3D class. Finally, will be redrawn the geometric construction, which will include now also the plan tangent to the previously created ellipsoid, by using the object of Graphics2D type.

The diagram illustrate in figure 4 shows the interactions between objects, which have as purpose the drawing of the normal to the hyperboloid into a point.

![Sequence diagram for drawing the normal to a hyperboloid](image)

One can notice that there are interactions between nine objects, out of which the objects of Desen3D, Vector<Element3D> and Graphics2D type are already created, and the objects of Parametru2, Punct3D, Dreapta3D, Element3D and Hiperboloid type will instantiate during the interactions.

These objects are represented on Ox axis and, on Oy axis, are represented the messages ordered increasingly in time. At the beginning, the execution’s control is undertaken by the object of Desen3D type which creates an instance of the Parametru2 class.

Now, the control is undertaken by this newly created instance, that will allow the display of a window where will be introduced data for determining a point of the hyperboloid.

Giving back the control to the object of Desen3D type, further will be instantiated the object of Punct3D type, then will be destroyed the object of Parametru2 type. Further, the execution’s control is transmitted to the object of Vector<Element3D> type which will lead to instantiate an object of Hiperboloid type, representing the hyperboloid of which normal will be drawn-up. Further, will be instantiated the object of Dreapta3D type, representing the normal to the hyperboloid and then will be destroyed the objects of Punct3D and Hiperboloid type. Further, the execution’s control is transmitted to the object of Vector<Element3D> type, in order to add the normal previously created in the list of 3D elements of the geometric construction, and then will be destroyed the instance of the Dreapta3D class. Finally, will be redrawn the geometric construction, which will include now also the normal to the previously created hyperboloid, by using the object of Graphics2D type.

Collaboration diagram, on the other hand, focus upon the relationships between the objects [12]. They are very useful for visualizing the way several objects collaborate to get a job done and for comparing a dynamic model with a static model. Collaboration and sequence diagrams describe the same information, and can be transformed into one another without difficulty. The choice between the two depends upon what the designer wants to make visually apparent.

### 2.3 Component Diagram

The components diagram is similar with the packages diagram, allowing the visualization of the mode in which the system is divided and the dependences between modules [13]. The components diagram puts the accent on the physical
software elements (files, libraries, executables) and not on the logical elements, as in the packages case.

For implementation of the informatics educational system shall be used the Java programming language [14].

The diagram from the figure 5 describes the collection of components which, together, ensures the functionality for the part of the interactive informatics system which allows the achievement of the 3D geometric constructions. The central component of the SuprafataDesen3D.class diagram, component obtained by the transformation by the Java compiler of the SuprafataDesen3D.java component in executable code.

As can be noticed, this component interacts directly only with the components Desen3D.class and MenuAction.class.

The diagrams were achieved by an approach in a new manner, multidisciplinary, of the informatics application, including both the modern pedagogy methods, and the components specific to the discipline to be studied.

3 User Interface

The program’s interface includes a menu bar, a bar with buttons corresponding to the most important operations and the drawing surface (Fig. 6).

From the application’s main window can be selected elements of the three-dimensional analytic geometry [15], such as: point, line, segment, vector, quadrics, pseudosphere, helicoid and geometrical transformations.

Among the most important operations we mention:

- drawing-up of free points or points with certain properties, i.e. the middle of a segment, centre of a quadric;
- drawing-up of certain lines or lines which fulfill certain conditions, i.e. the normal to a quadric;
- drawing-up of plan when are specified three points which will identify the plan or plan with certain properties, i.e. the plan tangent to an quadrics into a point;
- possibility to move the geometrical construction;
- utilization of any geometric transformations, i.e. symmetry, rotation, translation;
- utilization of any orthogonal projection for representing the 3D geometric elements as bidimensional images in the projection plan;
- possibility to change the observation point by rotations around the Ox, Oy and Oz axis.

In the figure 7 are represented the plan tangent and the normal to an ellipsoid using the following projections: ($\alpha$=-45, $\beta$=60, $\gamma$=90) and ($\alpha$=5, $\beta$=120, $\gamma$=90).

