Unity Application of Technology in Game Development

Haoyu Zhu
Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

Abstract: Unity is a 3D game development tool and game engine suite, which includes graphics, audio, physics, network and other engine support, and has a very powerful editor to integrate. This paper discusses the application of Unity technology in game development, including the features and functions of engine technology, interface and operation, collision technology and the overall system framework, in-depth research on game development technology, and looks forward to more excellent games.

Keywords: Unity Technology; Game Development; Engine; System

1. Introduction

With the progress of science and technology, the computer hardware and software have been greatly improved, and the quality of the games published in the world is also constantly improving every year, including the graphics, gameplay, sound and story effects. Especially in terms of the screen effect, under the premise of the current technology development, it gives the player the ultimate experience. The game industry generated $137.9 billion in revenue in 2018, according to a leading gaming and media market intelligence provider. In 2022, the game industry generated $196 billion in revenue. The game industry continues to break through technological boundaries, stimulate innovative thinking, and promote the development of software and hardware technology in the entire electronic industry. Major Internet companies have also been involved in the game industry, hoping to provide services for the world players and get a share of this big market. Thus, the development of the game industry is still on the rise. Exquisite game graphics, innovative gameplay can certainly win the love of players, but in addition to these, what direction can we work towards? In recent years, in order to meet the requirements of major game manufacturers for the machine performance, the hardware technology of both computers and mobile computers has been greatly improved, so that the artificial intelligence module originally at the end of the game can get more computing power.

2. Features of the Unity Engine

Unity3D Is a cross-platform game engine developed by the Unity Technologies Company. Games developed through this engine can be easily released to a number of platforms, such as Windows, MacOS, Linux, Android, iOS, Switch, and Xbox. Later, it was also extended to the HTML5 web platform based on WebGL technology. Unity3D Not satisfied with the development of the game industry, in the construction industry visualization and 3 D animation production has a wide range of applications. Unity Was born in 2005, announced in apple's global developers conference, opening to the outside world, as a rising star, Unity3D engine after years of technology iteration accumulation, actively pursues unreal engine 3 a game development engine, has developed into a full function of the market share, in the world's first, using its game development personnel proportion is also very high. Unity is an efficient cross-platform professional game engine developed by Unity Technologies. Users can easily realize various game creativity, architectural visualization and 3D interactive development to create wonderful 3 D and 2D games; one development can be easily deployed to Windows, Mac, IOS, Android and other platforms, and can also share and download related game resources on Asset Store. /Unity is an internationally leading professional game engine, which is favored by developers mainly because it has some advantages of its own. First, it is a highly integrated and scalable editor, that means the Unity editor is powerful, and it is easy to use; in the editor, you can adjust the game's terrain, lighting, animation, models, materials, physics, and other parameters, And it can also easily use third-party plug-ins.The scripting languages supported in Unity are also simple and easy to use, can be developed in both the C # and javascript languages.

3. Unity Interface and Operation

Download and install Unity3D, the first opening is the default project, if you use the trial version is opening, it is an empty project, need to create a new project, choose File-> New Project to create a new game project, and then before starting, let's determine the screen layout in 2 Split mode. Choosing Window-> Layouts-> 2 Split such view is now divided into five major regions,. The five main regions are respectively:

(1) Scene view : It is used to display the game objects.

(2) Game view : Show what the game looks like after running.

(3) Hierarchical View: List all those used in the current scene view the game object.

(4) Project View : displays a list of all available resources under the resource directory.

(5) Inspector View: displays the attribute information of the selected game object.

Start Unity, and when you enter the project we built, you will see a variety of game objects in the game scene. The start scene of the new project contains only one main camera, and you can see this unique item in the hierarchy view hierarchy. This trick can be used to find any game objects you created or imported in the scene.

To add some game objects to the scene, they can be created with the Unity menu Game Object-> Create Other or imported from outside. Import game objects from the outside, first to the project view Project, and then drag them directly to the scene or hierarchy. Creating game objects with Unity Game Object, such as selecting Game Object-> Create Other-> Plane, can create a plane in the scene for the characters to walk around. Add a cube to the game scene, select Game Object-> Create Other-> Cube.

Add a point light source and select Game Object-> Create Other-> Point Light.

You can view a 3D model from any angle in the following way:

Rotation: Alt + left mouse button

Pan in: Alt + middle mouse button

Scale: Alt + right mouse button

Or you can also choose the hand tool in the upper left corner of the Unity, select the hand tool, the left mouse key translation, the right mouse rotation, the mouse middle key zoom.

Computer games are made up of three-dimensional models, textures, sound files, the game code, etc., Which is known as the game resource assets. Then changes in the game view, because the first person controlsler preset contains their cameras, will replace the main camera field of vision, to ensure that the first person controller is not below the terrain plane, then delete the main camera, click the play button to start the game, so you can be in the game view by the cursor keys or W, S, A, D control the first person walking in the scene, look around with the mouse, press the space jump. Click the play button again to stop the game from running.

Copying game objects are a powerful function, and when copying an object, all of its functions and behaviors will also be copied, which is a very quick way to create multiple identical objects in a scene. The way of copying is to select the game object to copy in the hierarchy view, and then press Ctrl + D, or also from the Edit menu. Note the same game object in the hierarchy view, but you can not find it in the scene view, because it overlaps with the original object in the scene, just use the mobile tool to move one to the right location.

