



# Breakthrough Games with Tizen

Tizen Graphics

Samsung

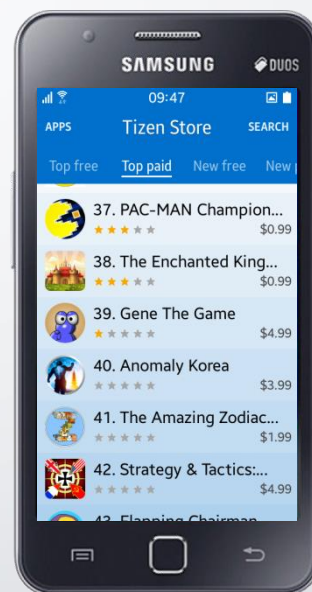
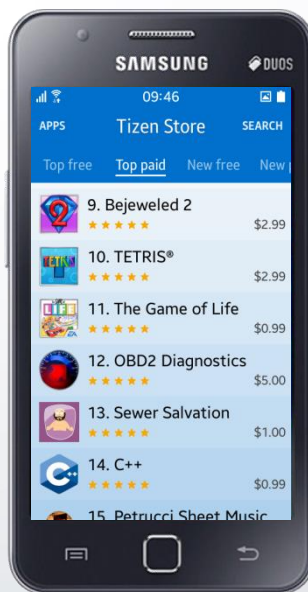
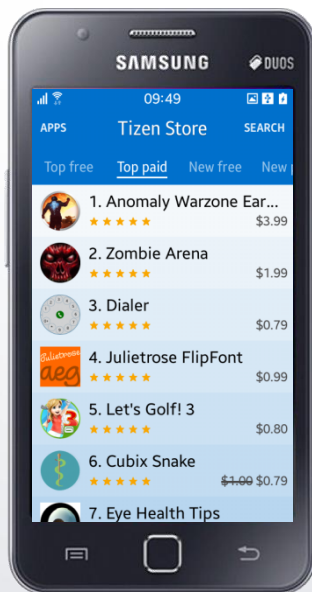
# Agenda

---

1. Introduction
2. Game Porting to Tizen
3. Tips for Development
4. Monetization
5. Summary

# Introduction

- Market Status
  - 27 Games in TOP rank 50 (2015.06)



# Why Tizen?

- Expandability and Convergence



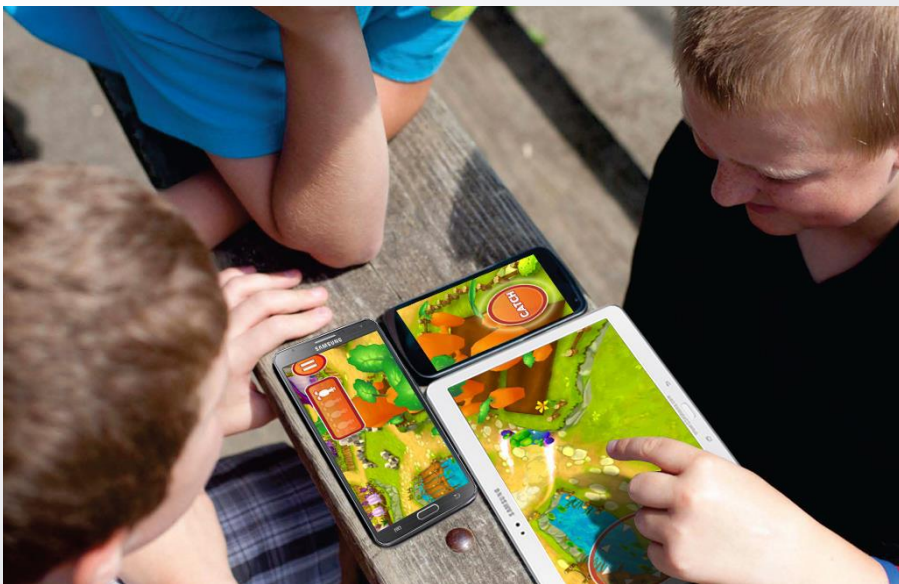
# Why Tizen?

- Unexplored and New Game Area

Multi-user Gaming Experience

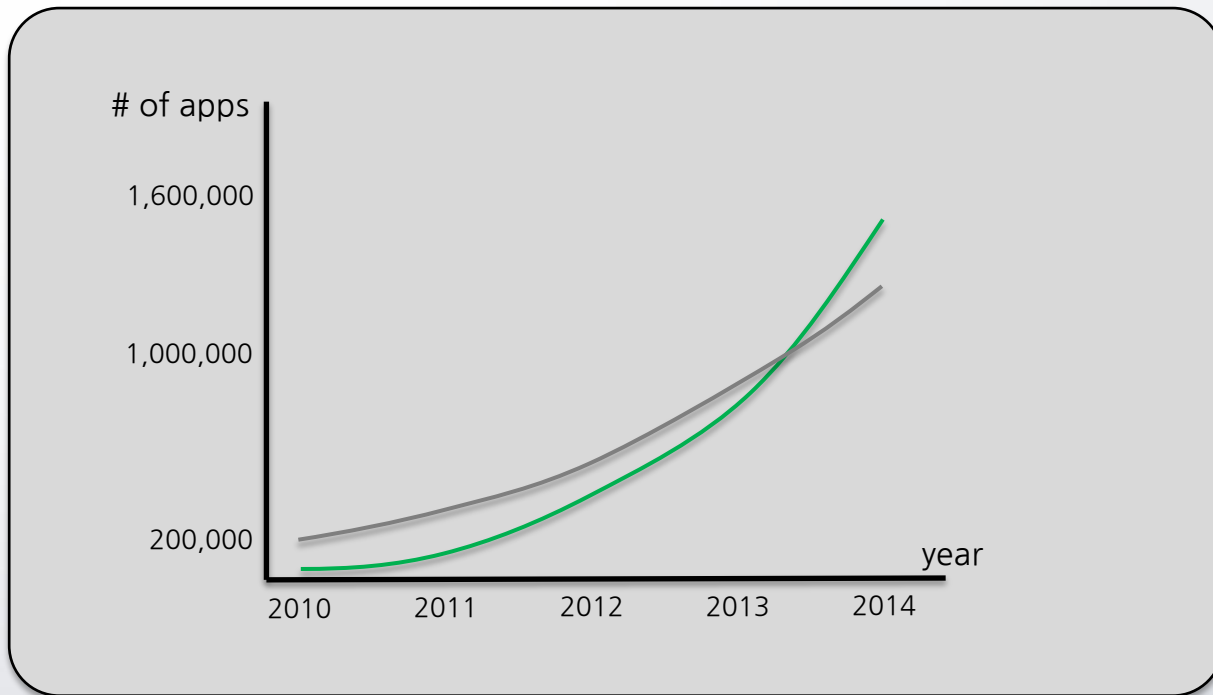


Multiple screens Game Experience



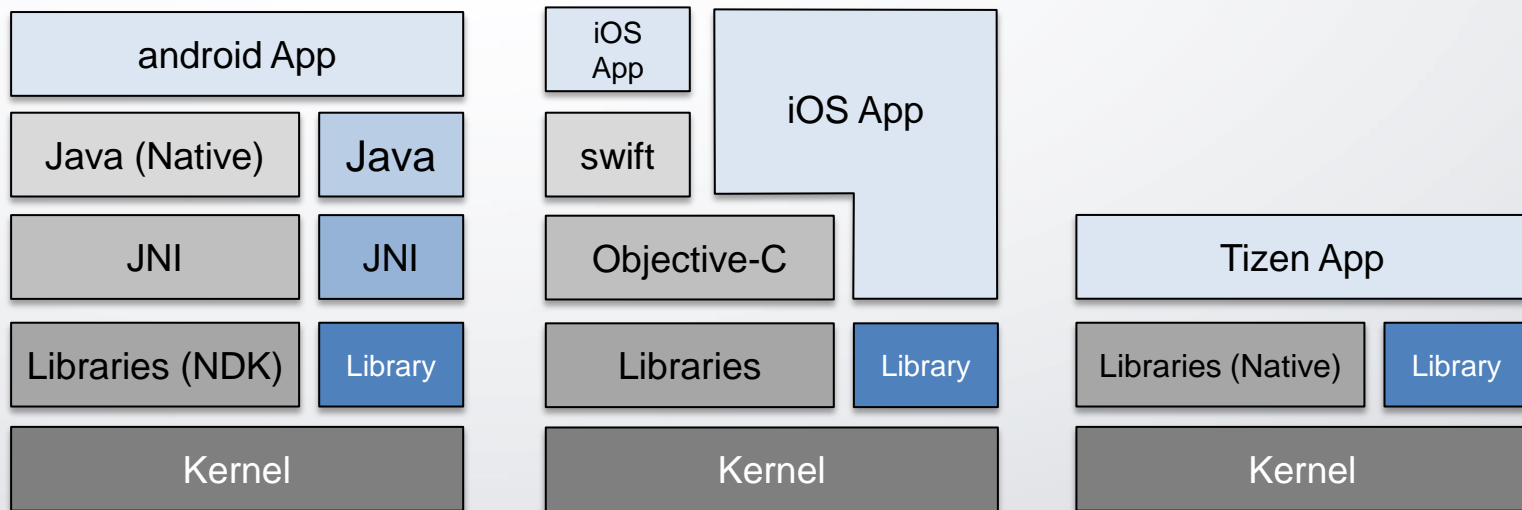
# Why Tizen?