Figure 5. Component diagram

Figure 6. User Interface
In the figure 8 are represented the plan tangent and the normal to an elliptic paraboloid using the following projections: \((\alpha=-120, \beta=-30, \gamma=90)\) and \((\alpha=80, \beta=170, \gamma=90)\).

In figure 9 is represented the one sheet hyperboloid obtained through the application of a symmetry against a plan. Are using the following projections: \((\alpha=-80, \beta=25, \gamma=90)\) and \((\alpha=80, \beta=-165, \gamma=90)\).

The interactive system allows the utilization of any orthogonal projection for representing the 3D geometric elements as bidimensional images in the projection plan. Specification of the desired projection is made by means of the three angles made by the three axis of the orthonormal benchmark from space with Ox axis from the projection plan.

After achievement of a 3D geometric construction there is the possibility to change the observation point by rotations around the Ox, Oy and Oz axis. In the figure 10 is presented a sequence from the animation of a pseudosphere and a helicoid, animation achieved by rotation around the Oy axis. The presented sequence includes 12 bidimensional images obtained by using the following projections: \((\alpha=-180, \beta=-60, \gamma=90)\), \((\alpha=-150, \beta=-60, \gamma=120)\), \((\alpha=-120, \beta=-60, \gamma=150)\), \((\alpha=-90, \beta=-60, \gamma=180)\), \((\alpha=-65, \beta=-60, \gamma=-150)\), \((\alpha=-30, \beta=-60, \gamma=-120)\), \((\alpha=0, \beta=-60, \gamma=-90)\), \((\alpha=30, \beta=-60, \gamma=-65)\), \((\alpha=60, \beta=-60, \gamma=-30)\), \((\alpha=90, \beta=-60, \gamma=0)\), \((\alpha=120, \beta=-60, \gamma=30)\) and \((\alpha=150, \beta=-60, \gamma=60)\).

Figure 7. The plan tangent and the normal to an ellipsoid

Figure 8. The plan tangent and the normal to an elliptic paraboloid

Figure 9. The one sheet hyperboloid obtained through the application of a symmetry against a plan

Figure 10. Animation achieved by rotation around the Oy axis

4 Conclusion

The research activity in this field was finalized by the following personal contributions:

- efficient implementation of the classes corresponding to the 3D geometric elements by achieving of inheritance, aggregation and composition relationships;

- implementation of some instantiation algorithms of the elements, of intersection or tangency, using special programming methods, algorithms which contribute to the rapidness and accuracy of the desired geometric elements graphic representation;

- description of the interactions between objects in different contexts, as well as visualizing the mode in which is divided the system and the dependencies between modules;

- introduction of options inexistent in case of other such systems for achieving accurate 3D geometric constructions;

- presentation of concepts and main results corresponding to some chapters from geometry accompanied by the related geometric constructions;
visualization of the 3D geometric elements as bidimensional images in the projection plan, using the desired orthogonal projection.

Comparing the diagrams presented in this chapter with the description of the virtual shop presented in [16], is found that the modeling of the interactive system allows:

- Concrete definition of each case of utilization by presenting the diagrams of activities;
- Efficient implementation of the classes corresponding to all geometric elements, by achieving of inheritance, aggregation and composition relationships;
- Description of the states which an object is passing by and the transitions between these states;
- Visualizing the mode in which the system is divided, and the dependences between modules by achieving of the components diagrams, diagrams which are inexistent in case of the above mentioned virtual shop for geometry.

Comparing the graphic interface corresponding to the option for achieving the 3D geometric constructions presented in the figures with graphic interface of the Cabri interactive system [17], one can notice that the new interface includes some instruments and options of the Three-Dimensional Geometry menu, inexistent in case of the Cabri interactive system, such as: ellipsoid determined by three parameters, hyperboloid determined by three parameter, the normal and the plan tangent into a point to a quadrics, as well as geometric transformations applied to all elementary 3D geometric elements.

References: