Research and Practice of Key Technologies in the Development of Virtual Driving System

Jin-song FAN¹, Lv-jie SHE²,* and Ming-liang CAO³

¹School of Industrial Design & Ceramic Art, Foshan University, Foshan, Guangdong, 528000, China
²School of Mechanical & Electrical Engineering, Foshan University, Foshan, Guangdong, 528000, China
³Guangdong Academy of Research on Virtual Reality Industry, Foshan University, Foshan, Guangdong, 528000, China

*Corresponding author

Keywords: Virtual driving system, Model simplification, Collision detection, Object pool.

Abstract. Unity3D has been a popular 3D game development engine in recent years, and it is also used in the field of virtual roaming, virtual education, virtual training, etc. This paper mainly expounds the technical characteristics and design method of developing virtual driving system with Unity3D, and realizes the goal of driving skill practice through modular design. In order to get better virtual driving experience, the principle and technology of object pool are investigated and utilized in order to reduce the cost of resource creation and revocation, so as to improve the overall performance of the system.

Introduction

The digital age has a profound influence on human psychology and between man and machine. In recent years, with the development of computer technology, the application of virtual reality has been gradually improved. In the virtual driving training field, compared with the traditional way of watching training videos, the virtual reality technology with the characteristics of immersion, interaction and imagination allows users to freely explore and interactive with a simulation environment in real time, which help to get better coaching.

At present, China's private car ownership has increased rapidly, the number of people who is going to learn how to drive is also increasing, the traditional vehicle driving training approach is limited by time, space, resources and other factors, resulting in low efficiency and poor effect, etc. The developed virtual driving system is characterized by the combination of technology and aesthetics, which gives people a sense of immersion and can be used in virtual driving training to achieve the goal of attracting users to learn and enhancing the safety and effectiveness of training.

Virtual Driving System and Evaluation Optimization

Build a computer-generated virtual driving system, is not only to make the user's senses (including visual, auditory, tactile and sense of gravity etc.) "immersed" in the virtual environment, but also to bring the user a good experience and exploration process, in order to achieve the goal of clarifying the space-time range of the story, explaining the story cultural background and enhancing the user's subjective experience, which is also the purpose of interaction and cognition.

The function of the virtual driving training system consists of three modules: model, interaction and evaluation. Among them, the model act as an interactive carrier, and the evaluation is the result of interaction. Therefore, the three subsystems are interdependent. The objective of the training system is to develop good driving habits through interactive learning with virtual driving. The development of the virtual driving training needs to provide attractive environment and interactive settings in scene which can not only bring sensory experience but also to ensure the stability and real-time feedback of the performance of computer. So in the evaluation module, it is needed to avoid repeated information...
promotion brought by the specific memory management problems which led to reduced user acceptance.

![Virtual driving system](image)

Figure 1. System module.

**Model Subsystem**

In virtual driving system, it is necessary to have a realistic model to build a lifelike virtual scene, and a realistic scene model should have the following characteristics:

(i) Diversity and rationality

The diversity and rationality of the model are conducted on the base of data collection. After the preliminary data collection, the real-site pictures were taken as reference, and the three-dimensional model library and environment were established.

(ii) The hierarchy structure of planning

First, the model is divided into subject and object categories. The perspective of the first person controller is act as the main viewport, which meets the requirement of output scene covering the windshield in front of the car body model. Models other than cars are defined as object models, such as training grounds, floors, fences, surrounding buildings, vegetation, streetlights, benches, etc. Finally, the training field is placed in the center, divergence the layout, and the object model is arranged reasonably according to high and low.

(iii) The fluency of equipment running

Considering the fluency of the equipment, it is necessary to optimize model. The optimization of the model can be summarized as the simplification of the topology of 3D model while maintaining the fidelity of a model to a real object. Therefore, according to the category of the model as well as the distance between the object model and the subject model, the corresponding model simplification is carried out to reduce the time of computer drawing and save storage space.

**Interaction Subsystem**

The interaction subsystem is guided by the training target, and the human-computer interaction is carried out by setting up the main menu, sub menu and pop-up options. The script is used to get collision detection and information feedback between models in the scene.

**Evaluation Subsystem**

As a part of the virtual driving system, which is closely related to user operations and frequently invoked, the evaluation subsystem is the core of the system functionality. Because the memory management cannot be entrusted to the common language runtime, aiming at the problem of resources wasting and efficiency of CPU, the evaluation optimization scheme was given, object pool technology is studied and implemented to promote the system performance.

