Research on Satellite Simulation for Mobile Terminals

Qi Su(✉), Xin Lin, and Qipeng Hu

School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
{suqi2014,lx}@buaa.edu.cn, w18679104034@126.com

Abstract. With the development of Internet+, developing an application of satellite simulation which executes on mobile terminals has an important significance for engineers to get visual knowledge of spaceflight. The application design and implementation method of satellite simulation software are discussed. A framework of satellite simulation system is presented, the modules and workflow of the satellite simulation system are described. Approaches to create the starry background of the satellite operation, environment lighting of the earth and the orbits of satellites in the software are presented in detail. The software uses Unity3D platform to create 3D simulation environment, uses SGP4/SDP4 orbit prediction models to calculate orbits of satellites, and uses databases to store TLE data.

Keywords: Mobile terminals · Satellite simulation · Unity3D

1 Introduction

The research of simulation is one of the effective ways to reduce the risk and cost of production. Since the beginning of the aerospace industry in 1950s, simulation technology has been widely used in aerospace engineering. Up to nowadays, the development of spaceflight simulation has experienced the following stages: physical simulation, analog machine simulation, digital machine simulation, multimedia simulation and visual interactive simulation, etc. In recent years, with the deepening of the scientific research and the breakthrough of the technology of computer hardware and software, the development and application of the simulation technology has been promoted to a new stage, and plays an important role for the development of space industry.

Nowadays, concept of "Internet+" affects deeply on technology research. Traditional industries are undergoing significant changes when fusing with the Internet technology. As intelligent mobile terminals has got a very big development on data storage, computing capability and the image processing ability, some spaceflight simulation and scenario demonstration executed on computers originally can be realized on mobile terminals. The research working on applying spaceflight simulation on mobile terminals has an important significance for aerospace engineering.

This paper researches application design and implementation method of the satellite simulation software. A framework of satellite simulation system is presented, modules and workflow of the satellite simulation system are described. Approaches to create the

L. Zhang et al. (Eds.): AsiaSim 2016/SCS AutumnSim 2016, Part II, CCIS 644, pp. 618–624, 2016.
DOI: 10.1007/978-981-10-2666-9_62

starry background of the satellite operation, environment lighting of the earth and the orbits of satellites in the software are presented in detail.

2 Application Design

The satellite simulation application system for mobile terminals uses Unity3D platform as the foundation of the development and implementation and is intend to be applied to Android platform. The Unity3D platform has cross-platform features. Applications developed on the Unity3D platform can be planted to multiple platforms, such as Android, iOS, Windows and so on [1]. Therefore, different from the general development of the Android applications, the design and implementation of the satellite simulation application system for mobile terminals are carried out according to the Unity3D platform [2, 3].

2.1 Framework Design

With consideration of maintainability, reusability and extensibility of the system, the satellite simulation application system for mobile terminals uses a four-layer framework which introduces a service layer into the classical three layers framework: presentation layer, business logic layer and data access layer (Fig. 1).

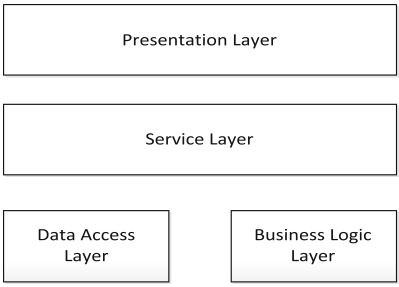

Fig. 1. The software framework

The functions of the four layers in the software framework include,

Presentation Layer: Gets input information and operations of user, and returns the specific business data to user. The purpose of the presentation layer is to accomplish interaction between the system and user.

Service Layer: Provides business logic entry to presentation layer, defines interface service forms and completes services by calling the interfaces. It also needs to manage interactive behaviors of all business logic objects.

Business Logic Layer: Accepts data transfer object (DTO) transferred through the service layer, processes the incoming DTO according to the business rules, returns the processed data to the service layer and provides business logic behaviors for business logic objects.

Data Access Layer: Provides access interfaces to local data and remote data.