- New devices, New marketplace, New opportunity
  - Hard to make your games visible to users



# Why Tizen?

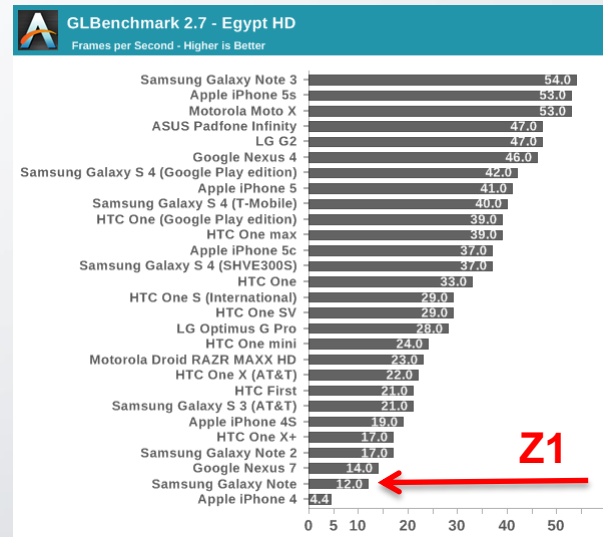
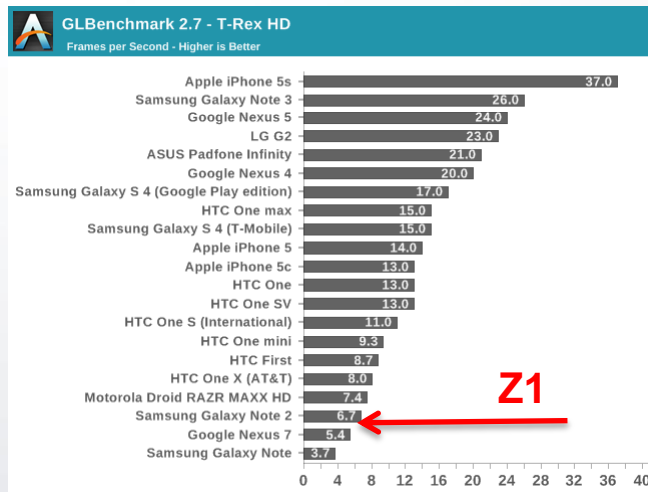
- Efficiency
  - Development efficiency
    - C-based modules accelerate porting of open source modules



# Why Tizen?

- Efficiency
  - Device performance
    - Graphics is highly optimized, which is deeply impressive for mass model with limited resources

Result	
Direct	
GLBenchmark 2.7 T-Rex HD	385 frames
C24Z16 Onscreen ETC1	6.9 fps
GLBenchmark 2.7 T-Rex HD	111 frames
C24Z16 Offscreen ETC1	2.0 fps
GLBenchmark 2.5 Egypt HD	1739 frames
C24Z16 Onscreen ETC1	15 fps
GLBenchmark 2.5 Egypt HD	960 frames
C24Z16 Offscreen ETC1	8.5 fps
None	
No results yet	-1





# Why Tizen?

- Developer friendly Tizen Store Seller Promotion



A promotional banner for the Tizen Store Seller Promotion. The background is dark blue with a large, bright yellow '100%' and the text 'is YOURS!' in a lighter yellow. Below this, white text reads 'No need to share your revenue for one year! Join us and Submit your Applications Now'. In the top right, a teal speech bubble contains the text 'ADD NEW APPLICATION' and a white right-pointing arrow. At the bottom left, the text 'PERIOD | One Year (January 14 2015 ~ January 31 2016)' is displayed in orange. The banner is decorated with colorful geometric shapes in orange, yellow, and teal.

**100%**  
is YOURS!

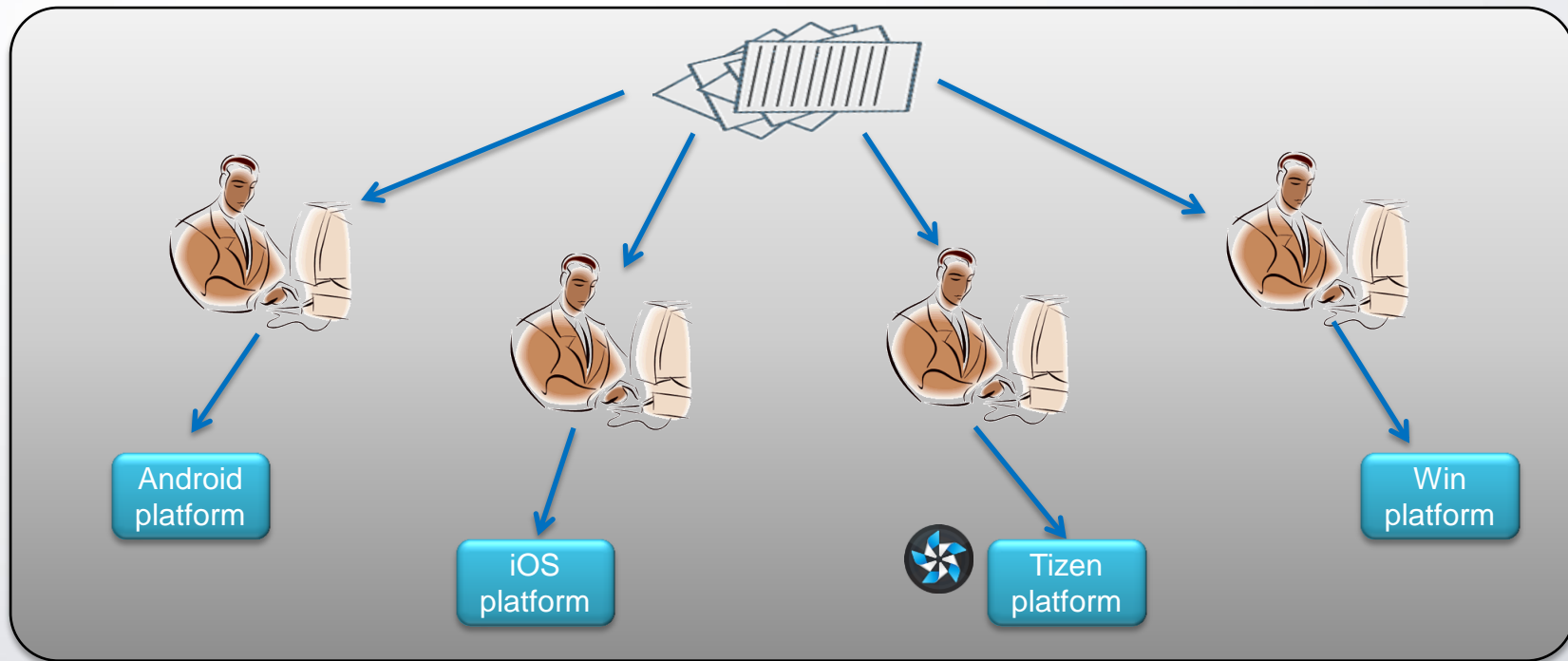
No need to share your revenue for one year!  
Join us and Submit your Applications Now

ADD NEW APPLICATION >

PERIOD |  
One Year (January 14 2015 ~ January 31 2016)

