

# osgGap: Scene Graph Library for Mobile based on Hybrid Web App Framework

Youna Lee<sup>†</sup>, Seungmin Rho<sup>†</sup>, Jae-In Hwang<sup>‡</sup>, Heedong Ko<sup>‡</sup>, Junho Kim<sup>†</sup>

<sup>†</sup>Visual Computing Lab., Kookmin University

<sup>‡</sup>Imaging Media Research Center, Korea Institute of Science and Technology

## Introduction



Android Platform

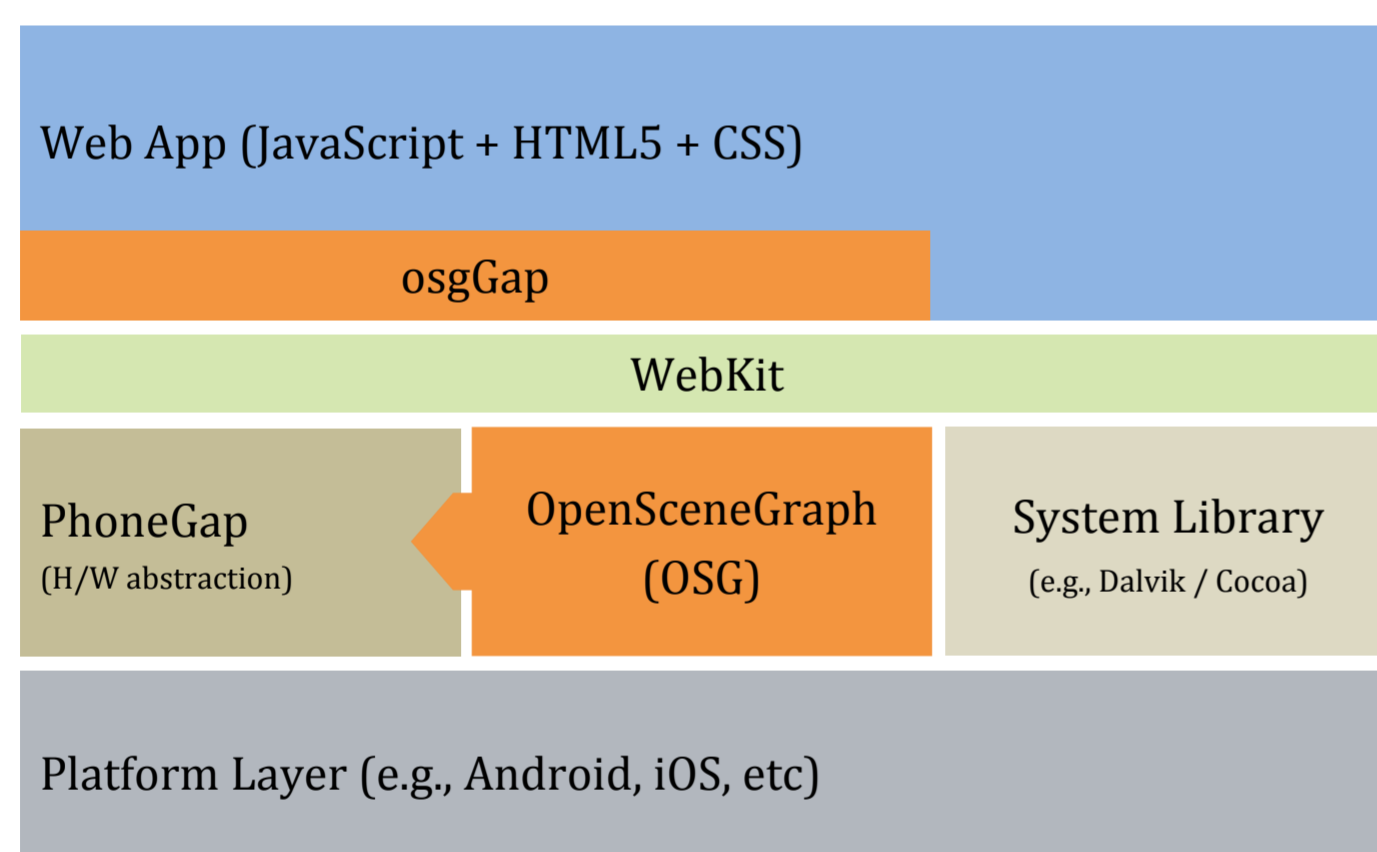
iOS Platform

The osgGap is an OpenSource JavaScript 3D scene graph library for mobile Web Apps based on hybrid mobile web framework. With osgGap, programmers can easily develop 3D applications for multiple mobile devices with web standards (i.e., JavaScript, HTML5, and CSS3), instead of using mobile platform-specific languages (e.g., Java for Android, Objective-C for iOS). To support web standards while providing high performances to 3D applications on mobile devices, we carefully design a bridge between the OpenSceneGraph as a scene graph engine in the native layer and our JavaScript functions in the web-app layer, via PhoneGap as a hybrid mobile application framework. As a result, we can provide the full OpenGL ES functionalities of mobile devices with high-level scene graph APIs in JavaScript.

