Generating and Building

Bootstrapping

Introduction

There could be multiple brenaches of code on same machine. Also depending on which OS we are, different tools might be available. Thus there is a bootstrapping mechanism to setup environment for particular branch on particula OS whenever we want to work in the branch.

We will run our tools mostly through Cygwin. So whenever we open a new Cygwin console we need to setup environment so that tool commands are recognized and tools can run.

For example, there is environment variable

PYENGINE_WORKSPACE_DIR

(the name is legacy and at some point will be renamed to PRIMEENGINE_WORKSPACE_DIR)

Some tools rely on this environment variable to be set to know where to look for/store to files.

For example, a FBX parser tool needs to save assets to AssetsOut folder in PEWorkspace, so it will use the PYENGINE_WORKSPACE_DIR environmnet variable to know where AssetsOut/ is on disk.

There can be a lot of things setup in boostrapping, like some tools can differ for Windows vs OS so they will be aliased to different locations (this is the case for premake)

Note, important folders are set in this file, like:

export MAYA_DIR="C:\\Program Files\\Autodesk\\Maya2014"

export VSDIR="C:\\Program Files (x86)\\Microsoft Visual Studio 10.0\\Common7\\IDE"

if your machine for whatever reason uses different folders for the programs, change the values in the script, but don't submit the file. (A good solution for this would be to potentially declare these vars in .bashrc script and only create them ir. env.sh if they dont exist)

0) Open Cygwin (each time we open cygwin, the envrionment is prestine, so we need to run bootstrapping scripts). [Possible task: add a shortcut to initialize Cygwin from a particular branch that automatically bootstraps environment)

1) Navigate to PEWorkspace/Tools folder

2) execute $ source setenv.sh

setenv.sh is the boostrapping scripts. Once it is sourced, the Cygwin instance has envrionment variables and aliases setup

Premake: Generating Visual Studio Solutions

Premake is a open source tool that can generate Visual Studio solutions by running a premake script (that we write).

There is a master premake script for the whole solution and a child premake scripts for each project.

premake.sh

There is a helper script in Tools folder called premake.sh. Inside it sources the setenv.sh, changes folder to Code, executes premake with all the arguments passed in into script and changes folder back to Tools:

****/Tools $ ./premake.sh --platformapi=win32gl vs2015

or

****/Tools $ ./premake.sh --platformapi=win32d3d9 vs2015

or

****/Tools $ ./premake.sh --platformapi=win32d3d11 vs2015

This will generate solution for open gl , d3d9, d3d11 versions of code for visual studio 2010

To see other options for platformapi you can execute

****/Tools $ ./premake.sh --help

Details:

Our premake scripts currently are:

PEWorkspace/Code/premake4.lua - for the solution. inside it includes the sub premake scripts

PEWorkspace/Code/PrimeEngine/premake4-primeengine.lua

PEWorkspace/Code/luasocket_dist/premake4-luasocket_dist.lua

PEWorkspace/Code/CharacterControl/premake4-charactercontrol.lua

PEWorkspace/Code/Basic/premake4-basic.lua (todo)

Important Details on Usability

1) Don't have spaces in your path to PEWorkspace

2) Use notepad++ to edit setenv.sh in case you need to change path to Maya. Don't use vi

3) Don't forget that full premake command has **vs2010** in it, so ./premake.sh --platformapi=win32d3d9 **vs2010**

4) There is an error message printed whenever we launch things from cygwin : ShellExecute(NULL,: command not found.

Just ignore it

5) When you run ./rundevenv.sh, on windows, it will launch visual studio **without** a solution open. This is expected because you can generate 5 different solutions on windows. On OSX it opens XCode with solution open, because there is only one solution on OSX

.

6) Install Windows SDK on C:\ drive or you will need to modify premake scripts or Visual Studio projects to have correct folders for include folders and library folders

7) When launching XCode through script, make sure XCode is not already running in hte background. If it is, it will not have the environment variable set.

Building

Once the visual studio / xcode solution is generated you need to run rundevenv.sh script from Tools folder in cygwin. Running it from console allows environment variables to be set and used in application build phases.

execute

****/Tools $ ./rundevenv.sh

On windows, this will open visual studio. From visual studio navigate to Code folder and open whichever solution you want.

O X, this command will open XCode and automatically the iOS solution, since there are no other solutions available on the OS

Normal Use Case

In general, you will likely be in Tools folder in Cygwin and execute premake.sh and rundevenv.sh. Note if Visual Studio or XCode is already open with a solution that you are generating with premake.sh, it will ask you if you want to reload it, so you don't need to close/open the development environment.

Running

To run the game, you just need to compile (F7) and run (F5) the solution.

In XCode, just press play button