**Analysis and Research of Key Technologies**

According to the modular design method, the key technologies involved in virtual driving system (Figure 2) are model simplification technology used in model optimization, collision detection technology and object pool technology used in script programming.
Model Simplification Technique

Model simplification technology is using algorithms to reduce the amount of model data. In the premise of maintaining the original model’s basic visual features, minimizes the number of triangular patches and vertexes of the original model, in order to reduce the segmentation accuracy of the model surface. In 1992, Schroeder[1] proposed grid simplification method based on vertex decimation method. After that the edge collapse method, triangle collapse and other methods have been proposed, in which the edge collapse is the most widely used simplified method.

Taking Hoppe’s Progressive Meshes algorithm as an example, it first simplifies the model by using edge collapse, as the edge is deleted by selecting a smaller cost, according to the topological relationship of the deleted edge continuation to a new point, thereby generating a base grid and a series of vertex split operations, then through the vertex split operations, gradually add the details to the simplified grids. Finally, the grid model of continuous detail level is obtained[2]. Garland and Heckbert[3] proposed the mesh simplification algorithm based on quadric error metrics. By a sequence of edge collapse operations, with two error metric to characterize the error caused by every vertex after moving, the implementation effect is simple and effective[4].

Therefore, the model simplification technique can be used to simplify the triangular number of grid models in the model library. In turn, a single primitive model in the model library is simplified and then exported as the .FBX format and imported into the Unity3D.

Collision Detection Technology

Collision detection (CD) as a key component in the virtual reality system, the main task is to determine whether there is a collision between the model and the scene, and given the collision position, puncture depth information[5]. In the collision detection algorithm, the bounding box collision detection algorithm does not detect the objects in the scene, but only tests the bounding geometry, which reduces the computation of the specific intersection test. A good bounding geometry should have the following characteristics: compact fitting, small computation, easy reconstruction and small memory occupation.
The common bounding box algorithms include Sphere, Axis-aligned bounding box (AABB), Oriented bounding box (OBB) and k-DOP bounding box.

![Four common types of bounding boxes.](image)

Figure 5. Four common types of bounding boxes.

Among them, AABB bounding box is one of the common collision detection methods in virtual reality applications, AABB bounding box is the smallest regular hexahedron which is along the direction of the three-dimensional coordinate axis. For close-distance rigid body collision detection, AABB can complete rapid intersection test, and after dynamic reconfiguration (translation or rotation), it is still relatively simple to recalculate the maximum value and the minimum value on the axis of X, Y, Z. Therefore, in case of in a good bounding fitting situation the bounding box collision detection method based on AABB is simple and effective.

Object Pooling

The object pool is a data structure that holds multiple instances of the same object\[^{[6]}\], the design pattern is adopted to create a number of objects in advance into the pool. When the system is called, the object pool will allocate the resource to use, and when the resource is used up, the object pool will recycle these objects, and can be assigned to other threads. In the game object creation, database link, public information service platform, which were involved large data sharing and security but limited the use of resources in the application, pooling technology as a caching mechanism for managing and sharing system resources, can effectively play a protective role.

In Unity3D, the programming language C# is used in labeling method to reclaim memory. First making a mark on all objects by reachability analysis, to determine whether an object is referenced by GC (Garbage Collection) roots to determine if it is recovered. The type of GC Roots are variables on the thread stack, static variables, variables in the register, and so forth. As in the object graph (Figure 6), after the implementation of function, No. 3 stack goes out of scope, there is no root reference, a, b, c object between the reference code will also be treated as waste recycling, thereby reducing pressure heap memory allocation.

![Objects and its reference.](image)

Figure 6. Objects and its reference.

Similarly, as for the node creation (Instantiate) and destruction (Destroy) operations in the evaluation system, memory is constantly allocated and released, in order to reduce the CPU usage, object pool technology can be used to optimize.

Solution

The main task of the virtual driving evaluation system is to get the real-time information feedback. That is, in the process of the user’s driving, the collision detection information can immediately pass to the UI thread, and then make the user interface display and close.

The specific design is as follows:

(i) Creating an object pool
Firstly, the object pool is built in singleton mode for managing a collection of alternative objects. Singleton patterns are instances of a class with only one object (component) in the scene, which is
responsible for creating its own object, while ensuring that only a single object is created. Then, six different sub-nodes are created according to the induction of the driving rules for different scenarios.

