Theoretical and Empirical Researches in Urban Management
Number 5(14) / February 2010

COMPUTER AIDED DESIGN IN URBAN ARCHITECTURE 3D MODELING

Nicolae Radu MÂRŞANU
Academy of Economic Studies Bucharest
Piata Romana, 6, Bucharest, Romania
radu.Marsanu@ie.ase.ro

Silvia Mihaela RUSU
Academy of Economic Studies Bucharest
Piata Romana, 6, Bucharest, Romania Romania
rususilvia@yahoo.com

ABSTRACT
The gap from the PC made sketches with the help of the china ink pen and ruler to the digitised drawing boards, high diagonal monitors and 3D projecting is truly spectacular. The increasingly efficient and more specialized programs allow the architects a whole range of facilities providing drawing commands and changes very easy to use, automatic rating, operating simultaneously in multiple windows, building sections and extracts of the plan, 3D views design and even projecting in virtual reality. Applications made to date using virtual reality technologies are characterized, essentially, by generating artificial world and immerging the user into this world. Evolution and effectiveness of such systems are conditioned, first, by the increased hardware performance (computers and specialized devices for interaction) so as to ensure a full immersion of man in the virtual world and, implicitly, a real-time communication with him.
Key words: Computer aided design, Architecture, C# interface, urban 3D modeling, urban design.

1. INTRODUCTION

The term architecture (from Greek αρχιτεκτονική, architektonike) may refer to a process, a profession or documentation.

Architecture is the science or art of design and builds buildings, ensembles of buildings, according to certain rules and proportions, depending on the character and destination of the constructions. Science consists in its functional and technical solving of the buildings. Along with the development of personal computers and software, work of the architects has become increasingly easier. The most used software programs are: AutoCAD for making 2D plans, 3D Studio Max for lifting up the construction (both programs are developed by the company Auto Desk) and the last program is Allplan from Nemetschek, the building is made almost instantly, but this is rudimentary.

The architect is concerned not only by how the building will look like and how it will fit into the environment, but also by the bearing the capacity of walls, the existence of efficient systems to ensure heating, ventilation, supply electricity and water. Architectural plans are drawings which show precise the various aspects of construction plans used for the development or presentation of ideas.

3D Studio Max software uses primitive graphic objects to obtain three-dimensional complex ones. These primitives are text, circle, rectangle, line, cube (parallelepiped right), scope, content, cylinder and *NURBS* curves, which are based on mathematical functions. Primitives can be changed, illuminated and animated, and areas can be covered with colors or images called textures.

For modeling primitives, in order to obtain the desired shape, there are three options:

- Modeling by polygons
- Modeling by NURBS or B-spline
- Tools used for modifying the Surface tool surface

Modeling the polygons is the most common for the design than any other modeling technique, because it allows control over individual polygons allowing efficient optimization. Usually the modulator starts with one of the predefined primitives of 3D Sudio Max software and using bevel and extrude (to lean and to push) tools new details to the model will be added. A more advanced alternative for the polygons is NURBS or B-spline modeling. It smoothes the surfaces by eliminating the straight margins of a polygon. NURBS modeling is a precise mathematical representation of surfaces, as those used for the machine, which can be exactly reproduced at any resolution, whenever is necessary. Surface tool is a tool used for the creation of normal 3D Sudio Max spline, and then applying a modifier called surface. This surface modifier makes a surface from three or four points in a network.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop applications of the console type or graphical interface using the Windows Forms software. While several implementations of the notebook requirements exist, Visual C # is by far the most commonly used. In most contexts, a reference to "C #" represents "Visual C #". C # is a simple language, with around 80 keywords and 12 predefined types of data. It enables structured, modular, and object-oriented programming, according to the modern aspects of the professional programming. The basic principles of programming on object are fundamental elements of C# programming. Unlike C++, it contains a number of new types of data or different data functions and in the spirit of achieving secure code sequences, some features were added (interfaces and delegation), diversified (struct type), modified (string type) or even eliminated (multiple inheritance and pointers to functions).

I chose C # language for the opportunity of creating with it an easy accessible user interface.

2. DESIGNING AND BUILDING THE URBAN HOUSE

2.1. Design the urban house

In order to realize the project, the architect must take into account certain aspects that he will discuss with the owner. A first aspect to be considered refers to the optimal positioning of the house on the ground, following in this sense for the building to be oriented south or regarding the neighborhood in order to determine the shape and volume of the house for a smooth integration into the architecture of the area. In our application the design was made manually, without many details, in order to obtain an overall picture of the scale and proportion to be followed according to the construction certificate. Then using the *3D Studio Max* software, the final plan of the house was made, which included the doors, the windows and the roof (Figure 1). A more detailed plan will be made for the customers, where it will be written what each space represents, where the doors, windows and dimensions are.