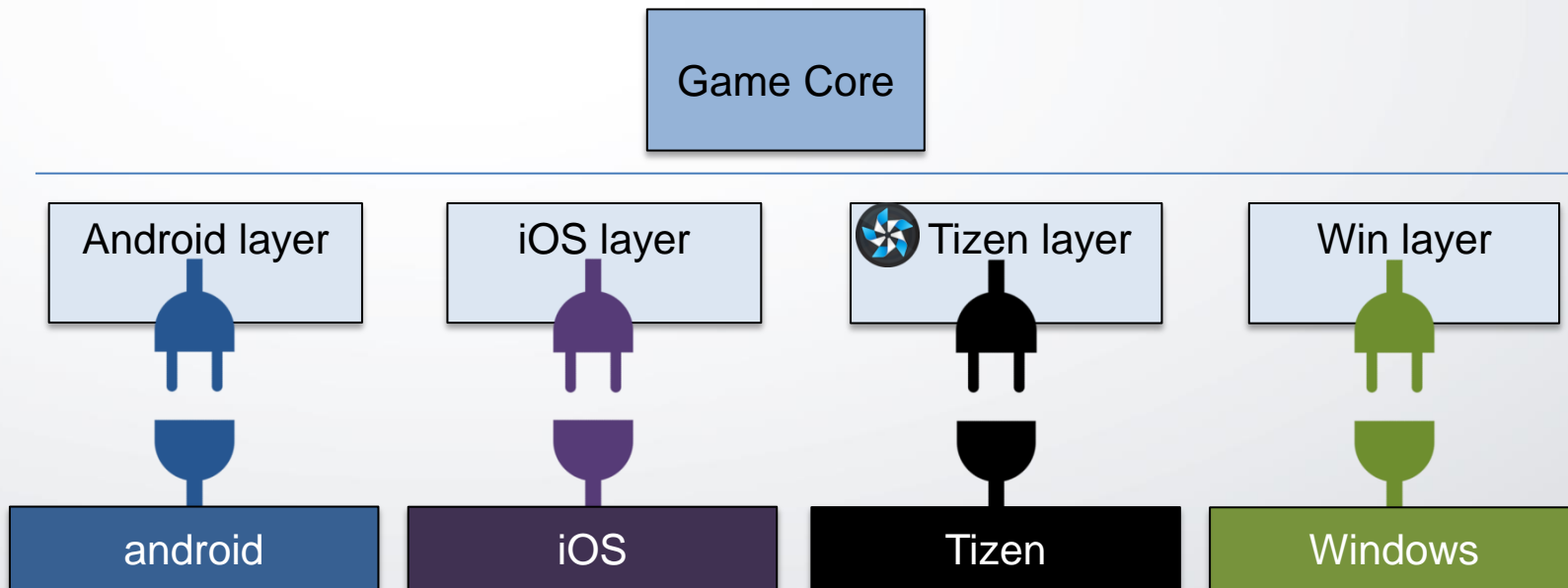
# Game Porting to Tizen

- Typical way
  - Develop game for Tizen with the same scenario



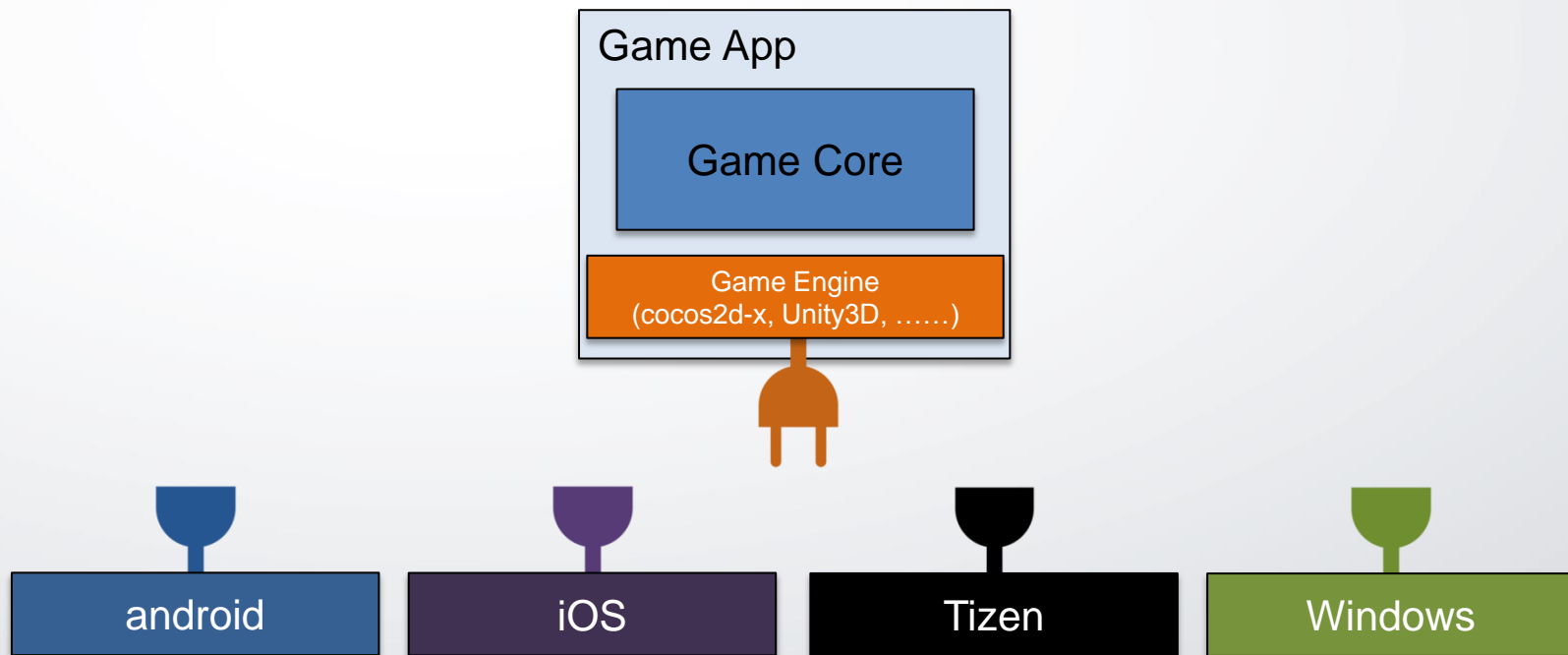
# Game Porting to Tizen

- Better way
  - Divide porting layer from game core, and adapt only porting layer



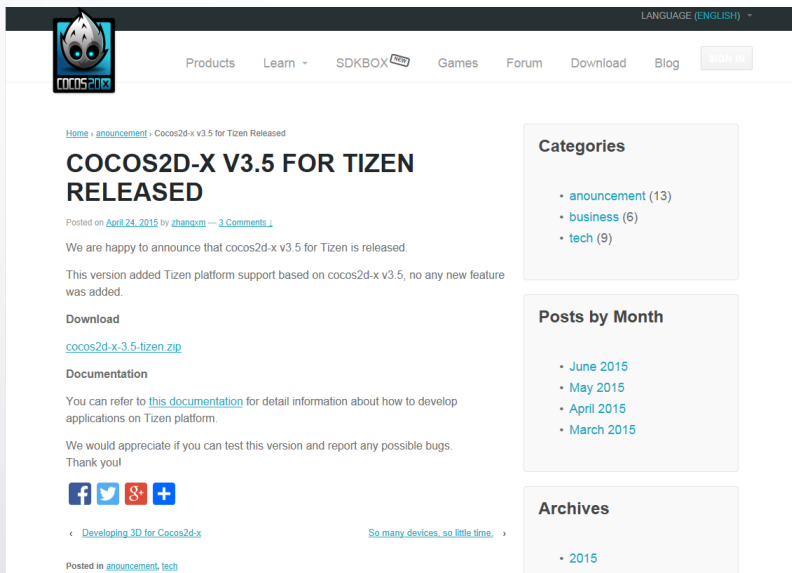
# Game Porting to Tizen

- Best way
  - Adopt game engines, such as cocos2d-x & Unity3D



# Game Porting to Tizen

- Famous Game Engines are ready for Tizen
  - Cocos2d-x (since ver.3.5.1)
  - Unity3D (since ver.5.1)



The screenshot shows the Cocos2d-x website with a header navigation bar including links for Products, Learn, SDKBOX, Games, Forum, Download, and Blog. The main content area features an announcement titled "COCOS2D-X V3.5 FOR TIZEN RELEASED" dated April 24, 2015. It states that the version added Tizen platform support based on cocos2d-x v3.5. A "Download" section provides a link to "cocos2d-x-3.5-tizen.zip". A "Documentation" section refers to a link for development details. Social media icons for Facebook, Twitter, Google+, and YouTube are at the bottom left. A sidebar on the right contains "Categories" (announcement, business, tech) and "Posts by Month" (June, May, April, March 2015).

Home › announcement › Cocos2d-x v3.5 for Tizen Released

## COCOS2D-X V3.5 FOR TIZEN RELEASED

Posted on April 24, 2015 by zhangm — 3 Comments ↓

We are happy to announce that cocos2d-x v3.5 for Tizen is released.

This version added Tizen platform support based on cocos2d-x v3.5, no any new feature was added.

**Download**

[cocos2d-x-3.5-tizen.zip](#)

**Documentation**

You can refer to [this documentation](#) for detail information about how to develop applications on Tizen platform.

We would appreciate if you can test this version and report any possible bugs. Thank you!

[f](#) [t](#) [g+](#) [y](#)

« Developing 3D for Cocos2d-x [So many devices, so little time.](#) »

Posted in [announcement](#), [tech](#)

**Categories**

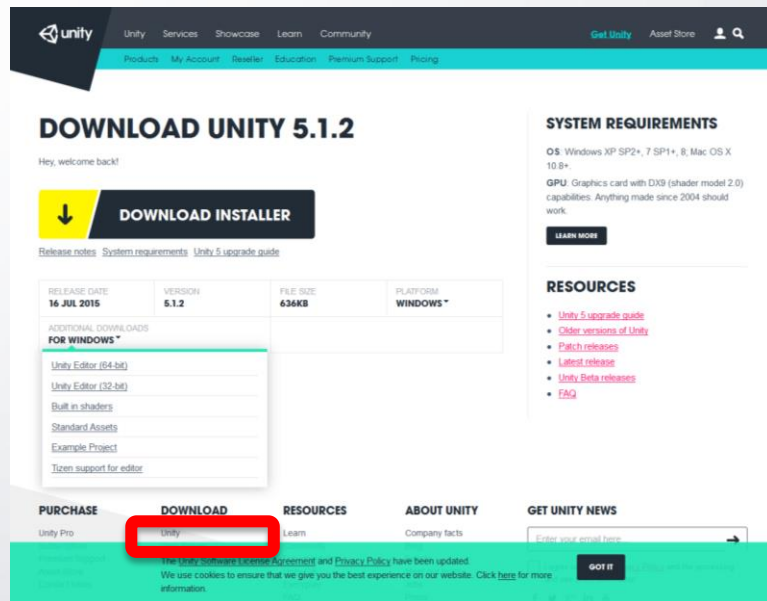
- [announcement](#) (13)
- [business](#) (6)
- [tech](#) (9)

**Posts by Month**

- [June 2015](#)
- [May 2015](#)
- [April 2015](#)
- [March 2015](#)

**Archives**

- [2015](#)



The screenshot shows the Unity website's "DOWNLOAD UNITY 5.1.2" page. It includes a navigation bar with links for Unity, Services, Showcase, Learn, and Community. The main content area features a "DOWNLOAD INSTALLER" button and a table of release information. A dropdown menu for "ADDITIONAL DOWNLOADS FOR WINDOWS" is open, showing links to Unity Editor (64-bit), Unity Editor (32-bit), Built-in shaders, Standard Assets, Example Project, and Tizen support for editor. The right sidebar contains "SYSTEM REQUIREMENTS" and "RESOURCES". The bottom navigation bar includes links for PURCHASE, DOWNLOAD, RESOURCES, ABOUT UNITY, and GET UNITY NEWS. The "DOWNLOAD" link is highlighted with a red box.

unity

Unity Services Showcase Learn Community

Get Unity Asset Store

Products My Account Developer Education Premium Support Pricing

## DOWNLOAD UNITY 5.1.2

Hey, welcome back!

[Download Installer](#)

[Release notes](#) [System requirements](#) [Unity 5 upgrade guide](#)

RELEASE DATE	VERSION	FILE SIZE	PLATFORM
16 JUL 2015	5.1.2	636KB	WINDOWS *

**ADDITIONAL DOWNLOADS FOR WINDOWS \***

- [Unity Editor \(64-bit\)](#)
- [Unity Editor \(32-bit\)](#)
- [Built-in shaders](#)
- [Standard Assets](#)
- [Example Project](#)
- [Tizen support for editor](#)

**SYSTEM REQUIREMENTS**

OS: Windows XP SP2+, 7 SP1+, 8, Mac OS X 10.8+

GPU: Graphics card with DX9 (shader model 2.0) capabilities. Anything made since 2004 should work.

