OPEN SOURCE IMPLEMENTATION OF PORTING 3-D BLENDER GAMES ON ANDROID OS USING OGRE GAMEKIT.

Nuradeen Maidoki, Emmanuel Awa, Yu Xia, Gaspar Obimba and Timothy J. Hickey

Volen National Center for Complex System - Brandeis University
Graduate School of Arts and Science
Department of Computer Science

ABSTRACT

The blender open source project is providing a competitive alternative to developing 3-D games when compared to other 3D authoring tools, like (Unity, Maya, 3D-Max etc) that offer their products for commercial licenses especially their mobile platform integration. In order to allow blender users' share this experience, our paper presents a simple tutorial for porting blender games using Ogre Gamekit on android OS. This tutorial presents a step by step instruction on how to port blender games and ”checklist” requirements, needed for seamlessly transferring a blender 3D game (a *.blend file*) and porting it on the Android OS platform (a *.apk file*). We intend to extend it to the iOS platform in our future work.

***Index Terms*—** Blender, 3D Game Design, Mobile Games, Ogrekit Gamekit, Porting 3-D Games

1. INTRODUCTION

OgreKit is a cross platform open source 3D game engine for Windows,Linux,Mac, Android, iPhone and iPad. It has a graphic engine backend called ”Gamekit” for Ogre.

The Ogre graphics integration is developed in svn/trunk and has a physics engine consisting of Direct 3D/Open GL Libraries for rendering 3D scenes, powerful RTshader library GLSL for light effects and meshes. It compiles on Visual C++ in windows and on gcc3+ on linux and Mac OSX system. Fig 1. shows images of the Ogre from http://code.google.com/p/gamekit/. This proves the Game kit engine works, and will suit our purpose. This work currently covers an implementation using the Microsoft Windows platform.

This paper is organized as follows. The next section provides a checklist on requirements needed before installing and the OgreKit library and integration to the android development environment using eclipse. The third section provides the reader a step by step process for using Ogrekit. It provides details of our implementation on how to take a *.blend* file, copy it to the SD card of the Android device, and run the processes that generates the *.apk* file. Then it explains how to push the files to the device using the android *adb command*.

Fig. 1. Diagrams of Ogre kit game from [3]

The fourth section prevents the errors we encountered during the project. The errors include common compilation errors, svn errors, RTShader errors and so on. Finally we conclude with an overview of the entire project and the future work this work has opened up.

2. CHECKLIST REQUIREMENTS FOR USING OGREKIT AND ANDROID ON WINDOWS OS.

Currently, we have an implementation that is tested using the Windows OS. The following are applications needed for using the Ogrekit.

- Subversion Client: SVN client is required for getting the ogrekit svn/trunk from the gamekit website (www.gamekit.org)

- Microsoft Visual C++ compiler: This is required to set up the vc environment settings to allow our cmake program compile and make our libOgrekit.so file. see microsoft website (www.microsoft.com)

- CMake: CMake generates native makefiles and workspaces that can be used in the vc compiler environment of Microsoft Visual Studios. Download using http://www.cmake.org

- Eclipse: GUI automated authoring tool using an extensible development platform, runtimes and application frameworks for building, deploying and managing Java/Android. Download in eclipse website http://www..eclipse.org

- Android SDK: Android SDK is the IDE for android development. it has a powerful plug-in available on the Android developer site for facilitating Android development. Get the android SDK platform (http://developer.android.com)

- Android NDK: Android NDK is a companion tool with the SDK that allows user to perform critical operations using native code using c/c++ and are packaged in ,apk file. use a stable version android-ndk-r5b is the most stable version for compiling android with ogrekit. download android-ndk-r5b [4]