## System Design

### 3D Scene Graph Library for Hybrid Mobile Apps

- Our goal is to design a high-level scene graph API that supports web-standards while providing high performances in mobile 3D apps.
- We have designed a bridge between the OpenSceneGraph as a scene graph engine in the native system layer and our JavaScript functions in the web app layer.



### Scene Construction with osgGap

- Example of osgGap that illustrates how a 3D scene is constructed with osgGap in the initial stage.
- The osgGap allow us to structure the scene graph at the web app level with JavaScript.

```
// Get the osgGap webview instance
var webView = osgGap.WebView;

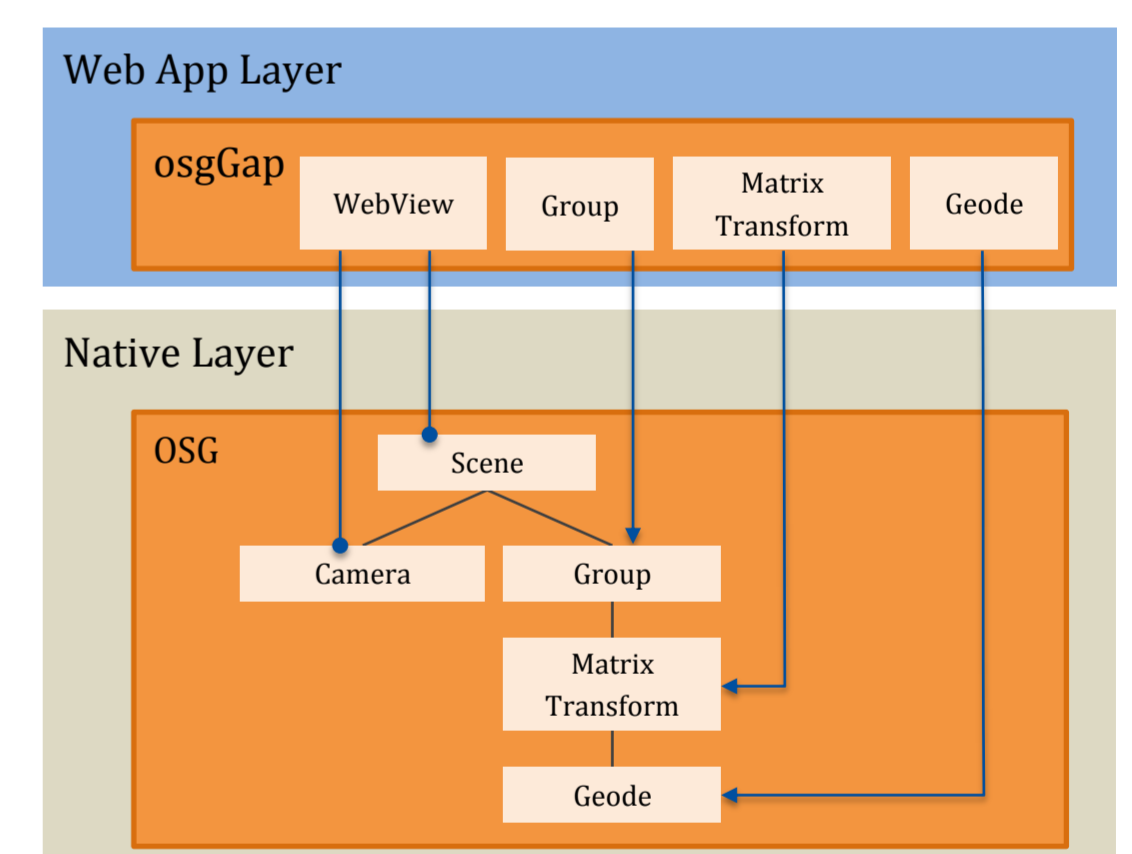
// Create nodes
var root = new osgGap.Group();
var geode = new osgGap.Geode();
var matrixTransform = new osgGap.MatrixTransform();

// Construct a scene graph
matrixTransform.addChild( geode );
root.addChild( matrixTransform );

// Specify the root node for the osgGap webview
webView.setSceneData( root );
```

### Scene Graph Synchronization

- Each node defined in the web app layer has a pointer to the corresponding node in the native layer.
- We synchronize the data into two layers in a way that the virtual nodes in the web app layer can access the actual nodes of the scene graph in the native layer.