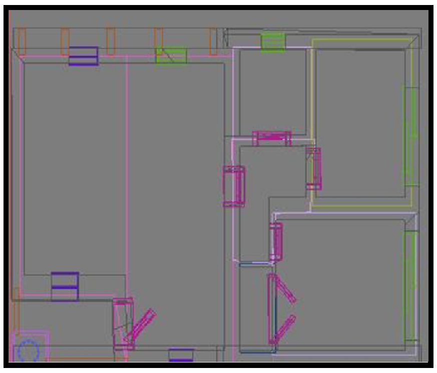


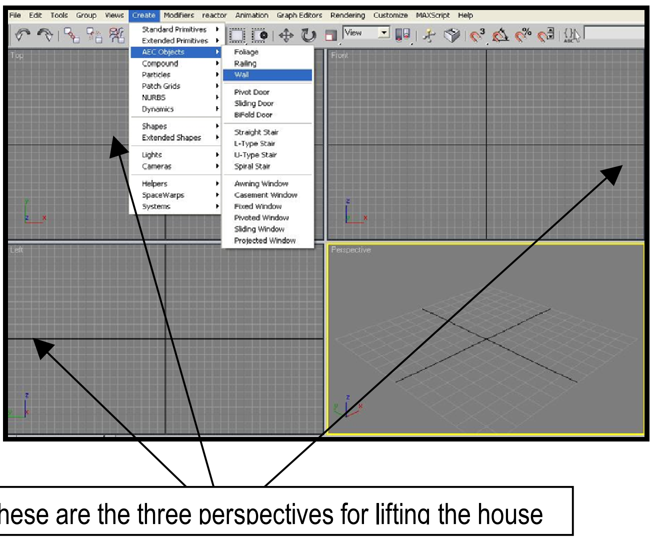
FIGURE 1 - HOUSE PLAN

2.2. Construction the urban house

In order to facilitate the construction of the house in 3D Studio Max software, a plan for every three viewpoints of the program will be created (front, side and top). A picture of every plan will be attached, representing the sketches of the house in the three dimensions. Thus the house can be constructed according to the conditions of the architect.

2.2.1. The walls

3D Studio Max software offers the rapid solution in lifting up the wall with the help of the *Wall (Create - AEC Objects - Wall).* This function is best used for lifting up the internal walls or construction plans, where the walls form an easy to follow shape (Figure 2).



.

FIGURE 2 - CREATING WALLS

A different method will be used because the plan by which the construction must be lifted has a rigid form: a rectangle in the *'TOP'* viewpoint is built (Figure 3). After its construction, it turns into an *editable polygon*.

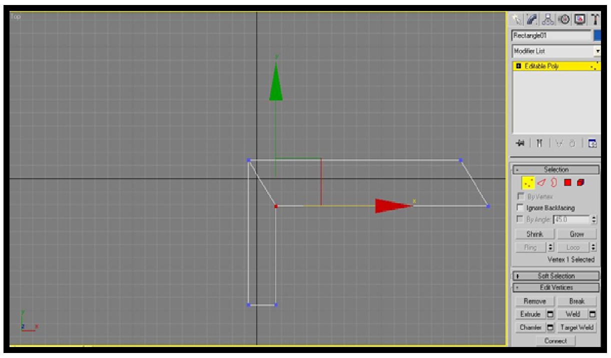


FIGURE 3 - SECOND MODE FOR CREATING WALLS

Then the function in which only the shapes of the form are chosen is selected and the rectangle is modified so that by pressing the *Shift* key the tracing of the next wall left or right is allowed.

2.2.2. The floor

The floors will be built from the some plane (*Create - Standard Primitives - Plane*). Each plan corresponds to a room, which are built in the *Top* viewpoint.

2.2.3. The roof

The *roof* is the most difficult part in building a house. The steps in the process of making this component are the following:

1. Start with a *rectangle* and then turn it into an editable polygon. All the polygons will be selected by using the *loop* function, then using *extrude* and a geometrical shape is obtained;
2. In order to create the tiles the next will be needed*:*

- several segments must be added in order to shape the tiles. This is achieved by selecting the three edges and connecting them (Figure 4).
- with all new segments selected the *Chamfer* function is used to create the space between them.

4. At this point we will have an approximately number of vertical lines. To complete the construction of the tiles the *Slice Plane* instrument is used. The yellow square that will appear will be moved bit by bit down, and each time this operation is made the *Slice* button is preset.
5. The last part is to use *Chamfe*r for the just obtained horizontal lines.

