Software Implementation of Forced Parachuting Simulation System

Ming-Hui Zhang [1,a], Yao-Yu Zhang [2]

[1] Air Force Aviation University,Changchun, China

[2] Changchun Institute of Optics, Fine Mechanics and Physics，Chinese Academy of Sciences，Changchun,China

[a]zhangminghui1974@sina.com.cn

Keywords: parachute、simulation 、MultiGen-Creator、MultiGen-Vega

Abstract: A set of command simulation system, which can be used for training of jumpmasters, later evaluating of training situation, as well as aiding of decision making, is developed in this paper. The software system realizes teaching monitoring, teaching quality assessment, and automatic forced parachuting action analyzing.

Introduction

Function of the simulation system can be expanded from the ground training to air applications. In addition, the system has the function of psychological training of parachuting, which can promote the parachute lifesaving skills training expand to the psychological training. The system is easy to operate, has the strong lifelike, can effectively improve the means of training, improve training efficiency, and improve the ability of personnel parachute lifesaving. Moreover, the system can be used for multi-airplane and multi-parachute lifesaving training, this model can simulate a variety of cabin environment and the outer environment that can be seen from the cockpit. Besides, it realizes various fault simulation of special airplane in an emergency, and simulates judgment and operation in an emergency, including taking parachute, wearing parachute, choosing machine door and parachuting. The system simulates the complete process, such as free fall, open parachute and landing, of the parachutists after getting off the airplane machine. Therefore, it is a comprehensive parachute lifesaving training simulation system involving getting off, manipulation, landing and special condition disposing.

Software Development Process

Software part of the simulation system includes two parts, namely three dimension visual simulation and application software. The whole system is based on Windows XP operating system, using MultiGen-Creator for 3-dimensional solid modeling. The software integrates vector modeling, polygon modeling, terrain generation and other advanced functions. It use MultiGen-Vega for three dimension visual motion control functions, to drive, control, and manage the virtual scene, and at the same time the software also supports rapid complex visual simulation program. Finally, Vega's main application is built on vc + + 6.0 platform. It realizes design and development of the whole system by calling the Vega API application interfaces and loading the configuration file. Software design flow of the system is shown in figure 1.

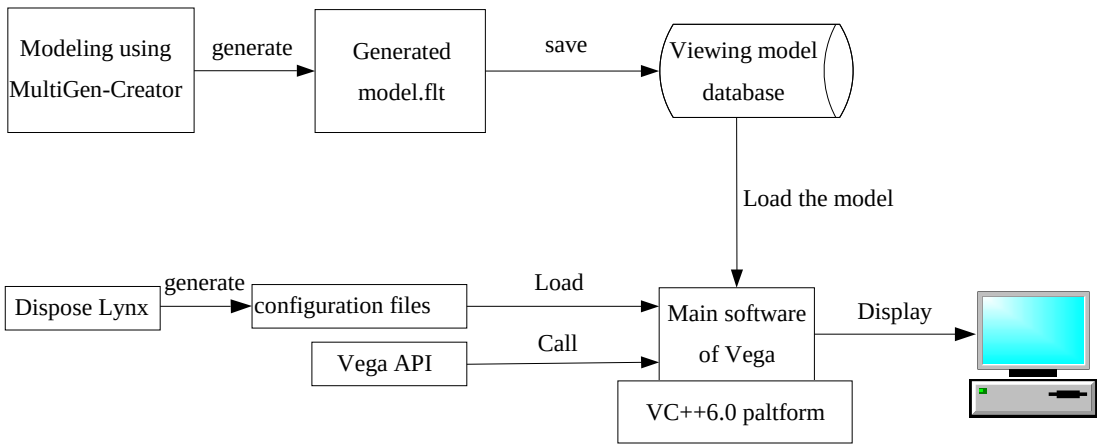

Fig1. Software design flow of command simulation system

Development process of the system includes:

dimensional model construction and terrain modeling

Construction of fine object model

Take the plane model for example. The basic structure of the plane includes the nose, the fuselage, engine room, the win, horizontal tail, vertical tail and so on. Lofting method is used to construct 3-dimensional model of plane in Creator. Loft is to take a two-dimensional shape object as a three-dimensional object formed by moving along a certain path, the same path can be in different period of given different 2-dimensional shape, so as to improve the accuracy of the 3 d model. According to the plane measurement data, the nose vertex, vertical tail setting, wingtip lofting points, on the horizontal tail and vertical tail edge characteristic line sketch in accordance with the characteristic line lofting line. Lofting surface consistent with the actual shape is constructed in the vertical cross section in different parts of the fuselage. The Loft tool of Creator, in turn, lofts all parts of the aircraft, finally complete the whole layout model. Considering the airborne command simulation system for aircraft drop the visual impact of the request is not high, we can use existing aircraft model to reduce the development cycle and complexity.

