Microsoft ESP Developer profile white paper

Reality XP Simulation
www.reality-xp.com

Background

Microsoft® ESP™ is a visual simulation platform that brings immersive games-based technology to training and learning, decision support, and research and development modeling for government and commercial organizations.

Microsoft ESP delivers a full, all-in-one platform—a powerful simulation engine, software tools, and realistic world content—to enable developers to affordably and often rapidly create compelling simulation solutions for their customers.

ESP models the entire world, while also allowing developers to easily add their own content, objects, scenery, simulation functionality, and scenarios to create custom training solutions.

Purpose of the white paper

Delivering custom applications on the ESP platform requires different developer profiles and skills. This white paper introduces these profiles per-ESP typical applications.

ESP as a visual simulation

Microsoft ESP is an open flight simulation platform for Visual Simulation. As a platform technology, Microsoft ESP provides a PC-based simulation engine, a comprehensive set of tools, applications programming interfaces, documentation to support code development, content integration and scenario-building capabilities, along with an extensive base of world content that can be tailored for custom solutions. Partners and developers can add structured experiences or missions, content such as terrain and scenery, scenarios, and hardware devices to augment existing solutions, or they can build and deploy new solutions that address the mission-critical requirements of their customers.

Simulations built on the Microsoft ESP platform can take advantage of an extensive set of capabilities to create dynamic, immersive environments, including the following:

Preparedness and decision-making

Includes experiences such as allowing a pilot to pre-fly a new route or become more familiar with new terrain, approach patterns and weather patterns

Solutions for terrain preparedness are based on terrain content authored to be rendered in real time in ESP. This requires modeling ground features like terrain slope (mesh), mapping aerial or satellite imagery on the terrain (ortho-mapping and digital enhancements/cleanup of textures/tiles). It can also require custom 3D modeling of man-made features like buildings, towers, airport buildings and runways.

Developers of terrain centric solutions use ESP SDK tools to process huge number of vector/shape data and bitmap data. For custom man-made features, a 3D modeler working with tools like 3D Studio Max is the ideal profile.

Research and development modeling

Allows a user to visualize ideas, create mockups of cockpits or instrument panels, design specs in 3-D, and evaluate runway configurations or lighting schemes

Research and development modeling require 3D modelers to build virtual representation of object as simple as a switch or a gauge, up to an entire aircraft or airport building. C++ developers are ideal profiles for research and development in instruments, which covers typically symbology assessment, user interface definition and instrument functionality simulation.

Configurable weather model

Includes user-adjustable weather conditions, limitless variations using four cloud types, wind-speed and directional controls, visibility effects and precipitation modeling, or use of near-real-time weather from an external data source feed

Defining and creating weather models is entirely done with ESP SDK tools. Using simple authoring functions, one can create advanced “weather scape”. In addition to the included authoring tool, more sophisticated weather conditions can be generated either from a scripted description file (with the ESP mission system) or from C++ code (with the ESP SimConnect system).

Developers skills range from mission script file editing to C++ development.

Extensive scene generation and scenery support.

Has support for up to five seasons including hard winter, continuous time of day, night textures and scenery support including automatic fill-in

ESP can automatically switch the visual representation of the terrain and the sky based on these factors. Developers of terrain features, as described in “Preparedness and decision-making**”,** can take advantage of this in texturing the terrain with the necessary seasonal variations. This requires designing up to five set of textures for the terrain (one texture set for each season plus hard winter).

ESP as a training platform

With its open platform architecture, ESP features an SDK (Software Development Kit) which permits authoring content targeted at the ESP runtime application. A wide range of solutions can run with, and interact with ESP core simulation engine. Beyond visualization, ESP is a flight simulation platform capable of sophisticated level D grade core simulation software.

Aircraft simulation

The ESP simulation engine is capable of modeling and simulating a wide range of aircraft flight characteristics, aircraft systems, instruments and avionics.