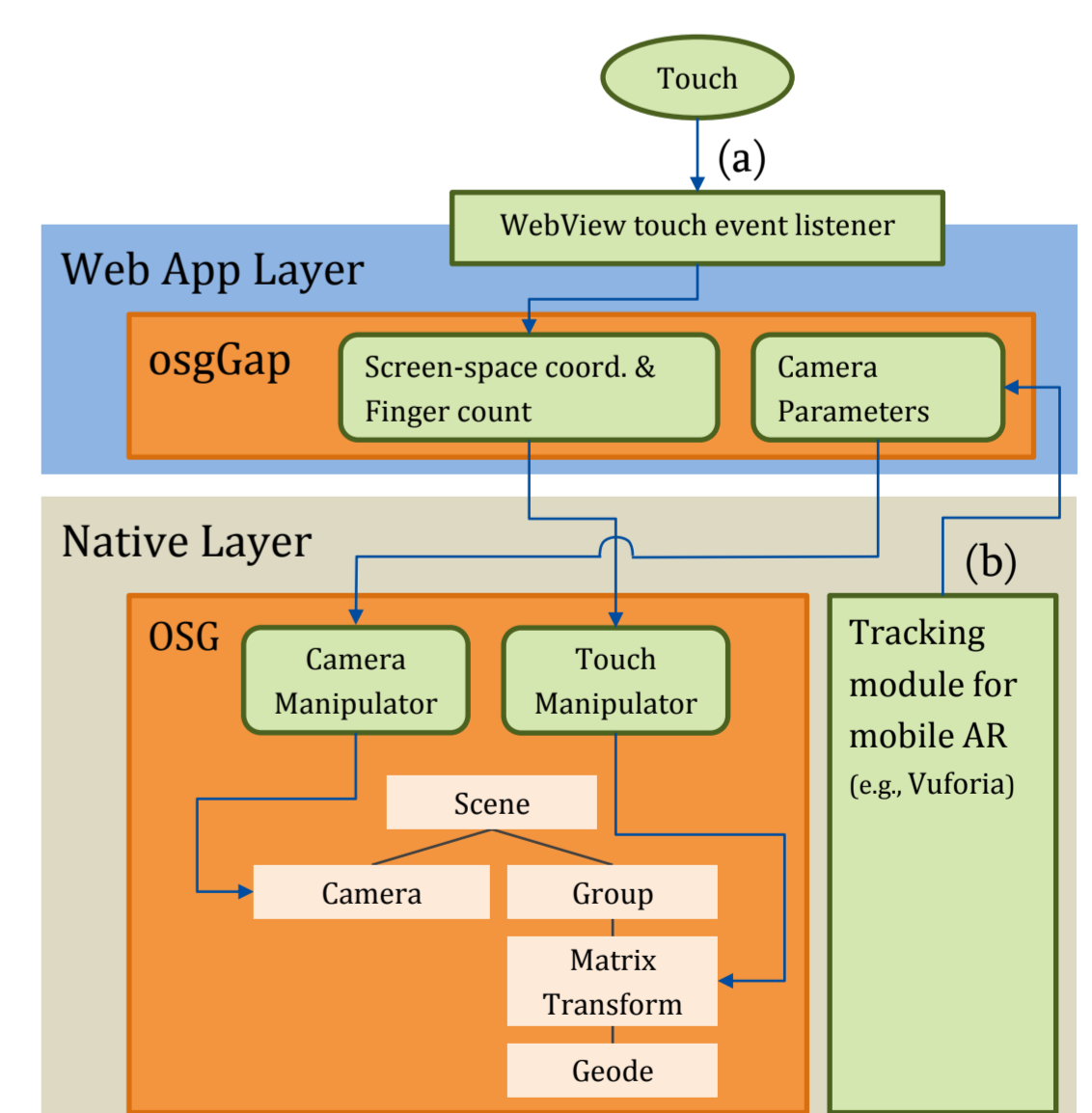
### Event Handling & Scene Manipulation

#### (a) Touch-based 3D interaction

- osgGap interprets the touch events from a webview as the finger gestures for 3D object interactions.
- osgGap receives the touch events from the event listener in a webview and sends the screen coordinates and a number of fingers related to the touch gesture to the manipulator node in the native layer.

#### (b) Scene navigation w/ external tracking modules

- In mobile AR applications, the 3D transformation information related to the camera node in osgGap should be manipulated by the motions of the physical camera in a mobile device.
- As the tracker module passes the camera parameters to the web app layer, the developer sets them for the virtual camera in osgGap that actually invokes the camera parameter updates in the native layer.



## Experimental Results

We perform the demos in two different mobile platforms; Android with the Samsung Galaxy Note 10.1 and iOS with the Apple iPad2.

Mobile Platform	No Interactions (fps)	User Interactions (fps)	
		Touch-based 3D interactions	Vision-based AR
Android	58	45	29
iOS	43	34	20

