Advances in Intelligent Systems Research, volume 147
International Conference on Network, Communication, Computer Engineering (NCCE 2018)

Dynamic Scheduling of Terrain Based on Unity

Fang Tian[1, a)], Lihong Lou[2, b)]

[1]*Guangdong University of Technology, Guangzhou 510006, China*
[2]*Guangdong University of Technology, Guangzhou 510006, China*

[a)] 915325482@qq.com, [b)] luo_lihong98@163.com

Abstract. Aiming at the problem that large-scale terrain loading in unity has large memory usage and the scene operation is not smooth enough, a terrain dynamic loading method suitable for Unity is designed. The ray collision detection method is used to detect the terrain block of the current viewpoint in real time, determine the surrounding terrain blocks, and use a two-dimensional array to cyclically store the current display area data, realizing the real-time scheduling of the terrain block data. Experiments show that this method can be well applied to Unity's loading of large-scale terrain, and the terrain can be smoothly drawn when roaming.

Key words: Unity, Terrain, Dynamic scheduling, Ray cast.

INTRODUCTION

With the wide application and demand of virtual reality technology in many fields such as game entertainment and urban planning, scene roaming has also become a hot topic. Real-time roaming of large-scale terrain is one of the important research directions. Its universal difficulty lies in How to deal effectively with the contradiction between large amount of data loading and terrain real-time drawing under limited hardware conditions. In the current study, the loading of large-scale terrain will generally be resolved from two aspects. One is to simplify the terrain model, such as multiple level of detail based on viewpoint change [1,2]; The second is to reorganize the terrain data in blocks, and then dynamically load the data in the visible range according to the position of the viewpoint when drawing the scene. This reduces the amount of data that can be loaded at a time. The most commonly used method is the data paging technology [3,4]. Usually these two methods will be combined to use, such as Yong Tang and others will divide the terrain data into two high-precision and low-precision block organization, in the drawing of high-precision LOD dynamic scheduling [5]; Zhaoting Ma, et al. used the ROAM algorithm to simplify the terrain, and used a quadtree index to dynamically schedule [6].

Unity is one of the mainstream 3D engines on the market [7], and the market share has been very high, and its use of C# open source framework Mono to implement CIL enables its products to be easily released across platforms. And its provided AssetBundle technology, can be a good realization of data network transmission and dynamic management, there is a good development prospect.

This paper combines the advantages and characteristics of the Unity engine to study the real-time rendering of large-scale terrain and designs a dynamic scheduling method suitable for Unity terrain.

TERRAIN BLOCK DYNAMIC SCHEDULING ALGORITHM

Get Terrain Blocks

Unity is a mature 3D engine that supports all common formats of models and texture files. It is compatible with most of the modeling software such as 3Ds max and Blender supported export formats and comes with a model editor

and a terrain editor. Its large user base also brings a variety of features to Unity's plug-in support. This greatly facilitates the developer's setup of the scene and makes it easy to obtain terrain data. Unity can be obtained through the following ways: It can be drawn by its own terrain tool, imported from 3D modeling software, and can also use the plug-in to obtain the terrain data of the specified area from the satellite map. The article obtained by Terrain Composer plug terrain resources, and cut into a regular grid block, compressed, packaged and stored prefab made into AssetBundle file, and record the number of ranks. Prefab is an important concept in Unity. It is a method of combining multiple components in advance for object abstraction. Similar to the factory method, it can make asset reusable. It has information such as location, rotation, and scaling of assets. Its instantiation or destruction can realize the dynamic creation of the scene. The terrain is a kind of resource and can be dynamically managed through it.

Packaged terrain resources can be stored and managed using XML files. Its structure is as follows:

```
<chunks>
   <chunk>
      <bundleName>AssetBundle name</bundleName>
      <numX>row</numX>
      <numZ> column </numZ>
  </chunk>
…..  <! --Multiple terrain blocks -->
</chunks>
```

Relevant Dynamic Scheduling Strategies

Unity is a powerful engine that can automatically perform LOD processing on the terrain, which greatly facilitates the work of developers. Here we will focus on the study of dynamic scheduling of terrain blocks.