3. STEP BY STEP INSTRUCTIONS

1. Download the android-ndk-r-5: Get the android-ndk folder you downloaded from the android website and put it in a gamekit folder *(C:\gamekit)*
2. Set Environment Settings: Environment settings sets our path to NDK file path so that we can compile our OgreKit for android use using c/c++. We perform the various steps when setting our environment variables:
   - set NDK=C:\gamekit\android-ndk-r5c
   - set NDK_BIN = %NDK%\toolchains\arm-linux-androideabi-4.4.3\prebuilt\windows\bin
   - set PATH = %NDK_BIN%;%PATH%
3. Get the gamekit sources from the svn/trunk: Gamekit svn houses the gamekit source files and trunks. We require SVN client (eg. Tortoise SVN) to extract the svn contents from the gamekit source code directory. To check out the files we need to install the SVN client and perform the checkout using:
   - cd C:\gamekit
   - svn checkout http://gamekit.googlecode.com/svn /trunk/ src
4. Using CMake for compiling, making and building libogrekit.so:Launch the cmake compiler program and perform the following steps. This sets our cmake environment for nmake and build in microsoft visual c++ environment.
   - Press tools→configure
   - Set native compilers to Visual C vcvars environment.
   - switch the advanced view to simple view (located at the choice box close to "add entry" button
   - set the source_code file path to the gamekit src folder i.e. c:/gamekit/src
   - set your build_path binaries to c:/gamekit/build_android
   - Configure your cmake environment by clicking "configure" button.
     (a) You would get a lot of options highlighted in red.
   - Set the make compilation to Microsoft visual Studios\vcvars
   - Set the variable path as shown below:
     (a) set CMAKE_BUILD_TYPE = MinSizeRel
     (b) set CMAKE_CXX_COMPILE = "C:/gamekit/android-ndk-r5b/toolchains/arm-linux-androideabi-4.4.3/prebuilt/windows/bin/arm-linux-androideabi-g++.exe"
     (c) set CMAKE_C_COMPILE = "C:/gamekit/android-ndk-r5b/toolchains/arm-linux-androideabi-4.4.3/prebuilt/windows/bin/arm-linux-androideabi-gcc.exe"
     (d) Configure the cmake environment again to set the render, RTshader and openGLES configurations. Perform the following steps.
       - set the OGREKIT_BUILD_ANDROID by clicking the checklist
       - Click "Configure"
       - set CMAKE_BUILD_TYPE = MinSizeRel
       - set the OGREKIT_BUILD_GLES2RS by clicking the checklist
       - set the OGREKIT_MINIMAL_FREEIMAGE _CODEC by clicking the checklist
       - set the OGREKIT_OPTIMAL_SOUND by clicking the checklist
       - set the OGREKIT_UNITY_BUILD by clicking the checklist
       - set the OGREKIT_UPDATE_LUA_DOCS by clicking the checklist
       - set the OGREKIT_RTSHADER by clicking the checklist
       - set the SAMPLE_ANDROID_TEST by clicking the checklist
       - set the SAMPLE_RUNTIME by clicking the checklist
     (e) Generate the cmake settings to create the cmake list.txt and the OgreKit libsogrekit.so files. You would notice the following in your build_android folder and src folders.
5. Make and build the libogrekit.so file: To make and build the libogrekit.so file, you will require to set the vcvars settings and chmod your config file for compilation and nmake perform the following steps:
   (a) Set your vc+ environment settings using the command line on the folder. Type "C:\Program Files\Microsoft Visual Studio 10,0\VC\...\vcvars32.bat

(b) Use the command ”chmod a+x ./configure-android”

(c) Go to ”cd../build_android folder” and ”make”

(d) If you are successful libogrekit.so will be built in your ⟨src⟩/Samples/AndroidDemo/Demo/libs/armeabi.

(e) If not you might be missing some steps [refer to our video tutorials on our blog]

(f) Handling Android Development Application Logic. After successfully building the liborgekit.so, you are ready to use eclipse for the android implementation perform the following steps before setting your ⟨src⟩/.../AndroidDemo” files in eclipse.

   i. install jdk
   ii. install the android sdk
   iii. install eclipse ide and set up the android-sdk plugins on eclipse [see android tutorials for this or our video tutorials]

(g) Connect your phone to the computer and transfer the momo_ogre_i.blend file (which is a sample file to your phone’s media storage device). The momo_ogre_i.blend can be found in your src folder: ⟨src⟩/Samples/IPhonDemo/momo_ogre_i.blend. copy and paste into your phones sd card.