Flight Dynamics

Aircraft flight characteristics, also known as flight dynamics, are modeled from a set of known aircraft laws of motion characteristics, usually sampled in flight tests sessions. This data is then formatted in an authoring tool and the ESP core engine computes in real time the virtual flight path of the aircraft. However, the ESP core engine can only make so many assumptions in an ideal virtual world, and the design of the flight model in ESP requires additional fine tuning with the help of flight dynamics engineers and the feedback of actual pilots, in order to recreate as close as possible the entire flight envelop of the aircraft in the simulated world. Part of the work on the flight model also comes from experience with the ESP flight model engine in both its strength and weaknesses.

Aircraft Systems

The ESP core simulation engine comes with the simulation of a broad range of aircraft system variables, typically covering lighting, pneumatic, electric, hydraulic and engine. Most of the design of the aircraft systems can be as simple as describing key values in a text file. However, for the most demanding simulations, ESP core engine must be augmented, or completely replaced, with custom coded aircraft system simulation in C++.

Aircraft Visuals

With its open platform, ESP permits designing representation of any aircraft in 3D. This permits recreating the shape of the aircraft as seen from the outside, as well as the complete cockpit and cabin from the inside.

In addition, every single part can be animated in order to recreate the actual operating conditions of the aircraft from the perspective of its elements. For example, the landing gear extension and retraction sequence follows the actual dynamic laws of rotation of the gear struts, and the cockpit knobs rotate with user actions.

Designing the aircraft visuals requires a broad range of skills and development best-practices usually found in the entertainment and gaming world, from 3D modeling with tools like 3D studio max to texturing with bitmap tools like Photoshop.

In order to animate the visual, custom code is designed both in a high level XML scripting language and in a lower level C++ code.

Aircraft Gauges and Avionics

In addition to the visual representation of the aircraft in 3D with its animations, the aircraft cockpit includes simulation of its gauges and avionics. The ESP platform offers two ways to design custom gauges and avionics simulation: using a scripted description language, one can layer gauge elements from bitmap and animate the bitmaps with ESP driven data. An airspeed indicator is designed this way with a bitmap (photo) of the ASI background plate, and a rotating bitmap representing the needle shape. The angle of rotation is directly taken from the “AIRSPEED” variable ESP exposes to the gauge code. This method remains very limited both in terms of functionality, extensibility and visuals.

More sophisticated gauges and avionics are coded in C++ in order to reach the level of sophistication the scripting method can’t deliver. In addition, C++ coded gauges and avionics can further expand beyond the ESP SDK and permit external communication to other applications, access to database and file based information and can further extend the user interface in enabling keyboard shortcuts or special hardware interfaces to the simulated gauge and avionics user input devices (knobs, keys).

Aircraft Sounds

The ESP platform comes standard with support for 5.1 surround sound. This permits recreating high fidelity sound environments as eared from the outside, or the inside. Typical developer profile is a sound engineer using sound editing tools.

In order to compete with Level D grade certification however, the ESP engine is not yet capable to drive sophisticated sound sources simultaneously. In addition, the “sound triggers” which are used in the ESP sound system are limited in number.

A much higher aircraft simulation is possible with ESP in developing custom coded sound triggers in C++ that can be driven from the aircraft simulation.

Flight training and rehearsal

Includes cockpit familiarization, checklists and cockpit flows, and capability-based training such as aerial refueling and basic sortie.

Developing flight training solution requires skills and experience in the training market, regardless of technology used. To that effect, ESP is a vector to deliver content that is authored in high level applications/languages. Training content in ESP is similar to training content usually found in CBT solutions. However, ESP permits complete interaction in the learning process as the student uses the platform to access to the authored content.

Multiplayer functionality and Internet support including Voice over IP

Allows up to 30 people to interact around the world using a peer-to-peer broadband connection.

Custom networked solutions can be developed to take advantage of ESP internet support. One such solution is called “shared cockpit” and permits sharing the same aircraft between two internet connected computers. Special sharing and synchronization code is developed in C++ at the aircraft simulation level in order to support this feature.