[LEARN MORE](#)

**RESOURCES**

- [Unity 5 upgrade guide](#)
- [Older versions of Unity](#)
- [Patch releases](#)
- [Latest release](#)
- [Unity Beta releases](#)
- [FAQ](#)

**PURCHASE** **DOWNLOAD** **RESOURCES** **ABOUT UNITY** **GET UNITY NEWS**

Unity Pro [Unity](#) Learn Company facts

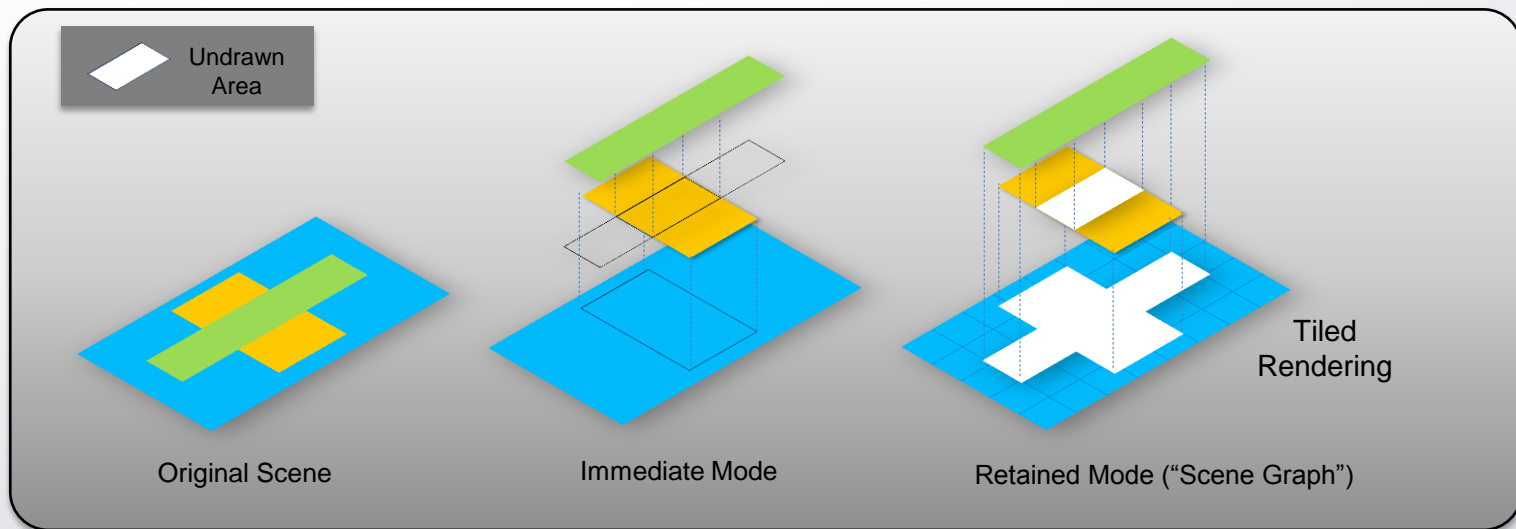
Enter your email here

[GET IT](#)

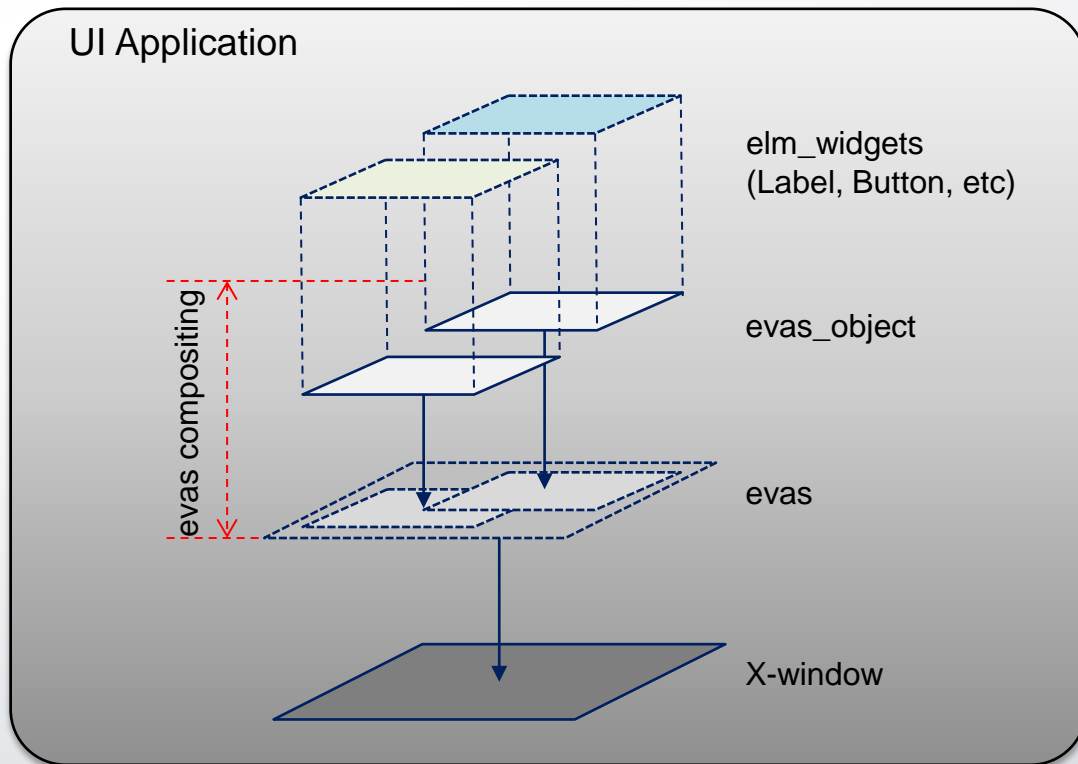
The Unity Software License Agreement and Privacy Policy have been updated. We use cookies to ensure that we give you the best experience on our website. Click [here](#) for more information.

# Tips for Development - evasgl basics (1)

- EFL (Enlightenment Foundation Libraries)
  - Collection of open source libraries from Enlightenment
- evas (Efl + canVAS)
  - evas is Scene Graph composed of 'evas objects'



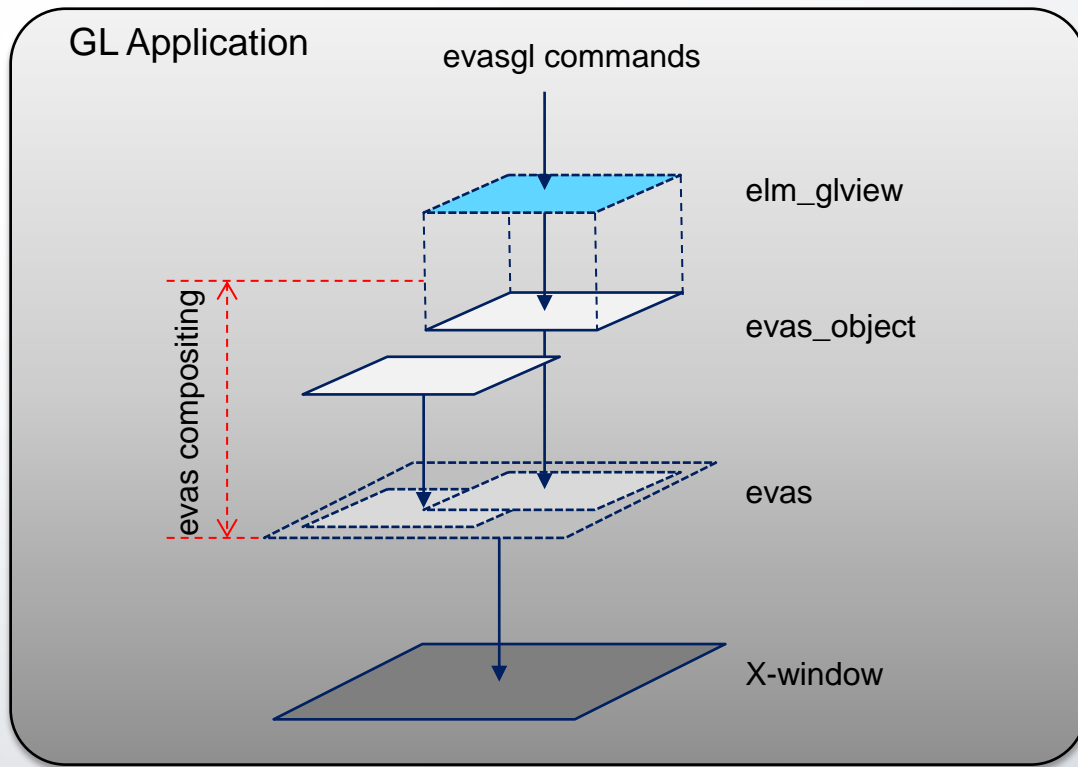
- EFL View Hierarchy



- GPU Accelerated Rendering in EFL
  - How to make a surface for GLES?
  - How the surface is composited with other widgets?
- **evasgl**
  - Abstraction for EGL and OpenGL-ES
    - EGL related operations are automatically and internally processed in evas
    - Provides wrappers for the native OpenGL-ES calls
  - Rendering results by evasgl goes to evas object
    - All evas objects are smoothly composited in EFL view hierarchy



- Revisit EFL View Hierarchy



# Sameple code – Draw one cube

---

- Overall sequence of sample codeS
  - Application initialization
  - evasgl initialization
  - Animation and rendering settings
  - Add animator and renderer to ecore main loop
  - Define rendering with evasgl functions



# 1. Application Initialization

app\_main part

```
#include <Elementary.h>
#include <Evas_GL.h>
.....
```

```
// Define a global context for the application
typedef struct appdata {
    Evas_Object *win;
    Evas_Object *img;
    Evas_GL *evasgl;
    Evas_GL_API *glapi;
    Evas_GL_Context *ctx;
    Evas_GL_Surface *sfc;
    Evas_GL_Config *cfg;
    unsigned int program;
    unsigned int vtx_shader;
    unsigned int fgmt_shader;
    unsigned int vbo;
} appdata_s;
```

```
int main(int argc, char *argv[])
{
    appdata_s ad = {0,};
    int ret = 0;

    ui_app_lifecycle_callback_s event_callback = {0,};
    .....
```

```
    event_callback.create = app_create;
    event_callback.terminate = app_terminate;
    event_callback.pause = app_pause;
    event_callback.resume = app_resume;
    event_callback.app_control = app_control;
```

```
.....
```

```
    ret = ui_app_main(argc, argv, &event_callback, &ad);

    return ret;
}
```