(h) Launch the eclipse environment and create a new Android project by performing the following steps:

   i. Select ”menu →file →New/Android Project”
   ii. Select ”create project from existing source”
   iii. Set Location to ⟨src⟩/Samples/AndroidDemo/Demo
   iv. Android Files would be visible in your package explorer
   v. Select your Target Device
   vi. Run android file
   vii. The Android Simulated Phone would launch with the momo_ogre monkey!
   viii. Enjoy your OgreKit by changing blend files in your sdcard and updating location in your org.gamekit.jni/main.java file

4. HANDLING COMMON ERRORS (FAQ)

1. I cant download the SVN files from the gamekit site?

   The SVN error can be caused by several things but the 3 main errors are:

   - Download SVN client: You need SVN client to check out files.
   - SVN commands: ensure you type the right svn commands by typing ”svn co http://gamekit.googlecode.com/svn/trunk/src”
   - Revision Error: Revision Error occurs if we dont completely check out our files.

2. Problem linking libogrekit.so?

   Problems with building our ogrekit are caused by cmake file errors. this happens during make and build on your SVN trunk files are incorrectly built. To solve the problem:

   - You need to make sure you have a stable android-ndk-version (I prefer android-ndk r5b).
   - You need to ensure your svn files are checked out completely and all environment settings including the vc environment path are up.
   - Ensure on launching the cmake program, you set the cross compilation to microsoft vc settings, and all build and source directories point to the right folder locations. ”source_dir: ”c:/gamekit/src” , build:”c:/gamekit/build_android”
   - Ensure parameters for configuration of cmake programs are correct

3. Having Problems compiling gamekit for android?

   Sometimes even after successful compilation, Android crashes when you run the emulator or a phone. We also get Application initiliazation error.

   Perform the following process:

   - Ensure your SD card has the blender file you want to port to your android phone using the *adb* push command or copying blender file to the sd card
   - Check your main.java and ensure it points to your blend file
   - If using phone detach from USB before testing it.

4. My graphics are dark and jaded?

   Graphics are dark and jaded due to renderer problems caused by the ogreRTshader library and the OPEN GLES, GL library.This are caused by either build problems or in our org.com.nvdia library. sometimes the problem could be from a higher android version.

   - We recommend to target devices with the range of 7-10 (Android 2.0.0 →Android 2.3.3)
   - Try rebuilding your ogrekit src it might be due to build problem.

5. CONCLUSION AND PERSPECTIVE

The art of Game Design and development should be carried out without fear of what platform the game shall be rendered on. We have shown that games developed with Blender could easily be ported to smart mobile devices with minimal integration effort from the game developer. The easy porting of blender games that we have shown in this tutorial serves as an incentive to game designers, who are not mobile developers, to quickly develop a game of their choice and rest assured that it will work on mobile devices. Continued research in this field will yield even better means for porting the games to devices.

6. FUTURE WORKS

For the fact that our project is on-going, we do not envisage so much future work beyond the primary goal. They are as follows:

- We plan to develop the same strategy to port games to iOS smart devices, kindle and Blackberry
- Complete coding the developmental model, interface it and obtain accurate results.
- Automate the steps and provide a website that lets non tech savvy game designers upload the .blend files into a URL that will yield the required smart device executable files.
- Our future work include providing a real-time porting as a service model that allows blender developers convert their .blend files, .apk files.
- We also want to see how we can integrate OpenSL sounds library on the android platform and other engines that blender supports which are not currently supported by the OgreKit Library.

7. REFERENCES

Ogre3d *Open Source 3D Graphics Engine* - http://www.ogre3d.org/ [Online Accessed last: 10th May, 2012]

[4] [1] *Porting Blender to Mobile Devices* - http://letsportblendertomobile.wordpress.com/ [Online Accessed last: 10th May, 2012]

[2] *Gamekit Forum* - http://gamekit.org/forum/ [Online Accessed last: 10th May, 2012]

[3] *A cross-platform 3D game engine using Ogre or Irrlicht and Bullet for Windows, Linux, Mac, Android and iPhone* - http://code.google.com/p/gamekit/ [Online Accessed last: 10th May, 2012]

[4] *Android Developers Website* - http://developer.android.com/