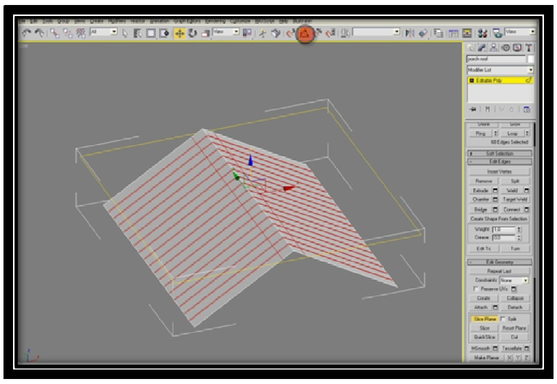


FIGURE 4 - CREATING THE ROOF

2.2.4. The windows

3D Studio Max software has an easy to implement solution, with options for building all types of windows. They can be found

as follows: *Create - AEC - Pivoted Window* (Figure 5).

Types of windows:

- **Fixed Lights** represent those windows which have no possibility of opening. These are the ones in *3DStudio Max*: *Create - AEC Objects - Fixed Window*;

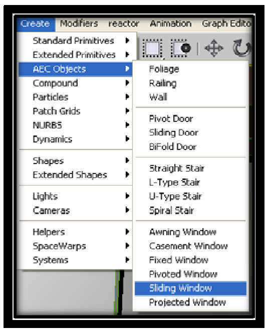


FIGURE 5 - PATH FOR WINDOWS

- **Light Open** represents a window that can be opened having the grip system hanged or pivotal by the framework and can be opened from the inside. Opening windows can be classified according to how it opens. These can be horizontally or vertically sliders (Figure 6) .

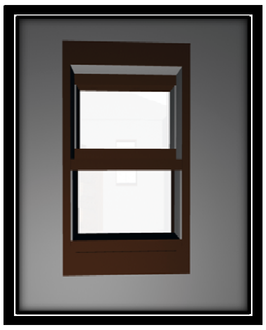


FIGURE 6 - VERTICALLY WINDOWS SLIDERS

2.2.5. The doors

Today, buildings have heated spaces, so you do not have doors to keep heat in separate rooms. Like windows, doors can be built using a function of the program. It is as follows: *Created - AEC Objects - Doors*. After these are positioned one can set if the door is simple or with panels.

3. URBAN HOME FURNISHING

Furniture is the way you live your life every day. Here you should start with all the choices in your house: a functional and welcoming kitchen, a comfortable sitting room which invites you to relax, a quiet bedroom where you can charge your batteries at the end of each day.

3.1. First bedroom (Figure 7)

3.1.1. Bed

The front side of the bed was created from a *rectangle*, which will be shaped as desired, and finally extruded. For the rear a clone of the just created front part will be created. The legs of the bed are four parallelepipeds. The link between the two sides is achieved by attaching two parallelepipeds (made in length), one side and the other of the body.

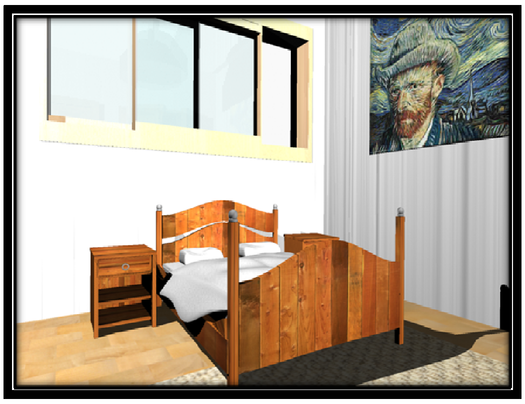


FIGURE 7 - FIRST BEDROOM

3.1.2. Nightstands

It all starts from a rectangle, which forms the side of the nightstand, which is shaped in the part after the shape of the legs. The *extrude and chamfer* tools will be used. It will be cloned and the two parties are merged into a cube, which represent the back of the nightstand. The shelves will also be cubes, which are placed at different levels. In the first shelf a drawer will be put, having as a handle a very thin cylinder.

3.1.3. Cushions

A cube is drawn and in the left viewpoint the two median points from the extremes will be selected and placed below, as to give the impression that the cushion has its ends down. The same operation is performed in the *Front* viewpoint.

The last phase in the realization is the one in which the bumps of the surface are made. The *Soft Selection* instrument will be used. The *falloff value* will be changed to obtain different results when the points will move in different directions. Finally *Mesh Smooth* will be used.