## 2. evasgl Initialization

### evasgl initialization

```
/* Set config of the surface for evas gl */
ad->cfg = evas_gl_config_new();
ad->cfg->color_format = EVAS_GL_RGBA_8888;    // Surface Color Format
ad->cfg->depth_bits   = EVAS_GL_DEPTH_BIT_24; // Surface Depth Format
ad->cfg->stencil_bits = EVAS_GL_STENCIL_NONE;  // Surface Stencil Format
ad->cfg->options_bits = EVAS_GL_OPTIONS_NONE; // Configuration options (here, no extra options)
```

```
/* Add Window */
ad->win = elm_win_util_standard_add("Evas_GL Example", "Evas_GL Example");
```

```
/* Get the evas gl handle for doing gl things */
ad->evasgl = evas_gl_new(evas_object_evas_get(ad->win));
ad->glapi = evas_gl_api_get(ad->evasgl);
```

```
/* Get the window size */
Evas_Coord w,h;
evas_object_geometry_get(ad->win, NULL, NULL, &w, &h);
```

```
/* Create a surface and context */
ad->sfc = evas_gl_surface_create(ad->evasgl, ad->cfg, w, h);
ad->ctx = evas_gl_context_create(ad->evasgl, NULL);
```

```
/* Initialization GLES including shader gneration and other stuffs */
.....
```

# 3. Animation and Rendering setting

## Animation and Rendering

```
/* Set up the image object. A filled one by default. */
ad->img = evas_object_image_filled_add(evas_object_evas_get(ad->win));
evas_object_image_pixels_get_callback_set(ad->img, img_pixels_get_cb, ad);

/* Add Event Callbacks */
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_DEL, img_del_cb, ad);
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_MOUSE_DOWN, mouse_down_cb, ad);
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_MOUSE_UP, mouse_up_cb, ad);
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_MOUSE_MOVE, mouse_move_cb, ad);
evas_object_event_callback_add(ad->win, EVAS_CALLBACK_RESIZE, win_resize_cb, ad);

/* Add animator */
ani = ecore_animator_add(animate_cb, ad->img);
```

```
static Eina_Bool
animate_cb(void *data)
{
    Evas_Object *img = data;
    evas_object_image_pixels_dirty_set(img, EINA_TRUE);

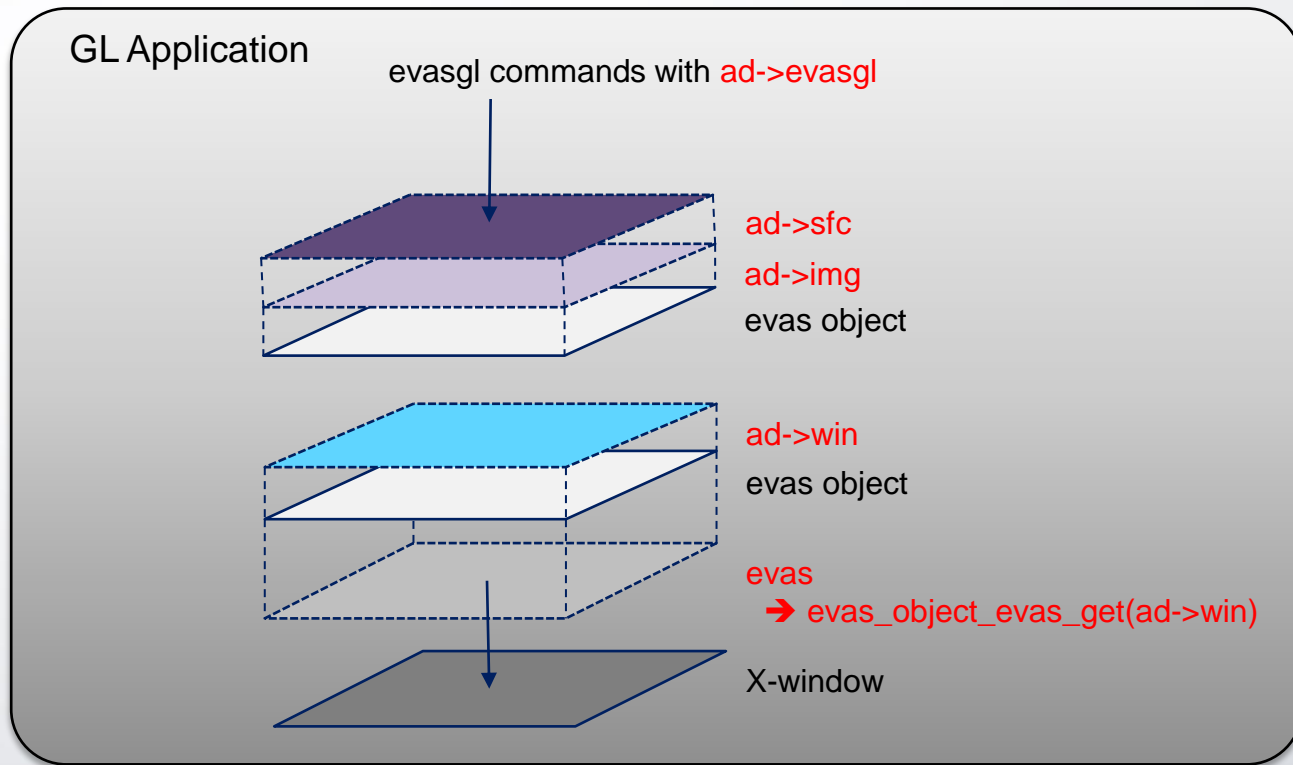
    return Ecore_Callback_Renew;
}
```

```
static void
img_pixels_get_cb(void *data, Evas_Object *obj)
{
    appdata_s *ad = data;
    Evas_GL_API *gl = ad->glapi;

    // Rendering process here
    .....
}
```

# Sameple code – Draw one cube

- EFL View Hierarchy for evasgl initialization



### 3. Animation and Rendering setting

#### Animation and Rendering

```
/* Set up the image object. A filled one by default. */
ad->img = evas_object_image_filled_add(evas_object_evas_get(ad->win));
```

```
evas_object_image_pixels_get_callback_set(ad->img, img_pixels_get_cb, ad);
```

```
/* Add Event Callbacks */
```

```
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_DEL, img_del_cb, ad);
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_MOUSE_DOWN, mouse_down_cb, ad);
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_MOUSE_UP, mouse_up_cb, ad);
evas_object_event_callback_add(ad->img, EVAS_CALLBACK_MOUSE_MOVE, mouse_move_cb, ad);
```

```
evas_object_event_callback_add(ad->win, EVAS_CALLBACK_RESIZE, win_resize_cb, ad);
```

```
/* Add animator */
```

```
ani = ecore_animator_add(animate_cb, ad->img);
```

```
static Eina_Bool
```

```
animate_cb(void *data)
```

```
{
    Evas_Object *img = data;
    evas_object_image_pixels_dirty_set(img, EINA_TRUE);

    return ECORE_CALLBACK_RENEW;
}
```

```
static void
```

```
img_pixels_get_cb(void *data, Evas_Object *obj)
```

```
{
    appdata_s *ad = data;
    Evas_GL_API *gl = ad->glapi;

    // Rendering process here
    .....
}
```

## 4. Rendering with evasgl

### Rendering with evasgl

```
static void
img_pixels_get_cb(void *data, Evas_Object *obj)
{
    appdata_s *ad = data;
    Evas_GL_API *gl = ad->glapi;
    .....

    /* Make the application context as current */
    evas_gl_make_current(ad->evasgl, ad->sfc, ad->ctx);

    /* Render the scene with evasgl functions */
    gl->glViewport(0, 0, WIDTH, HEIGHT);

    gl->glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    gl->glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    gl->glUseProgram(ad->program);

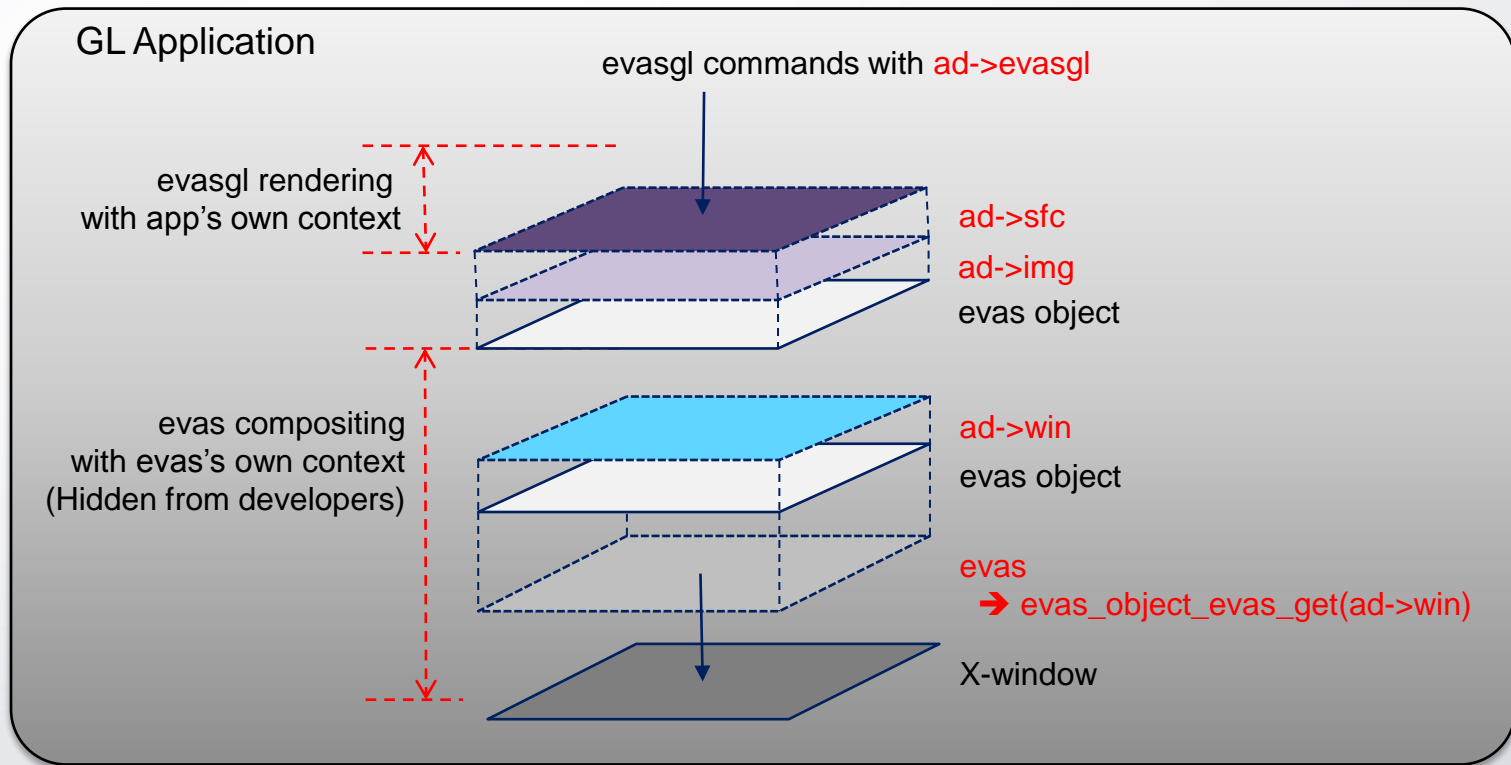
    gl->glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, sizeof(float) * 6, 0);
    gl->glEnableVertexAttribArray(0);

    gl->glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, sizeof(float) * 6, (void*)(sizeof(float)*3));
    gl->glEnableVertexAttribArray(1);
    .....
}
```



