Plant Simulation Research Based on GLSence

HUANG YUN-KAI
Department of Earth Science, College of Land Resource Engineering，Kunming University of Science and Technology，Kunming 650093，China

LIU FANG-CHENG
Department of Earth Science, College of Land Resource Engineering，Kunming University of Science and Technology，Kunming 650093，China

ABSTRACT **— Firstly, some present methods of plant simulation are briefly discussed. Then tree simulation program developing processes are put forward which take Delphi7 as the software developing tool and are combined with module database of GLSence. An example is given to explain the way that GLSence module database's application on tree simulation in detail, the create processes of simulated tree and relevant genuses' primary attributes and operations. This program developing method achieves a few applied functions such as 3D static display of simulated tree(overground part) and growing progress simulation.**

KEY WORDS-simulated tree; GLSence; genus; program development

I. INTRODUCTION

Virtual plant is the computer simulation of plant's configuration in 3D space which is a primary research tool for botanists, a main research aspect of digital city and computer simulation field. Virtual plant's computer modeling research possesses extensive prospect on agroforestry、landscape design、computer managed instruction and digital city research fields. Presently, there are three modalities to express virtual tree: Bill board tree, Simulated tree and Fractal tree. Fractal tree is not fit for real time demanding circumstances for taking up massive resources. Taking a tree picture as the veins to a rectangular surface, Bill.board tree processes non-tree structure parts in the picture by using scenography. By this way, a whole tree could be created and the veins could face the observer all the time. Due to all parts of Bill.board tree in one picture, we could not precisely control it. This modality is unsuitable for the tree growing process simulation. Simulated tree uses a series of polygons to piece up tree crown and trunk, and then a simulated tree is established after attached the given materials. Every part of the tree being modeled separately, therefore, this method could be used in the growing process simulation.

At the present time, the way based on L system is frequently-used in plant simulation which could create comparatively holistic geometry model. Put forward by biologist Aristid Lindenmayer in 1968, L system is applied in modeling developing process of simple multi-cellular organisms initially. Then, it is used in modeling advanced plants which are of complicated branch structure gradually. And now, some plant simulation softwares based on L system become more mature than ever before.

The research on virtual plant started later than other countries in China and no relevant civil softwares are used in this field. Therefore, to develop a plant simulation software is of significant meaning to promote Chinese software developing level. Taking simulated tree as an example, a brief discussion is made about virtual plant software developing in this paper. General speaking, virtual plant should be included in 3D simulation field. At present, OpenGL is normally used as the graphic interface and combined with some advanced computer language to develop program in 3D simulation field. This developing mode requests developers on top of OpenGL functions. Some disadvantages should be noticed about this mode such as demanding to developers, long developing period, vast devotion and poor feasibility. Delphi's program package GLSence could make up all above disadvantages. As a 3D scene developing activex group and based on OpenGL, GLSence packs OpenGL tightly and provides visual module, entity and applied 3D scene describing mode in design phase. And with the provided abundant 3D engines of base genus, developer must not on top of complicated OpenGL interface functions to develop programs. Furthermore, GLSence is a free activex based on Mozilla Public Licence and sharp-orm. According to the needs, user could perfect source code by adding new genus or activex to expand software functions.

II. STRUCTURE OF GLSENCE ACTIVEX GROUP

TObject is the base genus in all GLSence genuses, and all other objects are inherited from it. There are two subgenuses of TObject, TCameras and TScene Objects, which distinguish observer and observed entity. TCameras are examples to define lens, and each lens example owns its basic attribute, such as location, direction, focus and lens depth. TScene Objects derive TGLLight Source (light source) and TDummy Cube (entity). Light source is in possession of some attributes like light source type, direction, color and lighteness. Entity is in possession of some attributes like geometry confituration, size, location, exterior material and vein. Developer could control GLSence's performance by setting up different attribute values. TScene is GLSence's basic object and all other objects' parent vessel, all the created visual objects or special good effects are displayed in it. TScene is generated by GLSence(scene generator). There could be multiple scene generators in one project, in other words, there could be multiple scenes in one project at the same time. Vidicon is the visual interface between scene and user, with it user could observe the contents in the scene. Just as in the real world, GLSence need light to light up objects in the scene, and the light impacts the scene effects profoundly.