The relationships among four layers are listed as below.

The presentation layer collects input information of user and transfers the information to service layer by calling the interfaces. The service layer analyzes the information to get required services of user. For services that service layer can react directly, such as the rotation and scaling of the scene, service layer provides them to presentation layer and responds to the user's operations. For services that business logic layer and data access layer need to participate in, the service layer gets relevant data from data access layer, or feeds back some related data to presentation layer directly, or transfers data to the business logic layer by DTOs to perform relevant business rules. The business logic layer processes DTOs according to the corresponding business rules and sends the processed information back to the service layer. The service layer processes the feedback data and updates data in the presentation layer, and then responds to user's operations.

2.2 Modules Design

The satellite simulation application system for mobile terminals is composed of three main modules, the satellite simulation environment module, the satellite orbit prediction module and the satellite simulation database module. The modules of the software are shown as Fig. 2.

(1) Satellite Simulation Environment Module
Satellite simulation 3D environment is the window that shows to user and matters a lot to user's experiences. To some extent, it is the most important part of the satellite simulation application system. The module is designed based on Unity3D platform [4]. It is composed of several modules: UI module, to get user's input data and show corresponding data of satellites to user; skybox module, to create starry background of 3D environment; lighting module, to create lighting effect of 3D environment; dynamic model load module, to load 3D models of satellites according to user's input; satellite orbit draw module, to show satellite orbit in the 3D environment; scene control module, to control the scene demonstration reacting to user's operation.

(2) Satellite Orbit Prediction Module
Satellite orbit prediction is the core function of the satellite simulation application. The calculation accuracy and speed of the orbit prediction algorithm have a direct impact on the accuracy of orbit simulation and experiences of the user. Because calculation accuracy is inversely proportional to calculation speed, the orbit prediction algorithm should take balance between these two indicators. The module is designed to adopt NORAD two-line element(TLE) data as the data input of orbit prediction, and SGP4/SDP4 orbit model, which can produce very accurate results when used with current NORAD TLE data, as the core algorithm solution to calculate satellite position and velocity in earth inertial coordinate system [5].

(3) Satellite Simulation Database Module
The satellite simulation databases are an important part of satellite simulation application system for mobile terminals. The databases store satellite orbit data, two-line element(TLE) data, and provide data support for satellite orbit prediction.

The module is designed to develop three databases: SQLite database, LAN database, and Cloud database. The SQLite database is designed to install on the mobile terminals. When using SQLite database, users can accomplish satellite simulation within mobile terminals independently, and there is no need to connect internet. The LAN database is designed to be laid in the LAN. When using LAN database, users need to connect to the database through WLAN to get orbit data and accomplish satellite simulation. The Cloud database is designed to be laid in the Cloud. When using Cloud database, users need to connect to the database through Internet and accomplish the satellite simulation dependent on the orbit data stored in the LeanCloud database.

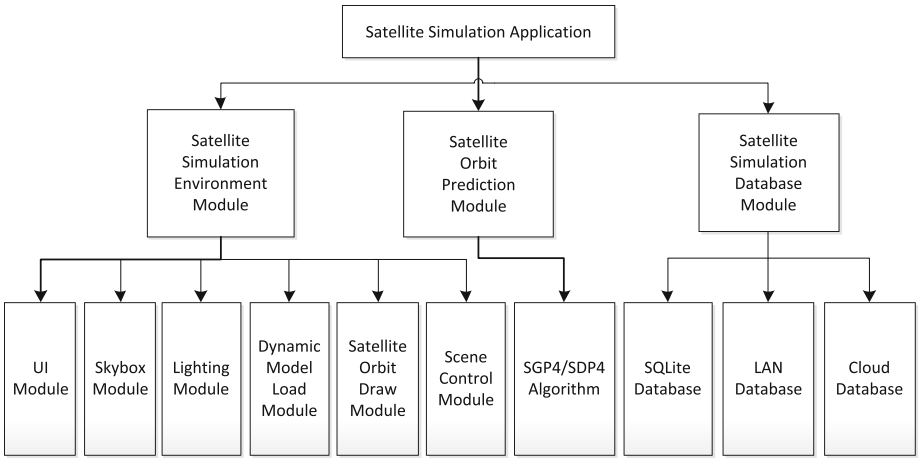

Fig. 2. The software modules

All above three modules are dependent on each other and work together closely to make sure that satellite simulation application system for mobile terminals executes correctly.

