Samsung Smart TV SDK 5.0 Overview

Published 2015-08-21 | (Compatible with SDK 5.0 and 2014 models)

Overview of Samsung Smart TV SDK 5.0 including descriptions of the development environment, existing features, and new features such as Native Client, Samsung Smart View SDK, and Smart Home. Also explains differences between Emulators and Smart TV hardware.

Contents

- Introduction
- Development Environment
- Smart TV Applications
- New Features in SDK 5.0
  - Native Client (NaCl)
  - Smart View SDK
  - Smart Home
  - Web UI Framework
  - Debugging with Web Inspector
  - Tutorial Explorer
  - Automatic Test Tool
  - Browser IDE
- Existing Features in Samsung Smart TV SDK
  - AllShare
  - Convergence API
  - HTTP Adaptive Streaming
  - Smart Interaction
  - Web Standards
- Differences between the Emulator and the Smart TV Hardware
- Additional Resources

Introduction

The Samsung Smart TV SDK 5.0 is designed for developers who want to create applications on the 2014 Samsung Smart TV platforms. The SDK includes the following tools:

Eclipse-based Application Editor and Visual Editor
TV Emulator
Debugger
The S Client Emulator, a tool for testing applications on remote TV Emulator servers.
Browser-based IDE and Emulator

Support for Windows, Mac OS X, and Linux platforms. Support for 32-bit and 64-bit OS's.

Development Environment

The Samsung Smart TV SDK 5.0 supports the Windows, Mac OS X, and Linux platforms. The Eclipse-based App Editor IDE provides developers with many of the tools they need to create Smart TV applications. Some of the features include:

- Context-sensitive text editors for HTML, CSS, and JavaScript
- A Visual Editor for creating GUIs
- Debugger
- TV Application Packaging
- Pre-Test Tool
- Automatic Test Tool
- Tutorial Explorer

The SDK also provides TV Emulators for all platforms. The TV Emulator can help developers test their application in their development environment before testing on a real TV.

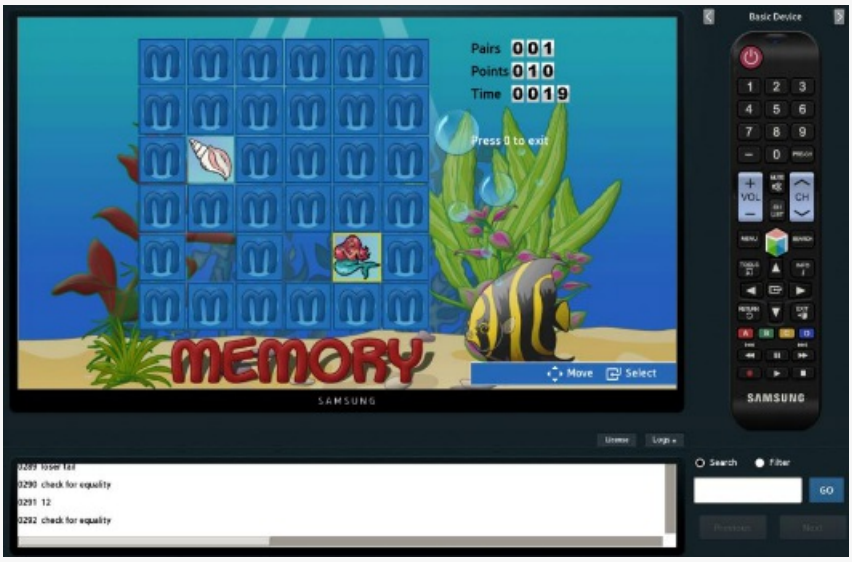

Figure 1: The Smart TV Emulator.

Smart TV Applications

Smart TV applications are web-based software programs that run on digital TVs connected to the Internet. With Smart TV applications, TV users can access web content via their TV screens. Unlike general web pages, applications for Samsung Smart TV allow users to employ TV-specific features.

For example:

- Application plugins can allow users to change the application's volume or play a video that is not part of the television broadcast.
- Functions provided by the File API allow users to use a file storage system.

In order to make application development easier, the Smart TV SDK categorizes applications into the following groups:

Basic Application

For developers who want to create applications using the Visual Editor.

JavaScript Application

For developers who are comfortable writing their applications entirely in code (HTML, CSS, JavaScript).

NaCl Application

For developers who want to write C/C++ applications.