By clicking on any game object, you can see the additional components in the review view, such as Transform, Box Collider and so on. We can add appropriate components to the game object through the Unity menu. For example, when we import models from the outside, we generally cannot produce collisions, because it has no physical properties, and we can add a rigid body (Rigidbody) component to it through Unity. Rigidbody The component allows the game object to have a gravity, in which the object drops down to meet the object containing a collision (collider) component. Operation method is: select the game object, select Component->Physics->Rigidbody.

4. Common-used Collision Detection Technology in Game Development

In game development, we often need to implement this function: whether the bullet hits the enemy in the game, whether the character picks up the equipment, and so on. Each of these features requires one of the technology in the Unity engine: collision detection. According to different applicable scenarios, we can divide collisions detection into three types: collision between actual game objects, trigger collision, and collision between pre-detected colliders with emission rays. Here are several specific games as examples to explain the use of these collision detection.

4.1 Basic collision detection

The two game objects in a direct collision have collider components, and at least one game object has a rigid body (Rigidbody) component. According to the different shapes of the game object, we can choose to add different shapes of the collision body.This collision detection implementation is achieved in the following steps:

4.1.1 Add collision components

Select the game object, select the menu bar Component-> Physics command, and select different collision types according to the shape of the game object.

4.1.2 Add the rigid body component

Select the game object and select the menu bar Component->Physics-> Rigidbody command to add a rigid body component to the game object. Attributes such as Mass (mass), Drag (resistance), and Use Gravity (using gravity) can be set through the Inspector view window.

4.1.3 Add the C # script

The C # script can also be added to the game object as a component. We create a new C # file and drag it to the game object. We can also add the C # component directly to the Inspector view window, and then add the collision detection method to the C # editor: MonoBehaviour. On Collision ; when two games objects contact, execute On Collision Enter method, such as 3D puzzle game "Jump Knight" game, when the little knight jumps down like the way.This method is executed automatically, when the device or trap is in contact. According to the collision information, we can obtain the collision game objects, and then we can conduct the corresponding setting, such as adding a shield to the little knight, reducing the health volume and so on.

4.2 Trigger collision detection

The difference between the trigger collision detection and the basic collision detection is that when the game object involved in the collision does not produce a physical collision effect; whether to detect the game object collision but not to affect the movement of the object, or when the game object enters a certain area. Trigger collision detection implements the same steps as basic collision detection, except that the Is Trigger option is checked when adding the collider component to the game object.

4.3 Ray collision detection

A ray projection is a non-ending ray in one direction centered on a point in the scene. During the launch, it stops launching if there is a collision with other game objects. Ray-like collision detection requires the method of class Physics: Physics.Raycast (transform. position, transform. forward, hit,distance);position is the position of the ray emission, forward is the angle of the ray emission, distance is the distance of the ray, the collision is only detected when the character and the game object is less than distance, and hit will store the information of the game object where the collision occurs.

5. Overall system Architecture of Unity Games

The Unity engine layer at the bottom provides a variety of basic functions needed during the game production process, such as graphics rendering, animation management, physical systems, and particle effects. Each is a necessary module for a mature game engine. And the reason for choosing the Unity game development engine has been discussed in the previous article, and will not be repeated here. The plug-in layer, located above the engine layer, exists because the author wants to speed up the development of the project and make the game perform well. Specifically, the UI system is used as the development of UI module in the whole game. The superior performance of UGUI plug-in has long been recognized by the industry, and it is very convenient to make the adaptation of games. The ML-Agents plug-in is designed to design and implement an in-game machine learning NPC.It has already been introduced in the article. The basic operation control of the player in the play, based on this basis to do secondary development.

Management controls the development of the game, the resources and the number and difficulty of NPCs. In the engine layer, developers use various API interfaces to complete the implementation of various small modules, while developers' resource management and game management need to be implemented independently.

The logical control layer is the core of the game, which contains the logical control of the entire game. Most of the work will be gathered here, which is also the key to achieve different types of games on the basis of the same engine layer and plug-in layer. The accompanying alarm generated. The NPC logic control section is responsible for writing various NPC behaviors, including patrol, attack and escape, and even the overall behavioral decisions. Level logic control is responsible for the beginning of the game, level switching, and the end of the game.

6. Conclusion

With the development of science and technology in recent years, the performance of computer software and hardware has been greatly improved, and the research on game artificial intelligence has begun to be paid attention to by people. As an important part of the game, game AI has always been playing a role in improving the player experience. This paper focuses on the design of games with more anthropomorphic perception and flexible behavior, and develops a game based on the Unity3D game development engine. A good game AI should also establish an emotional bond with the player to become the player's company and give an immersive experience.

References

[1] Liu DS. Research on infrared simulation and related technology based on Unity engine. Beijing Jiaotong University, 2016.

[2] Xuan YS. Unity3D Game development [M]. Beijing; People's post out of our society. 2012.

[3] Hu WF. Zhang Q. Mao YQ. Component-based hierarchical statemachine — A reusable and flexible game AI technology[C]. International Information Technology and Artificial Intelligence Conference, IEEE, 2011:20-22.

[4] Leandro GM. Alvim; Adriano Joaquim de Oliveira Cruz. A Fuzzy State Machine applied to an emotion model for electronic game characters[C]. International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), IEEE, 2008:1-6.