3 Application Implementation

The satellite simulation application system for mobile terminals is developed on Unity3D platform. Unity3D is developed by Unity Technologies and allows the user to easily create 3D visual simulation, architecture visualization, real-time 3D animation and other types of interactive content. It is a multi-platform integrated virtual reality development tools and a fully integrated virtual reality engine. The visual editor features of Unity3D are what you see is what you get. The developer can create the relevant scenes according to the real needs [6].

3.1 Starry Background

The starry background is developed on the basis of a component called *Skyboxes*. The thought of the skyboxes is putting the whole scene in a large cube and every plane of the cube is a square with the texture mapping. When the scene camera is put into the large cube, the scope of the field looks like in the real scene environment.

Methods as shown below are adopted to create the starry background,

(1) Get the coordinates of celestial body at a given moment through the ephemeris and map the coordinate data in the universe to the surface of a celestial sphere;
(2) Set a viewpoint at the center of the celestial sphere and divide what is seen at the viewpoint into six same-size seamless images;
(3) Import the prepared starry background images into Unity3D platform as assets which are used to construct starry background and change the images' properties to meet the needs of starry background to be constructed.
(4) Create a *material* in the Unity3D platform and set the material created in the *skybox* mode. Add reference assets to the skybox cube material at the right position and render the starry background environment in 3D scene.

3.2 Scene Illumination

Lighting is an important part of a scene and determines the color and atmosphere of 3D environment. There are four standard types of light source: directional light, point light, spot light and area light. Each light source has its unique properties, and developers can adjust position and direction of the light to construct an actual scene illumination.

The scene illumination is mainly developed to simulate the diurnal variation of the lighting on the Earth's surface. When satellites operate in the 3D environment, use scene lighting to reflect the satellites are operating in the sunlit side or the sunless side of the Earth. Generally the light from sun to earth is treated as parallel light. Considering that the directional light is close to parallel light, use the directional light to simulate the sun light and set the origin position of the light to infinity and the direction of the light orientation to the Earth in the scene. Add the sun lens to the light to simulate the sun and use the technology of dynamic shadows in the scene to realize the sunlit side and the sunless side on the Earth's surface (Fig. 3).

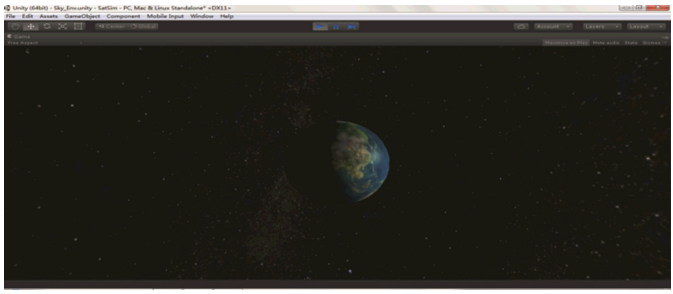

Fig. 3. The simulation environment

3.3 Scene Control

The application development based on Unity3D platform is using object-oriented method and modular modeling method. Every *GameObject* created in Unity3D platform has at least one *Transform* module. Through the *Transform* module, position and posture of the entity model in 3D environment can be obtained, changed and updated. If *Rigidbody* or *Collide*r module was added to a gameobject, the gameobject could support rigid body features or collision detection.