3.2. Second bedroom (Figure 8)

This is done using primitive geometric shapes such as cube, rectangle, cylinder (for handles) and a semicircle for the top. To achieve the desired forms *extrude and chamfer* tools will be used. They will be used to round off the corners, to size irregular lateral parts (modeling by points is chosen from the editable polygon; the superior points will be moved to the outside with *move).*

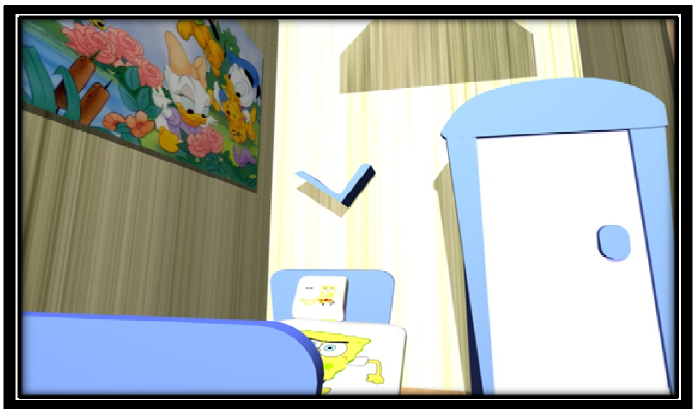


FIGURE 8 - SECOND BEDROOM

Seats are made from a cube, with feet made of cylinders and rounded two times. The feet of the chairs will be cloned for the table. Two cubes will be applied: one for the drawer and one for the top part. The support for the books is made of two lines in the form of V, then we use *extrude* to transform it into a 3D form.

3.3. Bathroom (Figure 9)

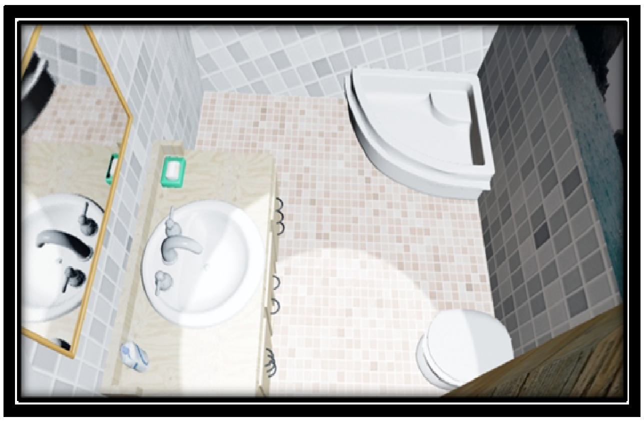


FIGURE 9 - BATHROOM

3.3.1 Sink

The making of the sink and its taps was a complex process, as follows:

1. It is made from a cylinder of which the bottom is cleared. Using the *Loop* function, all the polygons are selected to increase the distance from the center. The *extrude* function is used for the depth of the sink, having a small and negative value. A new segment around the center is created to form the support for the taps. This is done by selecting all the edges and using the *Connect* function. Then, the created segment is moved to the outside of the cylinder. Six points are moved from the back to the front side to obtain the support.

2. The final depth of the sink is given by selecting the other polygons and using the *extrude* function, this time using a much higher value. Then with the *bevel* we obtain the shape of the sink (this function will be used two-three times, with small values).

- Remove the center lines of the form with *Backspace* key.

The taps are made of two overlapped spheres, the above being smaller. The link between the two is obtained using the *Bridge* function (one chooses the bottom of the first and the upper of the lowest). To give the ornament that links the sink and the tap, the bottom of the entire shape is selected and with the help of with *extrude*, *bevel* and *Move* functions we obtain what we want.

The battery is also a cylinder, but this time the *Blend* function is used to form the curve.

The desired angle and the part where the curve is wanted are chosen. *Mesh Smooth* is certainly used for these forms.

The same principles will be used for the sink and the closet from the kitchen. Of course they will be changed according to furniture specifications.

3.3.2 Shower cabin

The cabin is made from a simple rectangle. It will rise three times in different proportions, in order to achieve different components. The *Chamfer* function is used for corners because when we turn on the *Turbo Smooth* function these will keep their perpendicularities. By *Chamfer* we also achieve the form of the shower cabin. The functions used for the sink will also be used here for the depth.

3.3.3 Toilet vase

Start with a *Vertex rectangle*. The points are modeled using the *Vertex* mode in order to get the shape of the vase. The same steps as above are respected for the depth of the vase. For the lid we use a rectangle, which in the *Top view* mode modifies itself by the shape of the vessel. The interior parts are deleted, and the top part is left as it is. *Shell* and *Mesh Smooth* will be used for flattening with two or three iterations. The action by which it gets cleaned is turned on by a button placed in the wall.

3.4 Kitchen (Figure 10)

3.4.1 Refrigerator

This is a simple form to create. It is simply a rectangle that was divided both ways with the help of *Connect* function. Then to give the feeling of doors, polygons have been selected from the segments and were extruded until almost the end of the shape. The handles are simple cylinders, the one for the cold box having enabled the *Bend* option, with a small value.