New Features in SDK 5.0

Native Client (NaCl)

The Native Client feature of the Smart TV platform provides the infrastructure to run native C/C++ code as part of a web application. Performance is generally higher than pure JavaScript applications, and existing C/C++ apps can be ported to

the TV. The Smart TV IDE supports the creation of Native Client apps, and the Emulator is capable of executing them.

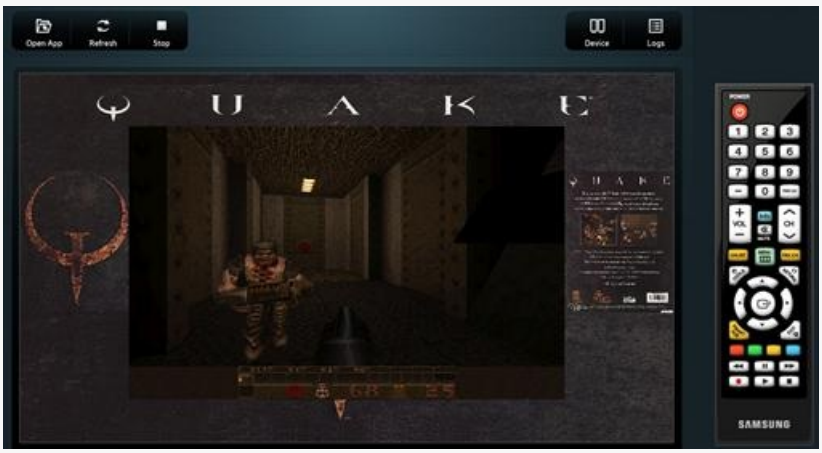

Figure 2: Use PNaCl to Port C/C++ Games to the Smart TV Platform.

Smart View SDK

The Smart View SDK allows developers to create applications that span multiple platforms including Smart TV, smartphone, and tablet. The Smart View SDK is provided as an easy-to-use JavaScript API. For other platforms, developers can choose between Android (Java), iOS, or other web-based applications. By combining the Smart View API with modern web standards such as WebGL, Web Sockets, and HTML5 Canvas, there is no end to the types of applications that can be created: social, games, fitness, music, and more.

Figure 3: The Smart View SDK can be used to develop applications that span multiple devices.

Smart Home

The Smart Home API allows developers to use a WiFi-based Samsung protocol to communicate between the Smart TV and supported home appliances.

Web UI Framework

The Web UI Framework is a new high-performance JavaScript framework for the Smart TV. Developers can write their applications in JavaScript and utilize improved performance to create a variety of TV apps including game, video, and entertainment apps.

Debugging with Web Inspector

The Emulator and IDE are integrated with the Web Inspector debugging tool. This feature provides many debugging options such as JavaScript breakpoints, watch expressions, network profiling, and more.

Tutorial Explorer

The Tutorial Explorer lets developers easily browser and read Smart TV application tutorials.

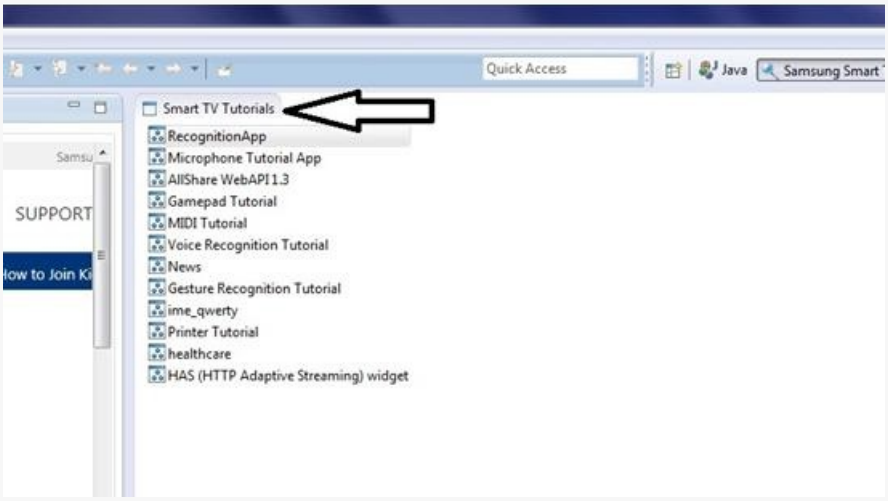

Figure 4: The Tutorial Explorer in the IDE allows the Smart TV tutorial applications to be easily viewed in one convenient place.

Automatic Test Tool

Create test scripts with input events (ex: remote control events) to automate testing of your applications.