## [Caution] Context Handling

- GLES context maintaining with **`evas_gl_make_current`**



- Elementary widget specialized for evasgl rendering
    - Preset tedious work for evasgl rendering for developers
      - Comparable to android.opengl.GLSurfaceView
      - Help developers to focus on only rendering task
    - What does elm\_glvview work for you?
      - Context & Drawable Surface generation
      - Setup all required callbacks including all useful events, such as touch and rendering
      - Guarantee the context maintaining automatically
      - Preset all necessary EGL properties according to the user input requirements
- ( ➔ **elm\_glvview\_mode\_set** )

# Sample Code – Change Initialization

```
EAPI int elm_main(int argc EINA_UNUSED, char **argv EINA_UNUSED)
{
    Evas_Object *win;
    Evas_Object *glview;
    .....

    win = elm_win_util_standard_add("glview", "GLView");
    evas_object_show(win);

    /* Initialize & Setup elm_glview */
    {
        glview = elm_glview_add(win);
        elm_win_resize_object_add(win, glview);

        elm_glview_mode_set(glview, ELM_GLVIEW_ALPHA | ELM_GLVIEW_DEPTH );

        elm_glview_resize_policy_set(glview, ELM_GLVIEW_RESIZE_POLICY_RECREATE);
        elm_glview_render_policy_set(glview, ELM_GLVIEW_RENDER_POLICY_ON_DEMAND);

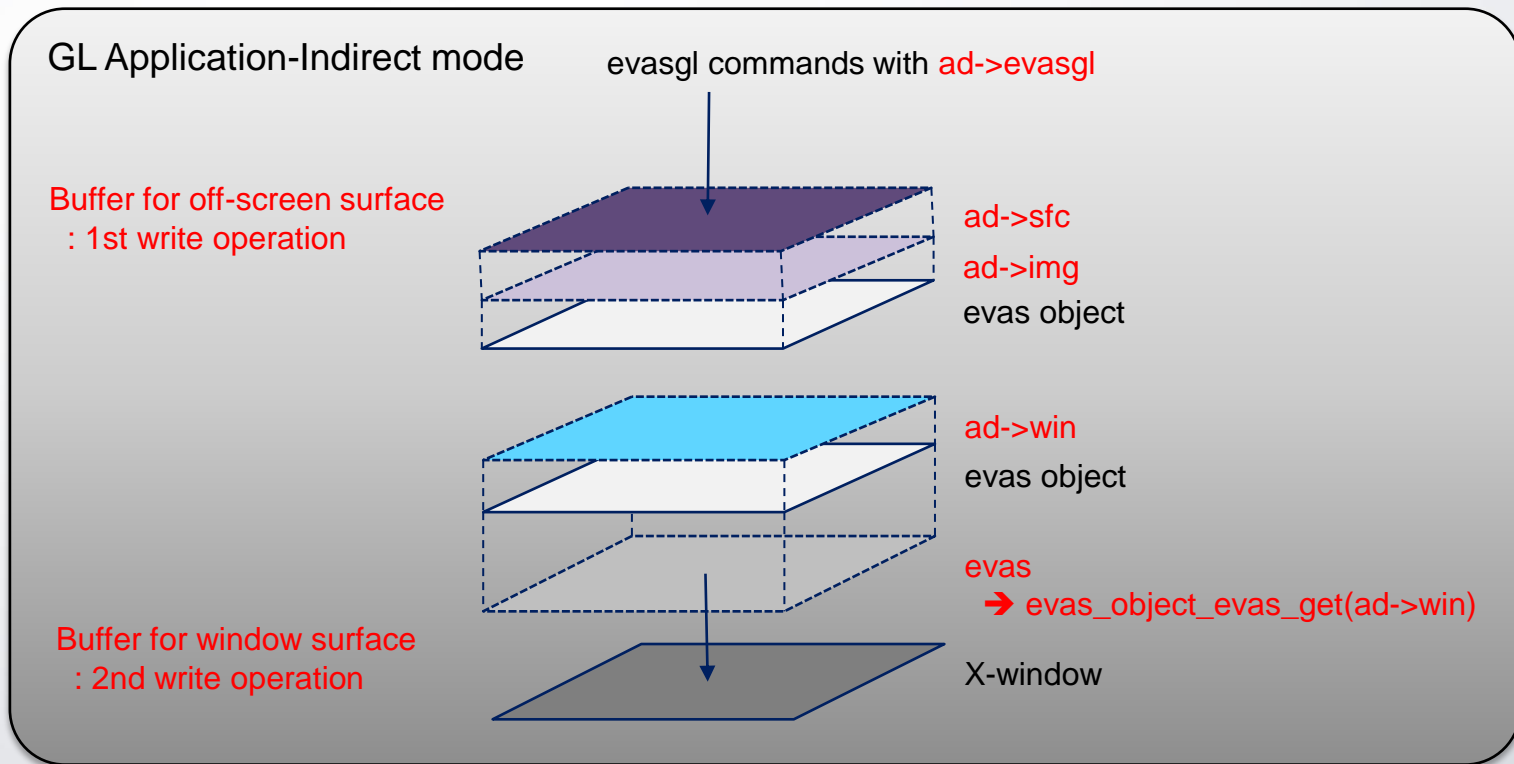
        elm_glview_init_func_set(glview, _init_gl);
        elm_glview_del_func_set(glview, _del_gl);
        elm_glview_render_func_set(glview, _draw_gl);
        elm_glview_resize_func_set(glview, _resize_gl);

        evas_object_size_hint_min_set(glview, 250, 250);
        evas_object_show(glview);
    }
    .....
}
```



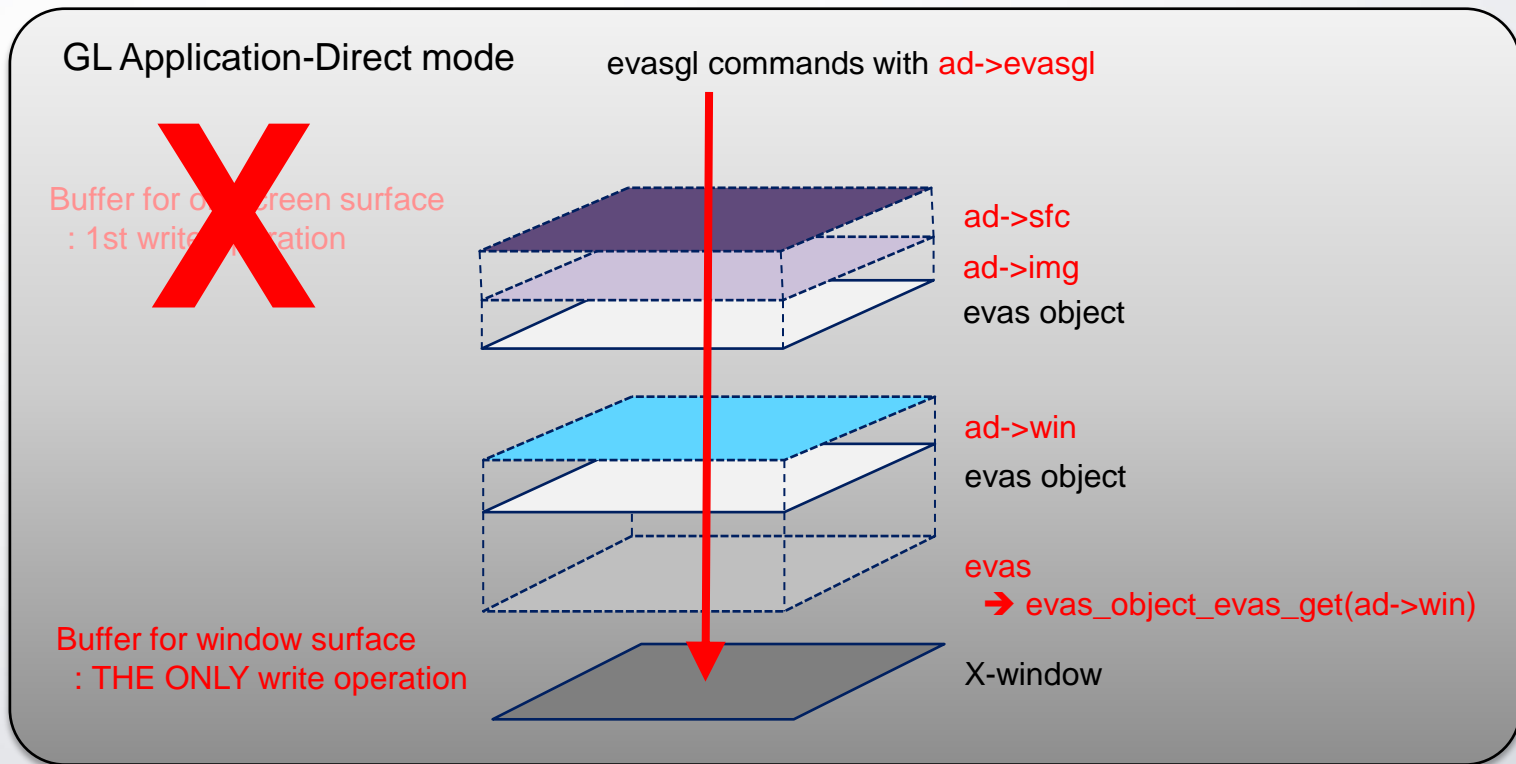
# Performance Improvement (1): DIRECT mode