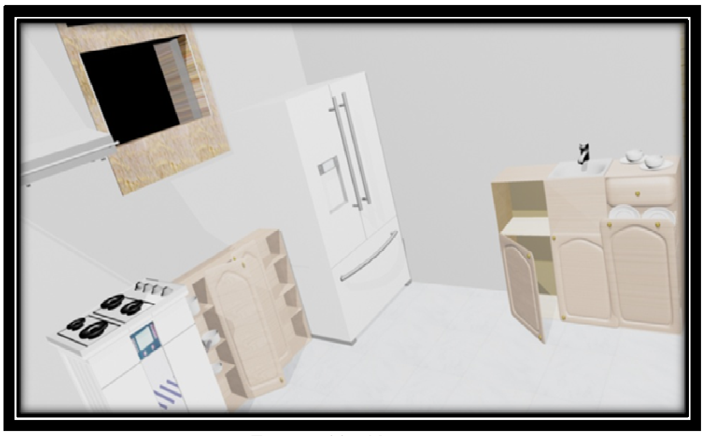


FIGURE 10 - KITCHEN

3.4.2 Cooker

All starting from a rectangle, we divide it by height in order to create the two spaces reserved for the digital screen and glass from the oven. The buttons for the fire ignition were made from a semi-sphere with the bade a little to the exterior.

The burners are made from circles, which were applied these functions: *Shell* and *Mesh Smooth* and having checked the option to enable *renderer* and *enable in viewport* in order to give wide. In order to make the glass of the oven one will choose *Raytrace* in the screen for choosing the material at the option *maps->reflection.*

3.5 Living room (Figure 11)

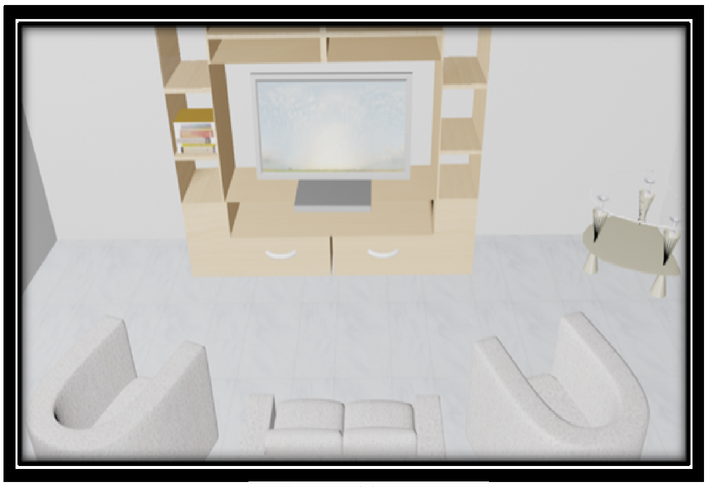


FIGURE 11 - LIVING

Theoretical and Empirical Researches in Urban Management

Number 5(14) / February 2010

3.5.1 Sofa and Armchairs

Both couch and armchair were made of a rectangle. The sections will rearrange for each one, in order to lift up the handles. Then the *Chamfer* is used for each of the corners, because when we apply *Turbo Smooth or Mesh Smooth* with three iterations, we don't want these to be affected. The cushions are made of cubes, shaped in *Top View* by the form of each one. Legs are four simple cylinders.

3.5.2 TV

The support is built up from a rectangle, where the *Slice Plane* function is applied. With this we can divide the rectangle so that the other sustaining part can lift up at 30 degrees angle. Then using the common functions *extrude* and *bevel* we shall make the other parts of the TV.

3.5.3 Table and vases

The construction of the table starts with the sustaining supports. They will be built from a simple line drawn obliquely that will take the shape of overlapping cones with the functions *Lathe and Shell*. The parts that connect the legs and the two windows are made also from lines, but they will be drawn by a specific shape. It is enough for the line to be drawn only for half of the form, because the functions above can rebuild the other half.

The windows are also made from a line modulated shape, but the *Extrude and Mesh Smooth* functions were applied for the shape to be uniform. The vase is built up by the same principles. Only the shape of the outline and the type of used material are different. The window for materials opens and we choose *ARCHITECTURE* instead of the *Standard* button. Then the desired material is applied in the *Phisical Qualities-Diffuse Map* section. The same material is selected in the *Special Effects–Bump* section (Figure 12).

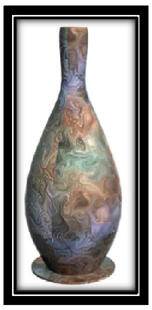


FIGURE 12 - VASE

4. PRESENTATION INTERFACE

The movie presentation is made in *3D Studio Max*. This software has a multitude of extensions that can save created animations. The most commonly used extensions are *.bmp .png* for images and *.avi . mov* for movies. Each room will have a separate movie in order to obtain a very good quality. An interface of the user will be made in *Visual Studio 2008 – C#* after all four movies have rendered. This interface can also be created in *Adobe Flash* or *Adobe Director*.

The *Adobe Flash CS3* software uses the principle of key frame animation. Each graphical component is attached to a different layer so that any changes made to the graphic objects may not affect other components. Also, using *Adobe Flash CS3* we can use *Action Script*. This is achieved by selecting and placing the object code associated to the object in a special window. The process of rendering a photo and the movies in *3D Studio Max* will be firstly explained.