Entity model features in 3D environment are organized by modules. Through modules, develop the scene control effect in 3D scene. Create an EmptyObject at the original point of the scene and drag the scene mainCamera object onto it in the Hierarchy View to create a Parent-Child relationship between the two objects. On the basis of the Parent-Child relationship, update the posture of the EmptyObject to realize the scene rotation control. Based on the field of view of the cameras, develop the scene scaling control. The field of view for the cameras represent the scope that the cameras can see. The larger the field of view, the larger the area of the scene the observer can see. At the same time, the smaller the object is in the scene. When *Transform* module added to the Camera gameobject, we realize the rotation and scaling of the scene, changing viewpoints and fields scopes. When *Transform* module added to satellite gameobjects, we realize the simulation of satellite operation in 3D scene.

3.4 Model Loading

For the 3D models created out of Unity3D platform, when they are imported into Unity3D platform, the engine will automatically create Prefabs for them. The prefabs, similar to the public class in object-oriented languages, can be loaded directly into different scenarios in the project. Using the prefab in project development can improve the efficiency of scene management and production.

Make 3D satellite models outside Unity3D platform and transform them to the format that Unity3D supports. Import satellite models into Unity3D and organize them in the form of prefab. If need to load a satellite model into the 3D scene, Unity3D engine would create an instance of the prefab assets. Most of work that we should do is writing scripts to control the process of instantiation.

3.5 Orbit Drawing

We choose SGP4/SDP4 orbital model as the core algorithm solution to calculates satellite location and velocity in the earth orbit [7]. In accordance with the SGP4/SDP4 model published by NORAD, program and realize it in the project in C/C++ programming language. To use it in the mobile terminals, compile the model as. so shared library and use the orbit model to produce the latitude, longitude and altitude of the satellites in the ECI coordinate system and transform the LLA data to the coordinate data of the scene coordinate system. Use a structure array to store the solution result and transmit the structure array to a third-party plug-in which is compatible for Unity3D platform to draw the orbit of the satellite operating in the scene (Fig. 4).

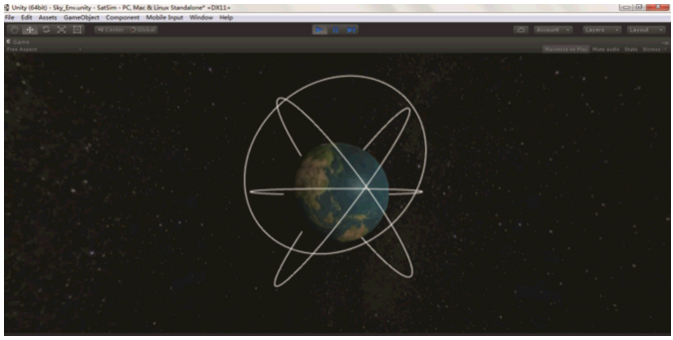

Fig. 4. Satellite simulation scene

4 Conclusions and Future Work

We study the framework and the modules of the satellite simulation application system for mobile terminals and elaborate the implementation method of some key modules. This paper is an explore for the spaceflight simulation carried out based on mobile terminals and make sure that it is possible for more complex spaceflight simulation task to be developed for mobile terminals. There are a lot of work to do in the future.

References

1. Zhu, H.-J.: Virtual roaming system based on Unity3D. Comput. Syst. Appl. **21**(10), 36–39 (2012)
2. Kim, S.L.: Research on the 3D game scene optimization of mobile phone based on the unity 3D engine. In: 2011 International Conference on Computational and Information Sciences (ICCIS), pp. 875–877. IEEE (2011)
3. Stamford, J.: Design and implementation of virtual interactive scene based on unity 3D. Adv. Mater. Res. **317**, 2162–2167 (2011)
4. Jin, L.: Research on visual simulation system of guided bomb. North University of China, May 2015
5. Liu, Y.-F.: Research on low earth orbit spacecraft orbit prediction strategy based on SGP 4 model. Harbin Institute of Technology, June 2009
6. Blackman, S.: Beginning 3D Game Development with Unity: All-in-One. Multi-Platform Game Development. Tsinghua Publishing, China (2015)
7. Diao, N.-H., Liu, J.-Q., Sun, C.-R., Meng, P.: Satellite orbit calculation based on SGP4 model. Remote Sens. Inf. **27**(4), 64–70 (2012)