Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 1041-1044
doi:10.4028/www.scientific.net/AEF.6-7.1041

The 3D Computer Game Design using Java 3D

Xinrui Gao, Deqing Zhao, Pingping Yao

Information Science Technology Institute of Hainan University, Haikou 570228, China

xr_gao2002@yahoo.com.cn

Keywords: 3D game, 3D models, interaction, Java 3D

Abstract. In this thesis, the two 3D computer games were designed by Java 3D. In the first game, by using 3DMax, one aircraft carrier, one helicopter, submarines, warships, and destroyer escorts etc were designed. By using Loader3DS of MicrocrowLoader in Loader3DS1_2.jar or ModelLoader of Java 3D, these 3D models are loaded into the Java 3D. The second 3D game displays the 3D fighting and collision of tanks. This function is very important for the 3D game design. By using textures, the different appearances of 3D models were designed. By using Java 3D classes such as Alpha and PositionPathInterpolator and Sounds etc, the motions of 3D models and interactions and the sounds were defined.

Introduction

Java 3D is a kind of API (Application Programming Interface) of Java language. It has a plenty of modeling functions which include basic 3D modeling, scene graph structure, 3D texturing, appearance and materials, behavior and interaction, and animation. So, it is a kind of programming tool to design computer 3D games. But Java 3D has limited functions in designing complex 3D models used in 3D games. So we need to use 3Ds Max or other 3D design software to design this kind of complex 3D models and transform them to Java 3D.

In this thesis, two examples of 3D game design were given. One includes one aircraft carrier, one helicopter, submarines, warships, and destroyer escorts which were designed by 3Ds max. The submarines could attack the enemy's aircraft carrier, submarines, anti submarine warship, and warships with their torpedoes controlled by keyboard or mouse. The other 3D game is about the fighting among tanks. All functions are implemented by Java and Java 3D.

3D model design in the first game

In this game, because of their complex shapes and complex appearances, the aircraft carrier, the helicopter, submarines, warships, destroyer escorts, and surveillance and control platform etc were designed by 3Ds Max. They are loaded into the Java 3D by the class Loader3DS of MicrocrowLoader in Loader3DS1_2.jar which was downloaded from www.j3d.org Web or ModelLoader of Java 3D.

The sea is defined by a huge box which is transparent and textured by the sea appearance. The far blue sky is textured by the blue sky appearance. The surveillance and control platform has its specific appearance. By using textures, the execution of the Java 3D game could be accelerated and needs less computer memory. The class TextureLoader of Java 3D loads the texture pictures into Java 3D and the class TexPlane maps the textures on the 3D models. Fig 1 is the surveillance and control platform. Fig 2 includes one aircraft carrier and one helicopter. Fig 3 includes one aircraft carrier, one helicopter, the sea water, and the blue sky. Fig 4 is the view which is watched from the under water.

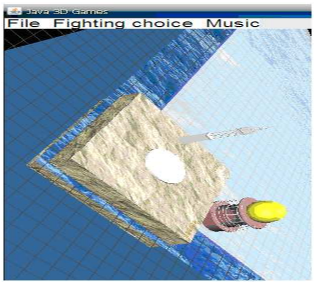

Fig 1. Surveillance and control platform

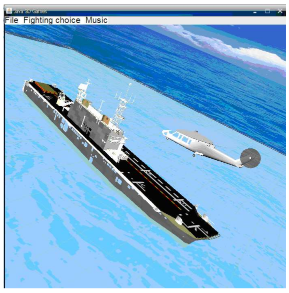

Fig 2. The aircraft carrier and the helicopter

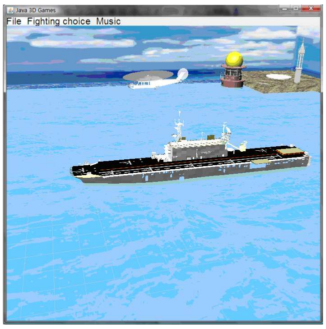

Fig 3. Aircraft carrier, helicopter and platform

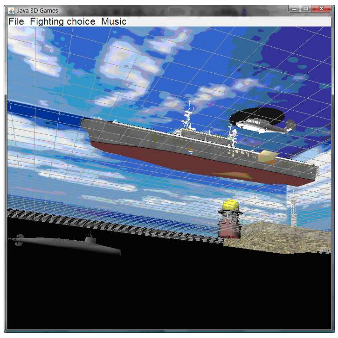

Fig 4.The view from the under water

The movements design of 3D models in the first game

For the helicopter, its flight is defined by Java 3D Classes such as Alpha, PositionPathInterpolator, and DistanceLOD etc. The classes Alpha and PositionPathInterpolator define the 3D models motions. The LOD is the level of details which defines the effect of a model from near to far. If a model is very far from us, its details is very vague and if it is very near, its details is very clear. The class DistanceLOD has this function. The helicopter flies by some directions. When it is flying, its rotating wings are rotating and it becomes smaller from near to far.

Behaviors, interactions, and animation

The movements of all moving 3D models are defined and these movements could be controlled by mouse or keyboard. The movements of all moving 3D models and the launch of torpedoes from submarines are controlled by keyboard. The sub class of the class Behavior could rewrite its two methods to accomplish some tasks.