4.1 Photos

The pictures are the easier to obtain. Select the desired viewpoint, select the scene, and then by pressing F9 a new window will appear. The graphical elements that will form the picture will be arranged in this window. There are four ways to visualize the pictures, the four channels*: red, blue, green* and *black & white (*Figure 13*).*

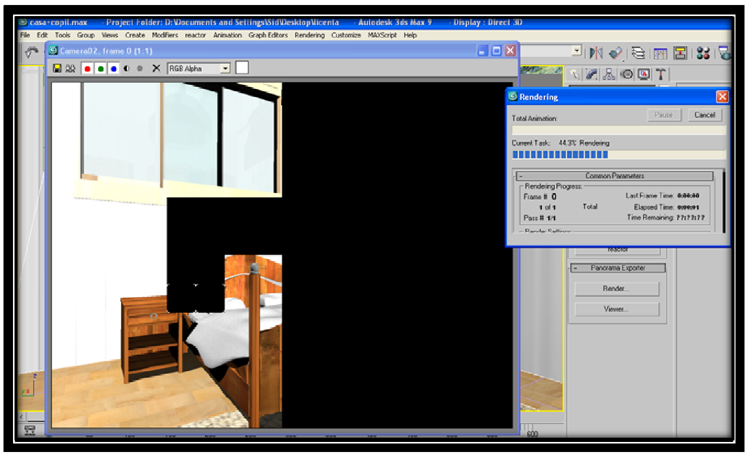


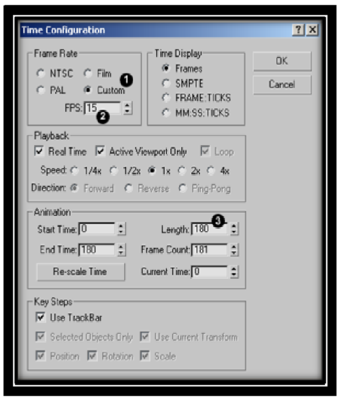
FIGURE 13 - RENDERING A IMAGE

4.2. Movies

The above function cannot be used for the movies. For this one must follow the way: *Rendering–Render*. To create the desired animation, *Active Segment time* must be checked. Thus all, the active

Theoretical and Empirical Researches in Urban Management
Number 5(14) / February 2010

frames will get selected. Then we choose the quality of the movie from *Output Size*, with options for pictures, slide shows and videos.



Time Configuration

Frame Rate: NTSC / PAL / Film / [✓ Custom] 1
FPS: 15 2

Time Display: [✓ Frames] / SMPTE / FRAME:TICKS / MM:SS:TICKS

OK | Cancel

Playback: [✓ Real Time] [✓ Active Viewport Only] [✓ Loop]
Speed: 1/4x / 1/2x / [✓ 1x] / 2x / 4x
Direction: [✓ Forward] / Reverse / Ping-Pong

Animation:
Start Time: 0 — Length: 180 3
End Time: 180 — Frame Count: 181
Re-scale Time — Current Time: 0

Key Steps: [✓ Use TrackBar]
[✓ Selected Objects Only] [✓ Use Current Transform]
[✓ Position] [✓ Rotation] [✓ Scale]

FIGURE 14 - TIME CONFIGURATION WINDOW

Finally, push the *Render* button and the same window from the image rendering will appear.

Another method is as follows:

- an outline of the route you want to view with a line is drawn and a camera is placed in *Left View*. Focal distance is selected and the *Show Cone* option is checked;
- the *Time Configuration* window is opened to set on how many frames the camera will follow the route (Figure 14);
- the *Motion Tab* window is opened to place the camera on the drawn route
- select *Position: Position XYZ* and press the question mark. A new window will appear. The *Path Constraint* button must be pressed in the window. Then select *Add Path* and then choose the line drawn earlier. Also do not forget to *check Follow and the Along Path option must have value 0*.

4.3 User interface

Of the three available programs, we have chosen to realize the interface in *Visual Studio 2008 - C #*. Although it requires knowledge of programming, the application can be run on any platform because the executable will be an *.exe* type and not an *.swf* type, like in *Adobe Flash*.

Pictures and animations made in *3D Studio Max* were firstly chosen, to create the presentation of each room. The interface will be composed of forms which include: *Picture Box, WMP, buttons, Panels* and also contextual menus that allow *print preview, page setup* and *print.* We also put for each form, a icon representing a house in the upper left corner. An image that integrates all the rooms was chosen to ease up the navigation in the whole house.