- EFL View Hierarchy of full-screen GLES application



# Performance Improvement (1): DIRECT mode

- EFL View Hierarchy of full-screen GLES application



# Sample Code – Change Initialization

elm\_glview case

```
EAPI int elm_main(int argc EINA_UNUSED, char **argv EINA_UNUSED)
{
    .....
    /* Initialize & Setup elm_glview */
    {
        glview = elm_glview_add(win);
        elm_win_resize_object_add(win, glview);
        elm_glview_mode_set(glview, ELM_GLVIEW_ALPHA | ELM_GLVIEW_DEPTH | ELM_GLVIEW_DIRECT);
    }
    .....
}
```

evasgl case

```
/* Set config of the surface for evas gl */
.....
ad->cfg = evas_gl_config_new();
ad->cfg->options_bits = EVAS_GL_OPTIONS_DIRECT; // Configuration options (here, DIRECT mode on)
.....

/* Create a surface and context */
ad->sfc = evas_gl_surface_create(ad->evasgl, ad->cfg, w, h);
ad->ctx = evas_gl_context_create(ad->evasgl, NULL);
.....
```



# Sample Code – Change Initialization

elm\_glview case

```
EAPI int elm_main(int argc EINA_UNUSED, char **argv EINA_UNUSED)
{
    .....
    /* Initialize & Setup elm_glview */
    {
        glview = elm_glview_add(win);
        elm_win_resize_object_add(win, glview);
        elm_glview_mode_set(glview, ELM_GLVIEW_ALPHA | ELM_GLVIEW_DEPTH | ELM_GLVIEW_DIRECT
        .....
    }
```

evasgl case

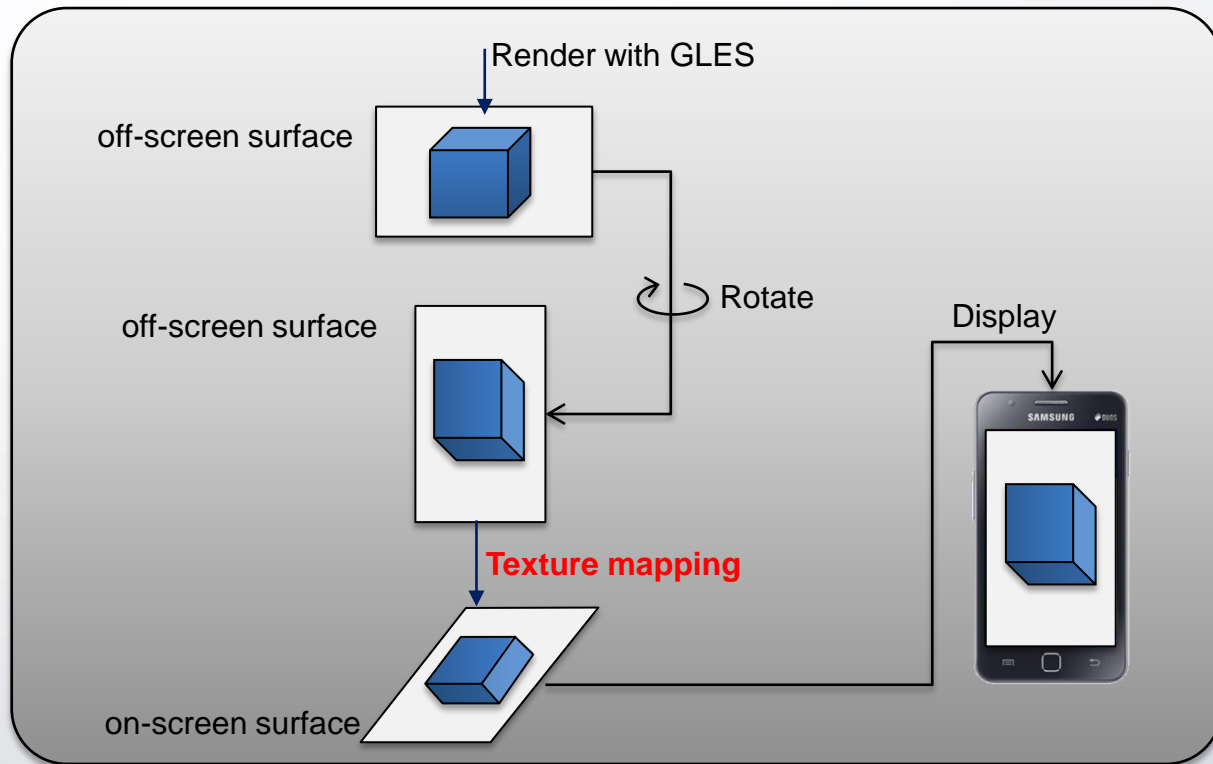
```
/* Set config of the surface for evas gl */
.....
ad->cfg = evas_gl_config_new();
ad->cfg->options_bits = EVAS_GL_OPTIONS_DIRECT Configuration options (here, DIRECT mode on)
.....

/* Create a surface and context */
ad->sfc = evas_gl_surface_create(ad->evasgl, ad->cfg, w, h);
ad->ctx = evas_gl_context_create(ad->evasgl, NULL);
.....
```



# Performance Improvement (2): Pre-rotation feature

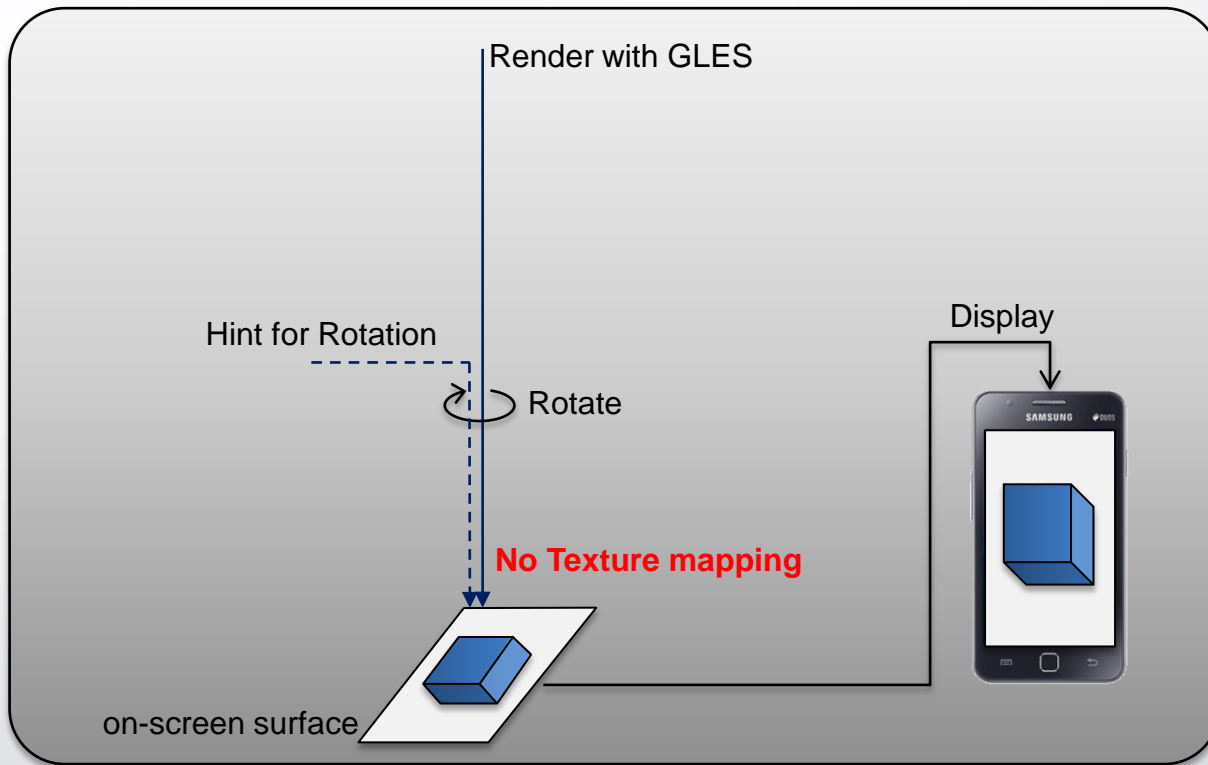
- Landscape typical way:
  - Use Intermediate off-screen Surface for Rotation





## Performance Improvement (2): Pre-rotation feature

- Landscape efficient way:
  - Pre-rotation which does not need the Intermediate Surface



# Pre-rotation in evasgl (1)

- How to use the feature?
  - Just turn-on DIRECT mode
  - Requirements
    - GPU Driver must supports pre-rotation feature
    - When GPU does not support, then the rendering mode fallbacks to INDIRECT mode