(ii) Configure the life cycle of the object pool

The lifecycle of object pool management is roughly divided into three phases:
Firstly, the object can be accessed by using instance method in the initialization phase.

```csharp
void Awake()
{
    if (instance == null)
        instance = this;
    else if (instance != this)
        Destroy(gameObject);
}
```

Secondly, when an instance is enabled, use the array and store it in dictionary.

```csharp
void Start()
{
    _pools = new Dictionary<string, ObjectPool> ();
    for (int i = 0; i< pools.Length; i++)
    {
        _pools.Add (pools[i].pooledObject.name, pools[i]);
    }
}
```

Lastly, the activated object is extracted from the pool and returned when deactivated.

```csharp
if (activate)
    obj.SetActive (true);
return obj;
```

(iii) Replacement and reset

Finally, replacement strategy can be used when many objects in the pool are rarely used. However, there is no other process waiting to release the memory, the replacement strategy can be delayed or undo.

Establishment and Implementation of the System

After the research and analysis of the key technologies of the development of virtual driving system, according to the modular design method, the three subsystems which contain the model, interaction and evaluation are studied and implemented respectively. First, build the simplified model library and then import them into the scene. Secondly write collision detection scripts for collided models, after the realization of the basic interactive functions, focusing on analyzing and optimizing the evaluation system.

Interaction Design and Implementation

In order to achieve the training goal of the virtual driving system—to help the user learn and understand the basic requirements and contents of the field training. In this design, the information interaction consists of prompt, warning and hit point. After reading the field training rules, if the user stopped half-way, then warning and deducted 5 points; if the user failed to meet the requirement of lighting operations or arrive at the specified place, then warning and deducted 10 points. Others like compaction lines, unfinished items, etc. are warned and deducted 100 points.
Through HP (hit point) to show the changes in the hit point, the method is to write the parameters of HP and collision events into the script of the subject model, and display them on the screen. Secondly, `SendMessage` is used to transmit information. They are separately bound to the objects labeled (tag) as ‘Player’ and ‘Obt’, and run. Every time the specified objects collide, the `OnCollisionEnter` function would be triggered, and then the $HP-=10$ will be passed to the `getMessage` method of the UI blood bar and text to be displayed in a timely manner.

```csharp
void OnCollisionEnter(Collision col)
{
    if (col.gameObject.tag == "Obt")
    {
        HP -= 10;
        if (HP < 0)
        {
            HP = 0;
            Debug.Log("Dead!");
        }
        UpdatedHP();
    }
}
```

![Figure 9. Code segment](image)

**Evaluation Design and Implementation**

First, create a script called Object Pool Manager, which acts as a pool for storing and managing different information interface objects, and caching by acquiring objects and recovering objects.

```csharp
GameObject obj = _pools[ID].GetObject();
```

```csharp
public void Recycle(GameObject obj)
{
    Obj.SetActive(false);
}
```

The object replacement means that when the number of objects in the object pool reaches the upper bound, selecting a remove out from the object pool to add a new object.[7]

In the evaluation system, users complete the field driving project according to a certain order, so according to the time sequence, the first accessed data should be used least so it can be eliminated by priority. And if the user does not end the virtual driving, collision events in the scene will lead to the occurrence of information interface, so the first accessed objects are put at the bottom of the accessed array in order to be eliminated. The program design of the object pool is as follows:

```csharp
public class NewPool(K, V)
{
    IDictionary<K, V> _dictionary;
    LinkedList<K> _linkedList;
    public NewPool(int maxsize)
    {
        _maxsize = maxsize > 0?maxsize: size;
    }
```
Next, use \texttt{set (key, value)} and \texttt{get (key)} to determine whether the object is in the pool or not, delete the bottom object and add a new \texttt{(key, value)}, and reset the value if it exists. The specific definition of the class is as follows:
\begin{verbatim}
    _dictionary[key] = value;
    _linkedList.Remove (key);
    _linkedList.AddFirst (key);
    If (_linkedList.Count > _maxsize) {
        _dictionary.Remove(_linkedList.Last.Value); _linkedList.RemoveLast ();
    }
\end{verbatim}

Finally, use \texttt{void (Reset)} can reset the state when the object is reclaimed. In summary, in the development environment: the operating system--Microsoft Windows10; the development platform--Microsoft Visual Studio 2017; the programming language--C#, using object pooling principle to implement recycling a small number of nodes in the scene in Unity3D, saves system resources.

**Conclusions**

In this paper, the key technologies for the development of the virtual driving system are studied and analyzed, and a method of object pool technology is applied to optimize the performance of the system. This study provides a good reference for the development and realization of similar systems.

**Acknowledgement**

This work is supported by the Characteristics and Innovation Project of Guangdong University from Department of Education of Guangdong, China (No. 2016KTSCX148), and the 2016 Foshan Science and technology innovation project from the Science and Technology Bureau of Foshan, Guangdong, China (No. 2016AG100321).

**References**