The script that connects two forms is:

```
private void button1_Click(object sender, EventArgs e)
{
Aplicatie frm2 = new Aplicatie();
frm2.Show();
}
```

A panel was used for each room, the option for transparency was checked and the script above was attached to it. In addition, when we choose to *enter* one of the rooms the previous form will close in order not to create a suite of windows of form 2 (the one from which one chooses what room to view). The script to close a form is: *this.Close ();*

Then the same steps will be followed for each of the rooms. Thus we have:

4.3.1 Integration of the video part(where it is necessary) in the form[1]

The script is:

```
private void IncarcaVideo()
{
string cale = @"C:\video\parinti.avi";
axWindowsMediaPlayer1.URL = cale;
axWindowsMediaPlayer1.settings.autoStart = true;
}
```

The function that loads the animation must be declared in the constructor of the form. The only disadvantage of this component in C # is that: if the file is put on another computer or another partition we must change the code a little and enter the new path. The two competing programs (*Adobe Flash* and *Director*) integrate them automatically, without being necessary to copy them on that partition.

4.3.2 Integration the pictures in the form

This part of the application is the only one that does not require a script because it does not perform or process any event. Right click and choose the option *Choose Image* to assign a picture to each *Picture Box*. All the pictures are automatically stored in the *Resources* folder.

[1] http://www.msdn.microsoft.com/enus/library/dd564585(VS.85).aspx

4.3.3 Print Preview

The option *Print Preview Document* must be chosen from *Tools* in the first phase.

The script will be:

```
private void printPreviewToolStripMenuItem_Click(object sender, EventArgs e)
        {
            pageSetupDialog1.PageSettings = new System.Drawing.Printing.PageSettings();
            if (pageSetupDialog1.ShowDialog() == DialogResult.OK)
                printDocument1.DefaultPageSettings = pageSetupDialog1.PageSettings;
        }
private void previewToolStripMenuItem_Click(object sender, EventArgs e)
        {
            printPreviewDialog1.ShowDialog();
        }
```

The first line allocates memory for the *page settings* type created object. The following condition is whether the user closes the window by pressing the *OK* button, and then the settings for Object 1 will be equaled with the ones from the *setupdiag* object.

For viewing each photo in a page, the script is:

```
private void printPreviewToolStripMenuItem_Click_2(object sender, EventArgs e)
        {
            this.printPreviewDialog1 = new PrintPreviewDialog();
            this.printPreviewDialog1.Document = this.printDocument1;
            this.printPreviewDialog1.Show();
 }
```

This script first allocates memory for the picture that was selected for preview, then the property of the document with the printer in set, although practically the application simulates the printing, by making it only visual. The last instruction is to copy each picture in part because it was chosen the option of printing a picture on a page.

4.3.4 Page Setup

We must drag the print Document instance from Tool Box, just like it was at print preview. The script will be:

```
private void pageSetupToolStripMenuItem3_Click(object sender, EventArgs e)
        {
            pageSetupDialog1.PageSettings = new System.Drawing.Printing.PageSettings();
            if (pageSetupDialog1.ShowDialog() == DialogResult.OK)
                printDocument4.DefaultPageSettings = pageSetupDialog1.PageSettings;
}
```

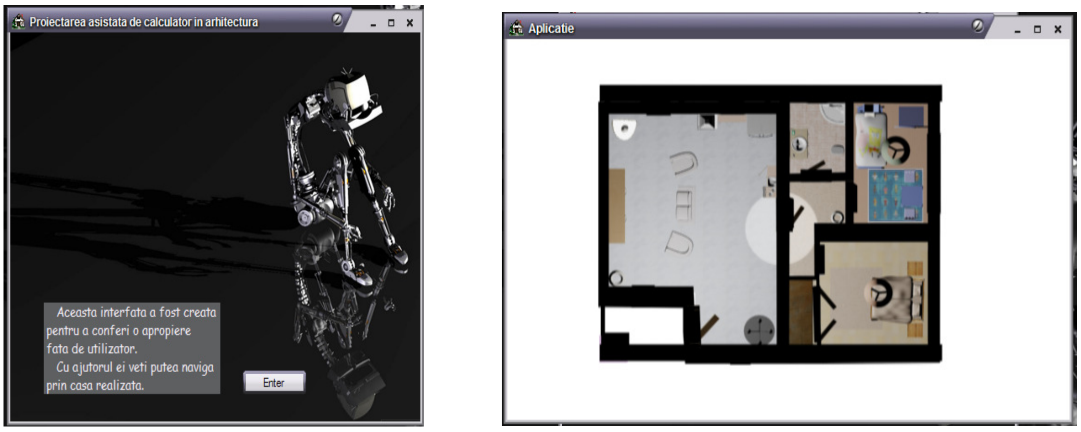
4.3.5 Print

Here we also have to drag up the print tool from *ToolBox* and write the following code:

```
private void printToolStripMenuItem_Click(object sender, EventArgs e)
        {
            printDocument1.Print();
  }
private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
        {
            e.Graphics.DrawImage(Properties.Resources.w, new System.Drawing.Point(100, 100));
}
```

The last function from the actually played script draws the selected picture on the printing page. *Properties.Resources* is the resource that will draw the picture with the name *w. new System.Drawing.Point (100, 100)* is the place where the drawing will start. Upper left corner has the coordinates: 0 and 0.