The class Behavior of Java 3D is an abstract class which includes two abstract methods such as void initialize() and void processStimulus(Enumeration wakeupCriteria). When we use the objects of the Behavior, we must call the method initialize(). In some specific wakeup conditions, the program of Java 3D call the method processStimulus(). The method, void wakeupOn(WakeupCondiotion wakeup), defines the wakeup conditions. When some wakeup conditions appear, the object of the class WakeupCondiotion call the method processStimulus() to wakeup the object of the class behavior and to accomplish some behaviors which were defined in the class Behavior. The method WakeupOnElapsedTime() controls the interval of the wakeup events. The method WakeupOnAWTEvent(KeyEvent.KEY_PRESSED) defines the keyboard events and the method WakeupOnAWTEvent(Event.MOUSE_DOWN) defines the mouse events. If we need several AWT events, we could use the "or" or "|" to combine them. For example, The method WakeupOnAWTEvent(AWTEvent.MOUSE_EVENT_MASK| AWTEvent.MOUSE_MOTION_EVENT_MASK) combines the mouse down event and the mouse motion event.

By using these methods and other Java 3D classes, we define the keyboard events and mouse events. By using the method WakeupOnElapsedTime(), we control the intervals of the launch of torpedoes. By using the method WakeupOnCollisionEntry （）, we could judge if a warship was hit by torpedoes. Before a warship will be hit by a torpedo, the warship could shun the torpedo by keyboard. We could combine different events.

The first 3D game design

This game includes 26 classes which define the aircraft carrier, one helicopter, submarines, warships, destroyer escorts, surveillance and control platform, the sea and the sky, the motions of 3D models, keyboard and mouse events, the collisions between warship and torpedoes, and 3D textures etc.

By using the class Sound, we could define different sounds. This game could calculate the fuel of submarines and warships, the number of launched torpedoes, and the number of the successful attacked warships and submarines etc. One submarine could attack several targets at the same time. Some functions of this game need to be enhanced.

The other 3D game designed using Java 3D

In this 3D game, we designed several tanks which include our tanks and enemy's tanks. Each tank could move, turn around, and fire under control of keyboard. Tanks, controlled by keyboards, can move anywhere and fire. These tanks and the grounds have their textures, for examples Fig 5, Fig 6, Fig 7. Each tank's fired shell has its direction and attacked tanks would explode. Our tanks and enemy tanks have different colors.

The class wakeupOnCollisionEntry specifies a wakeup when the specified object collides with any other object in the scene graph. The class wakeOnCollisionExit specifies a wakeup when the specified object no longer collides with any other object in the scene graph. In computing collisions, these classes use geometric bounds as an approximation or use geometry.

By using Java 3D classes wakeupOnCollisionEntry and wakeOnCollisionExit, we could detect and process the collisions among our tanks, enemy tanks, and fired shells. If some tanks were destroyed by shells, then the classes of these tanks and shells were terminated in this game and their 3D models disappeared in the scene.

In this game, we used Java 3D classes such as Behavior and its methods initialize() and processStimulus(Enumeration wakeupCriteria), WakeupCondiotion, wakeupOn(WakeupCondiotion wakeup), WakeupOnElapsedTime(), WakeupOnAWTEvent(KeyEvent.KEY_PRESSED), wakeupOnCollisionEntry, WakeupOnAWTEvent(Event.MOUSE_DOWN), and wakeOnCollisionExit, etc. Each function has its class.

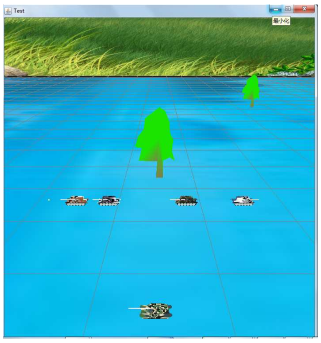

Fig 6. Tanks and their moving

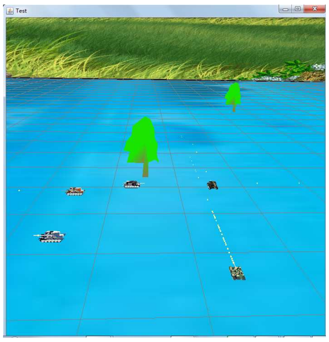

Fig 6. Tank's firing

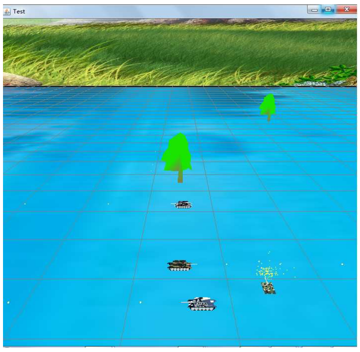

Fig 7. Attacked tank's explosion

Conclusions

According to these two 3D games designed by Java 3D, we concluded that Java 3D has powerful functions in 3D game design. By combining Java 3D classes, Java programming, and 3Ds Max etc, we could design complex 3D games. In Java 3D, it is not easy to design complex 3D models. So, in the first 3D game, we design complex models such as one aircraft carrier, one helicopter, submarines, warships, destroyer escorts, and surveillance and control platform etc in 3Ds Max, then transform them into Java 3D, and use them in Java 3D. The second 3D game displays the 3D fighting and collision. This function is very important for the 3D game design.

Acknowledgement

Acknowledgements for the support of Ph. D Scientific Research Foundation of Hainan University, China.

References:

[1] Java, Java 3D and Computer Geometry Design [M].Gao Xinrui. Publishing House of Electronics Industry (PHEI) of China.2007.8

[2] Computer Graphics using Java 3D and Java 3D [M]. Hong Zhang,Y. Daniel Liang. Prentice Hall .2006.12. New Jersey, USA.

[3] http://www.j3d.org/

[4] http://java.sun.com/developer/onlineTraining/java3d/

[5] http://www.javagaming.org