Construction of terrain model

Build the terrain model commonly used digital elevation model (DEM) format of terrain elevation data, but the original data cannot be used directly in the Creator, must be converted to digital elevation data (DED) format. We can use the Creator's own conversion program transformation conversion tool or a third party. Later, we still need to convert DED again to Creator database formed by pure triangles to improve the display speed significantly. Use Creator's own conversion tool, select the Delaunay terrain transformation algorithm and parameters to complete the conversion. Surface texture is achieved by taking the satellite images as texture images after processing (normally BMP format) to cover on the terrain model.

Configuration of simulation environment

Use Lynx for rapid configuration

To run a Vega application, setting values of initial parameters in the initialization phase is required, and it also needs to maintain or update a large number of parameter values in operation stage. Vega stores these data as file (ADF) format defined by application program. In order to facilitate users to create and edit ADF files, Vega provides a friendly interface, easy to use tool - Lynx, ADF files are generated for the Vega program after the configuration.

Use Vega API to achieve advanced functions

MultiGen Vega contains several hundreds of Application Program Interface (API) functions written in C language. The API functions can be applied for real-time interactive control of virtual scene effectively, which meets the application of complex visual simulation. Vega packages these API functions in the classes, which are included in Vega's development and practical function libraries.

Realization of application program

After configuration of Lynx user interface and the generating of ADF files, programming should be done in VC++6.0 to achieve scene driving and rendering. A Vega application program generally can be divided into two stages, namely the static description and dynamic loop stage.

Static description

(a) Call function vgInitSys() for initialization of the system, which includes graph initialization and creation of shared memory and signal area;

(b) Create various samples of Vega class, which are required in simulation applications, by reading pre-defined ADF files or explicitly calling of Vega API functions.

(c) Call function vgConfigSys() to complete the system configuration and prepare to start the main loop of the Vega application.

Dynamic loop

(a) Call function vgSyncFrame() for synchronous processing of application process of the current frame, that is to ensure that the application process is synchronize with the given frame frequency.

(b) Call function vgFrame() to finish the eliminating and rejecting process of the current frame.

(c) Call various Vega API functions to achieve specific functions of the visual simulation system.

Role definition

There are five roles in command simulation system of airborne and airdrop, namely parachute-carry area commander, pilot, airborne and airdrop, ground-to-air commander, and service team.

The roles above are designed as follows.

Description of parachute-carry area commander

Function: Command to lay parachute-carry area. Determine the position of rest area, parachute-carry area, the first check-line, second check-line, boarding line, and parking station. The corresponding parameters are put in by parachute-carry area commander’s computer. The commands of the ground-to-air commander can be received using an intercom system.

Input: The commands of the ground-to-air commander.

Output: Current status and layout parameters.

Ground-to-air broadcast commander and those onboard

Function: Prompt the parachutists to complete the air movements timely using the intercom. Inform the height and wind speed on the ground. Command the parachutists to complete the landing movements and help them to do with special conditions. Receive the commands of the ground-to-air commander. Input the parameters by keyboard to control the flight direction, and then tell the commander the status of the flight through an intercom.

Input: The commands of the ground-to-air commander.

Output: Current status and correction angle of the flight.

Parachutist

Function: Receive the commands of the ground-to-air commander through the intercom, input the parameters by keyboard for adjusting the current falling orientation.

Input: The commands of the ground-to-air commander.

Output: Current status and correction angle of the parachutist.

Ground-to-air commander

Function: Send commands to the crews, the parachutists and the parachute-carry area commanders using intercoms. Input the parameters by keyboard to control the angle adjusting of flight and the parachutist's airborne and airdrop. Meanwhile the role computer of ground-to-air commander can be regarded as a teacher. It inputs the meteorological parameters by keyboard.

Input: The feedback status of other roles.

Output: Commanders and setup parameters.

Service team

Function: Report the weather situation using the intercom system and input the meteorological parameters by keyboard. Receive the commands of the ground-to-air commanders and feed back the current status. Input parameters to adjust the orientation of landing.

Input: The commands of the ground-to-air commander.

Output: Current status and setup parameters

Conclusions

A forced parachuting system is developed in this paper for ground based teaching of parachuting. So, in a sense, it is a bridge between ground based teaching and airborne teaching. The simulation system achieves the primary functions such as teaching monitoring, teaching quality assessment, and automation and digitization of forced parachuting movement analysis.

References

[1] Wang Cheng, Zhou were clear, Li Lijun. Creator visual simulation modeling [M]. 2005.

[2] Di Liping reality 3D modeling and implementation MuItiGen of [D]. Based on 2005.

[3] Wang Cheng, Zhou were clear, Chen Dawei. Vega time three-dimensional visual simulation technology [M]. 2005.