III. CREATE SIMULATED TREE

There are two parts of virtual plant, overground and underground parts. Only the overground part's(trunk's) simulation is discussed in the paper. Leaves and trunk's configurations determine tree's appearance and those are the keystone of simulated tree research. Leaves and trunk's simulation processes are discussed separately.

A leaves' simulation

TGLTreeLeaves genus is defined in GLSence for simulating leaves. What TGLTreeleaves genus main attributes and operations are outlined in Figure 1. It is very easy to make use of TGLTreeLeaves. AddNew, the important operation, is defined in TGLTreeLeaves genus to create the leaves. AddNew operation takes chartlet to simulate leaves' configuration, that is to prepare a leaves' picture first, then adjust the picture size according LeatSize's attribute value of TGLTree genus. Lately, as a chartlet, the disposed picture is put to the end branch's top.

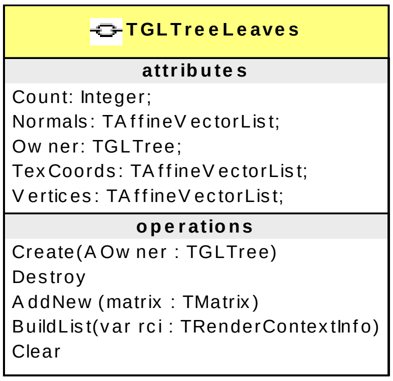
TGLTreeLeaves
attributes
Count: Integer;
Normals: TAffineVectorList;
Owner: TGLTree;
TexCoords: TAffineVectorList;
Vertices: TAffineVectorList;
operations
Create(AOwner : TGLTree)
Destroy
AddNew (matrix : TMatrix)
BuildList(var rci : TRenderContextInfo)
Clear

Figure 1-Structure of tgltreeleaves

In GLSence, GLMaterialLibrary genus takes charge of all the materials in the system, namely, all the materials are leaved in its care. The needed materials are added to the GLMaterialLibrary by AddTextureMaterial opration for later use. The adding process is done only once, therefore, this managing method could cut down system spending. A materials database could be established according to the types of trees, just as outlined in Table 1. In order to do so, TreeType should be added in TTree genus. Then the corresponding materials could be loaded in accordance with the different tree types to be simulated.

TABLE 1-STRUCTURE OF MATERIALS TABLE

FieldName	Type	FieldUse
TreeType	Char(20)	Load tree's type
MateriaKind	Char(10)	Load material kind
MaterialFileName	VChar(12)	Load material file's full name

B trunk's simulation

TGLTreeBranch is defined in GLSence for simulating trunk. Its main attributes are outlined in Figure 2.