Case 1: EVAS\_GL\_OPTIONS\_DIRECT mode

```
/* Set config of the surface for evas gl */
.....
ad->cfg = evas_gl_config_new();
ad->cfg->options_bits = EVAS_GL_OPTIONS_DIRECT; // DIRECT mode on
.....

/* Get rotation angle for developers */
angle = evas_gl_rotation_get(ad->evas_gl); // angle is zero, and there is nothing for developers to do
                                           // when pre-rotation is not supported,
                                           // render mode fallbacks to INDIRECT mode for LANSCAPE state
.....
```

## Pre-rotation in evasgl (2)

- Workaround for devices not supporting pre-rotation?
  - Rotate the scene by application side
  - *EVAS\_GL\_OPTIONS\_CLIENT\_SIDE\_ROTATION*
    - System is rotated (ex. touch), exception the on-screen surface

Case 2: EVAS\_GL\_OPTIONS\_CLIENT\_SIDE\_ROTATION

```
/* Set config of the surface for evas gl */
.....
ad->cfg = evas_gl_config_new();
ad->cfg->options_bits = EVAS_GL_OPTIONS_CLIENT_SIDE_ROTATION // DIRECT mode on,
                                                                // Rendering is always for Portrait
.....

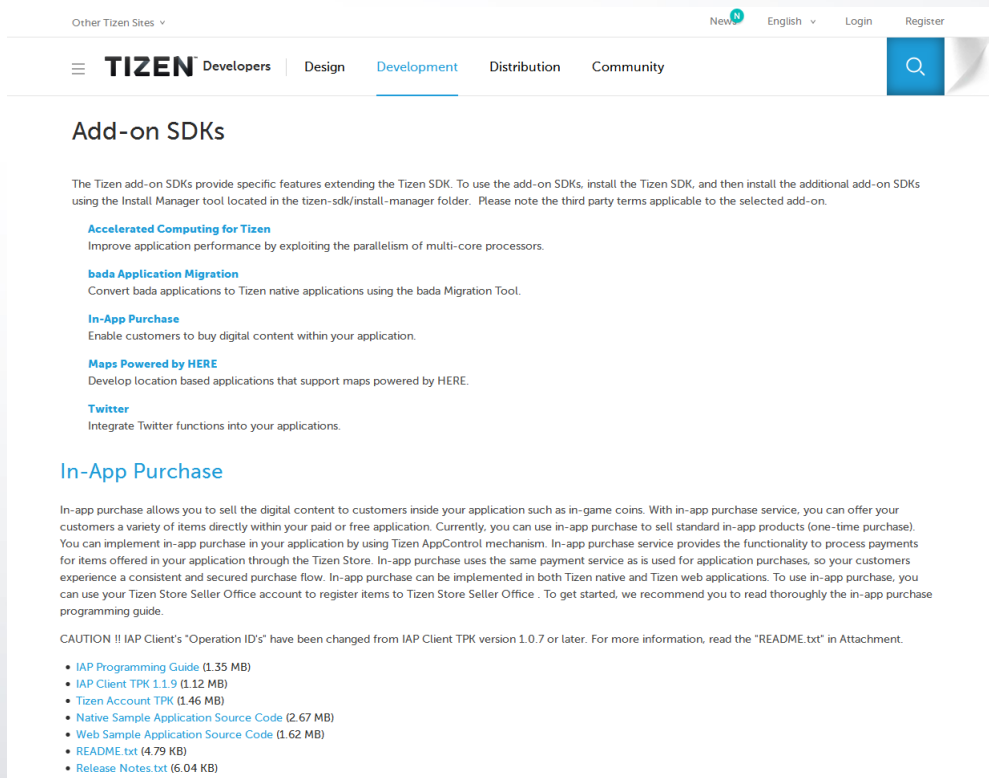
/* Get rotation angle for developers */
angle = evas_gl_rotation_get(ad->evas_gl); // angle shows the current device orientation
                                           // developers must rotate the rendered scene according to angle
.....
```

- Tizen IAP
  - IAP feature based on AppControl mechanism
    - You can borrow the functionality of TizenStore Client
    - There is no prerequisite in your projects
  - Basic work flow
    - Register items to Tizen Store Seller Office (<http://seller.tizenstore.com>)
    - Make your applications to work with IAP
    - Test and upload your application
  - Just check the 'IAP Programming Guide' and do IAP right now

# In-App-Purchase (IAP) in Tizen (2)

- Materials for IAP feature

<http://developer.tizen.org/downloads/2.2.1-add-on-sdks>



The screenshot shows the Tizen Developer website. At the top, there's a navigation bar with 'Other Tizen Sites', 'New', 'English', 'Login', and 'Register'. Below this is a main navigation bar with 'TIZEN Developers', 'Design', 'Development' (highlighted), 'Distribution', and 'Community'. A search icon is on the right. The main content area is titled 'Add-on SDKs'. It contains a paragraph about Tizen add-on SDKs and a list of links: 'Accelerated Computing for Tizen', 'bada Application Migration', 'In-App Purchase', 'Maps Powered by HERE', and 'Twitter'. Below this is a section titled 'In-App Purchase' with a detailed paragraph about the service. At the bottom, there's a 'CAUTION' note and a list of download links for IAP programming guides and sample applications.

Other Tizen Sites ▾ New English ▾ Login Register

≡ **TIZEN** Developers | Design Development Distribution Community 🔍

## Add-on SDKs

The Tizen add-on SDKs provide specific features extending the Tizen SDK. To use the add-on SDKs, install the Tizen SDK, and then install the additional add-on SDKs using the Install Manager tool located in the tizen-sdk/install-manager folder. Please note the third party terms applicable to the selected add-on.

- Accelerated Computing for Tizen**  
Improve application performance by exploiting the parallelism of multi-core processors.
- bada Application Migration**  
Convert bada applications to Tizen native applications using the bada Migration Tool.
- In-App Purchase**  
Enable customers to buy digital content within your application.
- Maps Powered by HERE**  
Develop location based applications that support maps powered by HERE.
- Twitter**  
Integrate Twitter functions into your applications.

## In-App Purchase

In-app purchase allows you to sell the digital content to customers inside your application such as in-game coins. With in-app purchase service, you can offer your customers a variety of items directly within your paid or free application. Currently, you can use in-app purchase to sell standard in-app products (one-time purchase). You can implement in-app purchase in your application by using Tizen AppControl mechanism. In-app purchase service provides the functionality to process payments for items offered in your application through the Tizen Store. In-app purchase uses the same payment service as is used for application purchases, so your customers experience a consistent and secured purchase flow. In-app purchase can be implemented in both Tizen native and Tizen web applications. To use in-app purchase, you can use your Tizen Store Seller Office account to register items to Tizen Store Seller Office. To get started, we recommend you to read thoroughly the in-app purchase programming guide.

CAUTION !! IAP Client's "Operation ID's" have been changed from IAP Client TPK version 1.0.7 or later. For more information, read the "README.txt" in Attachment.

- [IAP Programming Guide \(1.35 MB\)](#)
- [IAP Client TPK 1.1.9 \(1.12 MB\)](#)
- [Tizen Account TPK \(1.46 MB\)](#)
- [Native Sample Application Source Code \(2.67 MB\)](#)
- [Web Sample Application Source Code \(1.62 MB\)](#)
- [README.txt \(4.79 KB\)](#)
- [Release Notes.txt \(6.04 KB\)](#)

# Tizen IAP with Unity

- Unity Plugin for Tizen IAP
  - Integrate C-based Tizen AppControl into .NET-based Unity scripts

The screenshot shows the Unity Asset Store interface. At the top, the Unity logo and navigation links (Unity, Services, Showcase, Learn, Community) are visible. The 'Asset Store' tab is active. Below the navigation bar, the page title 'Tizen IAP (In App Purchase)' is displayed. The main content area features a large, stylized graphic of a white cube with blue and purple geometric shapes and the text 'In-App Purchase TIZEN'. To the left of the graphic, the following information is provided: Category: Scripting/Integration, Publisher: Samsung Mobile, Rating: Not enough ratings, Price: Free. Below this, there is a button 'Open in Unity' and social media icons. A note states 'Requires Unity 5.1.0 or higher.' and provides a description: 'This is the Tizen Store interface for In-App Purchases(IAP) for Unity Applications. For more information, please visit the site below. https://developer.tizen.org/downloads/add-on-sdks#In-App Purchase'. Below the description, the version '1.0 (Jul 02, 2015)' and size '44.3 kB' are listed. A row of three small images shows the plugin's integration into a Unity project. The 'Package Contents' section at the bottom lists the files included: Plugins, Examples, TizenSampleApp, Scripts, TizenSampleApp.cs, TizenSampleAppScene.unity, and README\_TizenIAPStore.txt. On the right side of the page, a search bar and a list of categories (Home, 3D Models, Animation, Applications, Audio, Complete Projects, Editor Extensions, Particle Systems, Scripting, AI, Animation, Audio, Avatar Systems, Camera, Effects, GUI, Input - Output, Integration, Modeling, Network, Physics, Video, Other, Services, Shaders, Textures & Materials, Unity Essentials) are visible.

- Brute force way to integrate Tizen IAP and cocos2d-x
  - Use Tizen AppControls in cocos2d-x app directly
- cocos2d-x plugin for Tizen IAP
  - cocos2d-x is open-source, and we are considering,
    - to integrate Tizen IAP to Plugin-x
    - to integrate Tizen IAP to SDKBOX

- Games for Tizen
  - New opportunity for business
  - Expandability, convergence and performance
- Porting to Tizen
  - Game engines help your joining to Tizen
- Basics and Tips for your development and optimization
  - evasgl and elm\_glvview
  - DIRECT mode and pre-rotation
- Monetization
  - Tizen IAP, and plugin supports



---

# Q&A

and **THANK YOU** for your time.

**Qingli Wang**

**qingli6.wang@samsung.com**