The XML file is parsed, and a terrain block is used as a node in the following structure, and the corresponding row and column number of the record is stored in a two-dimensional array.

```
class Node
{
      string bundleNane;
      int row;
      int col;
}
```

As shown in Figure 1, at the time of initialization, 9 terrain blocks will be loaded and the viewpoint will be in the terrain block (2, 2). When the viewpoint moves to the right and enters (3, 2), the 3 shown in the figure will be loaded. Terrain blocks, and unload (1,1), (1,2), (1,3) terrain blocks, so that the amount of data in the memory within the controllable range.

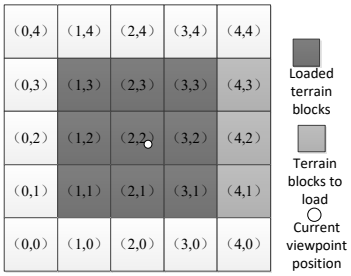

FIG.1. Terrain data status

The ray collision detection method is used in this paper to obtain the terrain block where the current viewpoint is located. Unity provides developers with Ray and RaycastHit projection collision information classes. You can use the camera as a ray origin and emit a ray in the negative y-axis direction (ground direction) to detect the underlying terrain block and obtain the current view point block. name.

Dynamic Scheduling Algorithm Implementation

1. When the program is initialized, depending on the position of the viewpoint, the stereotyped block where the viewpoint is located and the 8 terrain blocks around it are loaded, and the storage location of the shape block where the viewpoint is located is recorded. As shown in Figure 2, a 3 × 3 two-dimensional array of loops is used to store the terrain nodes that need to be displayed. The storage location of the current block is (1,1).

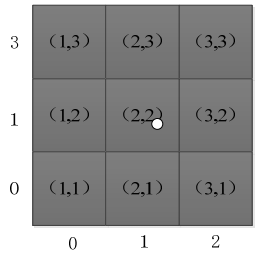

FIG.2. Scene initialization state

2.The program enters the frame update and performs ray detection. If the detected terrain block does not change, nothing is done; If the terrain block changes, then iterate through the two-dimensional array, find the storage location of the terrain node according to the terrain name, and determine the terrain block to be unloaded according to the current storage location and the previous storage location, and according to the row and column numbers stored in the node. Update the terrain block that needs to be displayed.

As shown in Figure 3, the storage state of the array when the viewpoint moves to the right into the (3,2) terrain block.

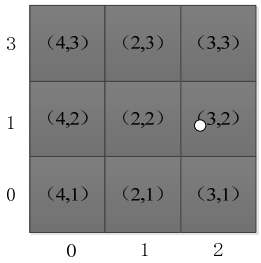

FIG.3. View point to the right

CONCLUSION

For the above scheduling method, this article compares it with the one-time loading method, analyzes the program memory occupancy and frame rate, as shown in Table 1. It can be seen that after the dynamic loading, the program performance is greatly improved. The algorithm is designed to make the scene run smoothly.

TABLE 1. Comparison of main performance of the two schemes

Index	One-time loading	Dynamic loading
FPS	34.3 fps	68.6 fps
Total memory	272.7 M	108.5 M

ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of Guangdong Province China (Grant No.2015A030310112) and the Science and Technology Project Guangdong Provincial China (Grant No.2016A040403110). Corresponding author is Lihong Luo.

REFERENCES

1. Lindstrom P, Koller D, Ribarsky W, et al. Real-time, continuous level of detail rendering of height fields[C]// Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 1996:109-118.
2. Hoppe H. View-dependent refinement of progressive meshes[C]// Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co. 1997:189-198.
3. Hongmei Sun. The Research and Implementation of Real-time Rendering Techniques in Distributed Virtual Scene[D]. Graduate University of Chinese Academy of Sciences (Institute of Computing Technology), 2001.
4. Xingquan Liu, Zhihua Tang, Liming Wang, et al. Research on Virtools Software and Geospatial Data Integration Technology[J]. Journal of System Simulation, 2009, 21(10):2971-2976.
5. Yong Tang, Dongliang Guo, Mengya Lv. Research about Large Scale Terrain for View-dependent Out-of-core Visualization[J]. Journal of System Simulation. 2009,21(8).2424-2427.
6. Zhaoting Ma, Mao Pan, Jinxing Hu, et al. A Fast Walkthrough Method for Massive Terrain Based on Data Block Partition[J]. Journal of Peking University (Natural Science Edition), 2004, 40(4):619-625.
7. Unity Technologies. Unity 5.X From Entry to Master[M]. 1st Edition. Beijing: China Railway Publishing House,2013.