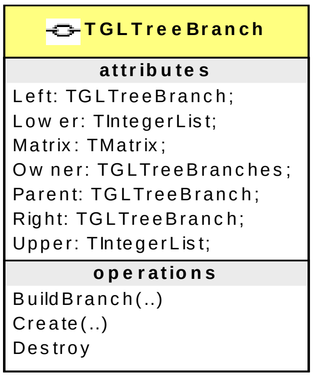

Figure 2-Structure of tgltreebranch

It is very complicated to simulate trunk for a simulated tree. That is because divarication laws and curvatures caused by gravity are of great differences for different tree types. Plants configurations have a lot diversity, but most of them follow a basic law, that is self-similarity law. The frond is a multilevel self-similarity system. Subsystems, which are relatively free-running parts of parent system, are the epitomes of the whole configuration. For example, secondary divarication system is often similar to the whole plant. On account of divarication self-similarity law, recursion transfer operation could be used to simulate trees. BuildBranch is the leading operation to simulate branch in TGLTreeBranch genus. The branch creating process is depicted as following:

```
  if (((Level+1)>=Tree.Depth) or
(LeftRadius<Tree.LeafThreshold)) then begin
        Tree.Leaves.AddNew(LeftMatrix);
  end else begin
        Inc(Branches.FCount);
        FLeft:=TGLTreeBranch.Create(Owner,Self);
       FLeft.FCentralLeader:=central_leader and
       LeftRadius>=RightRadius);
       FLeft.BuildBranch(LeftBranchNoise,LeftMatrix,Tex
       CoordY,BranchTwist,Level+1);
  end;
```

Firstly, the above codes estimate the relation between Level value and Depth. If the Level value is greater or equal to Depth, then transfer Leaves.AddNew to add leaves, or continue to create divarications until Level value is greater or equal to Depth. After all the branches are created, pastes corresponding materials to the branches, and then, the almost real trunk are established. The codes suppose the divarication law is biramous, and Level is the present created divarication level of the whole plant, Tree.Depth is

TGLTree's attribute which leave the whole plant level in its care. In Figure 3, the Depth of the created tree is 3, Level value is from 1 to 3. While creating the simulated trees, programs establish the trunk which Level value is 1, the secondary branches which the Level value is 2, etc, until the trunk and branches of the whole tree are created.

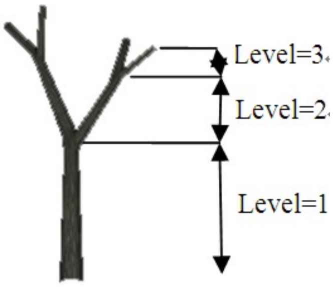

Figure 3-level relations of tgltreebranch genus

C create simulated tree

Until now, most work of creating simulated tree is finished. TGLTree genus is defined for the developer later use. A couple of important attributes are defined in TGLTree to simulate different types of trees. BranchAngle: to confirm the angle between present and superior branch, different angle determines different tree type. BranchRadius、BranchSize: to confirm the new created trunk spatial location which could adjust the integer configuration of the trees. BranchTwist: branch bending degree. LeafSize: leaf's size. It is easier for simulating trees with the defined relevant genuses by assigning value to the corresponding attributes of TGLTree genus, and tree configurations are so different with different attribute values (Figure 4).

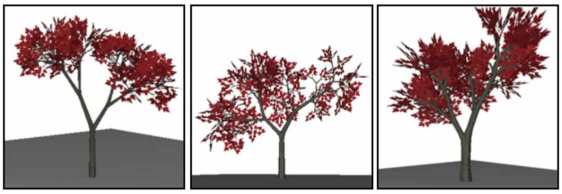

Figure 4-Simulated trees with different parameters

With all the work above, the main characters of a simulated tree are parameterized, that is relevant attributes are defined to main characters. We could simulate trees' growing process by transferring corresponding attribute value at fixed time. Loop structure and timer could be used in time fixing operation. In view of circumstance factor, a circumstance factor table could be established in the database. The system could extract corresponding data in time order to perfect relevant attributes of tree genus for a more realistic simulation.

IV. CONCLUSION

A brief discussion of GLSence module database for creating simulated trees is made in this paper. The key point is parameterized simulated tree which the method could be employed in some fields such as agroforestry and virtual city. And there are some shortages should be noticed in the method like if the Depth value is excessive, the EMS memory deficient condition could occur. That needs further more research work.

REFERENCES

[1] XU LI-MING, JIANG YU-MING. 2006. Modeling and Realization of RoamingVirtual Scene. Journal of System Simulation, 18:120-124.

[2] ZHAO XING, XIONG FAN-LUN, PHILIPPE DE REFFYE. 2001. A New Image Synthesis Algorithm for Bending Branches. Journal of University of Science and Technology of China, 31:714-720.