This script will have to be adapted for each picture in hand, with the name associated to the photo. The final project will have this final form (Figure 15) :



Aceasta interfata a fost creata pentru a conferi o apropiere fata de utilizator.
Cu ajutorul ei veti putea naviga prin casa realizata.

FIGURE 15 - USER INTERFACE

For example inside of a room will look like this (Figure 16):

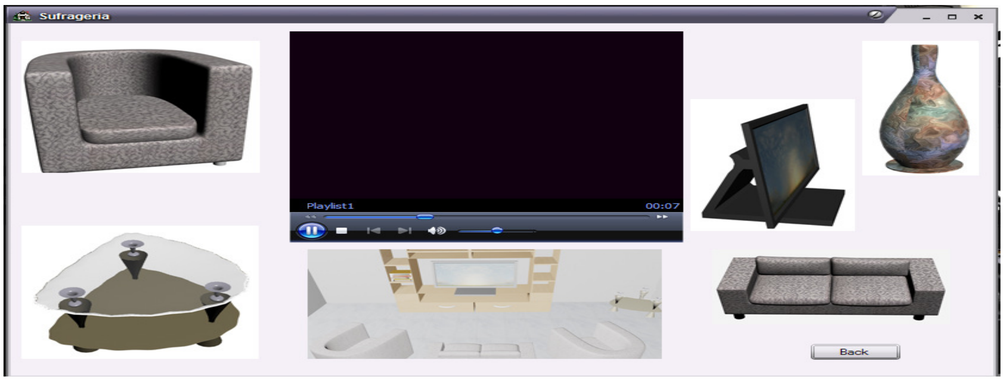


FIGURE 16 - INSIDE OF THE LIVING ROOM

Theoretical and Empirical Researches in Urban Management
Number 5(14) / February 2010

Theoretical and Empirical Researches in Urban Management
Number 5(14) / February 2010

6. CONCLUSIONS

Optimizing human-machine interface using virtual reality technology is certainly a topic of interest in computer assisted design capability through this innovative tool to provide a contact between humans and the system of *computer-aided design*.

In such an approach, the subject on which the designer works is a virtual prototype that is generated in a particular initial state (situation) and can be reconstructed at any stage of the design process, without a lot of effort. The main advantage is that, in general, everything is started from simple drawings and can evolve to complex objects and ways of positioning them in the picture after some considerate in the interior of a synthetic environment using 3D projecting techniques (traditional or modern).

The prototype on which it operates in relation to ones own creativity and experience, the human designer can be played in a natural context at any stage of development, so that fundamental defects, adjustments, unwanted characteristics can be immediately identified and corrected, before the actual achievement of the project.

Using this solution in architectural design, one can create an advanced environment, by simulating all the "features" of a building, including lighting, heating and acoustics. One can also circulate through the rooms, experience some features of the project (for example: colors, volumetric space, fixing the position and intensity of the light, ventilation and devices of heating / cooling the room). We remind you that, in the classic design, testing and verification of design solutions are achieved by carrying out simulations to scale (or low physical models) of the designed object, and in a costly and in some cases less relevant way. The computer-aided design in architecture based systems, based on virtual reality, allow rapid generation of full scale virtual models, on which you can make direct experiments. An additional advantage offered by this type of virtual simulation of the building (the final product of the design process) arises from the fact that the eventual buyers (customers) can examine themselves the project at an early stage, having the chance to see what they are buying and the chance to become effective partners of the designing project through virtual reality. An approach that will present a real interest in the coming years is the integration of a computer-aided design for architecture in a much more complex system based on visual arts (virtual theater, teleconference, virtual museum).

The conclusion is that we can create a virtual world with the collaboration of the architects, visual artists, teams of computer assisted designers, programmers and specialists in sound engineering can contribute to the reconstruction of monuments and historical sites, thus conserving the authenticity of such existent complexes.

REFERENCES

Barry, R. (1999). The Construction of buildings (Vol. 1). Blackwell Science.

Barry, R. (1999). The Construction of buildings (Vol. 2). Blackwell Science.

Smeureanu, I., Dârdală, M. and Reveiu, A. (2004). Visual C#.NET. Bucureşti: Editura CISON.

Microsoft. (2009). MSDN. Retrieved from http://www.msdn.microsoft.com/enus/library/dd564585(VS.85).aspx

Murdock, K. L. (2008). 3ds Max 8 (Vol. 1). Willey:Indiana.

Theoretical and Empirical Researches in Urban Management
Number 5(14) / February 2010