Browser IDE

Browser IDE is a powerful and flexible online integrated development environment for Smart TV applications on browser. There are four main functions, including project manager, code editors, terminal, and running application in Smart TV emulator. Firstly, for project manager, basic information of project, such as public URL, is showed and description of project is available to edit. Secondly, there are three kinds of editors, HTML editor, JavaScript editor, CSS editor. They all provide highlighting syntax, automatic completion and some other powerful features. Thirdly, there is a terminal for entering more complex commands. Additionally there's a Smart TV emulator for developers to run application on the browser.

With Browser IDE, you can develop applications anywhere and work more efficiently.

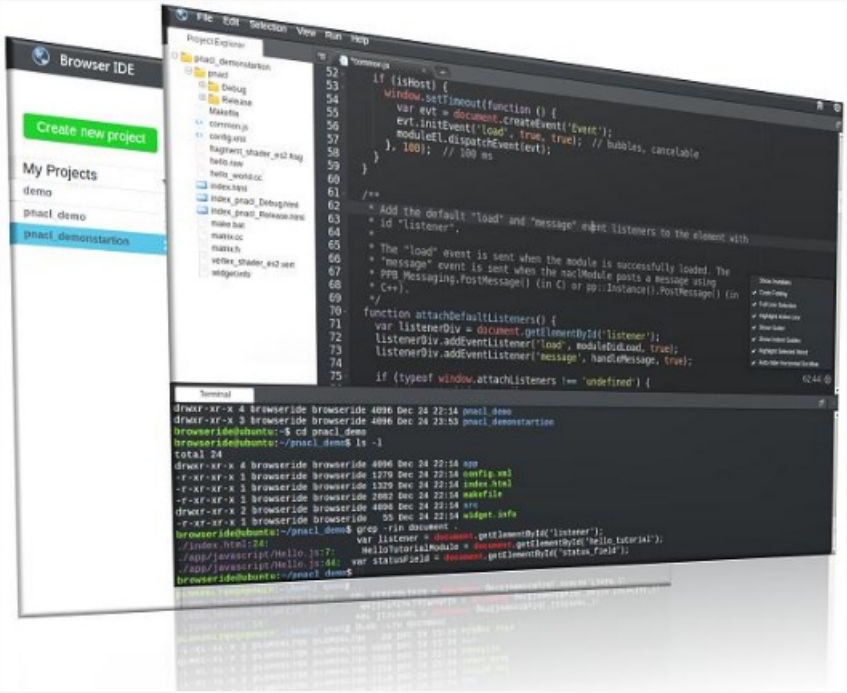

Figure 5: The Browser IDE

Existing Features in Samsung Smart TV SDK

AllShare

Samsung AllShare enables content sharing across devices. Pictures, movies, and music can be shared between computers, TVs, mobile devices, and more. The AllShare API gives Smart TV applications the power to interact with AllShare-enabled devices. For more information about AllShare, please refer to AllShare Tutorial.

Convergence API

The Convergence API is designed to enable communication between mobile and TV devices. An example of a convergence application is using a smartphone as a remote control for a SmartTV application. For more information about the Convergence API, please refer to Creating a Convergence Application.

HTTP Adaptive Streaming

The Smart TV SDK supports applications with HTTP Adaptive Streaming (HAS). This feature allows video applications to dynamically adjust bitrate streaming based on network conditions. For more information about HTTP Adaptive Streaming, please refer to Creating a Video Application With HAS (HTTP Adaptive Streaming).

Smart Interaction

The Smart TV SDK provides an API for interacting with the TV via voice and gestures. Additionally, the Emulator contains simulation GUI for gesture and voice input.

Figure 6: Simulator for Smart Interaction input.

Web Standards

The Smart TV SDK continues to support web standards such as HTML5 (audio/video/canvas tags) and WebGL.

Differences between the Emulator and the Smart TV Hardware

The Emulators are provided so that developers can conveniently test their applications in their workstation environment. However, the Emulators do not exactly simulate the real TV, and developers should always perform final testing on a real TV device.

Some differences between the Emulators and the real TV device are:

Performance: RAM, CPU, and network status are just some of the factors that influence an application's performance. These variables are different between the PC and TV, and as a result performance should be carefully observed on the real TV. In general, the TV will have less memory than most developer workstations.

Interface Viewing Distance: Even though applications maintain the same aspect ratio in the Emulator and the TV, viewers in the TV environment are further away from the screen. The usability of applications should be tested on a real TV.

Remote Control: On a real TV, the application cannot override the default behavior of some of the remote control buttons. Additionally, the response time of button pushes may vary.