CryEngine 5.2.1/5.2.2 Linux Build Instructions

October 10, 2016

These are a set of instructions for compiling and running CryEngine on Linux that have worked for me. I am sharing them so that others attempting to get the engine running on Linux might benefit from them. Note that the Sandbox editor is not currently available for Linux.

Also note that the Linux version of CryEngine does not appear to be fully functional. I am able to run on Debian 8.6 using a GTX 660 with the NVIDIA proprietary driver (version 370.28). Although gamezero and airfield run there are various issues with the Linux version of the engine. But... if you are willing to go through this process, you may also be willing to debug. :)

These instructions are specific to building CryEngine 5.2.1 and 5.2.2 on Linux. They may not work for other versions. They have been tested on Debian 8.6 with gcc 4.9.2. These steps should get things compiled and running gamezero and airfield. However, you may run into different issues with your distribution.

If you only want a Linux binary and don't want to build it from source yourself, you should follow the steps in section 1 to download the launcher and engine for Windows. Then copy the CRYENGINE_5.2 folder from Windows to Linux and run the released Linux version of the engine.

Copy the following Windows folder to Linux:

```
C:\Program Files (x86)\Crytek\CRYENGINE Launcher\Crytek\CRYENGINE_5.2
```

Then run the engine:

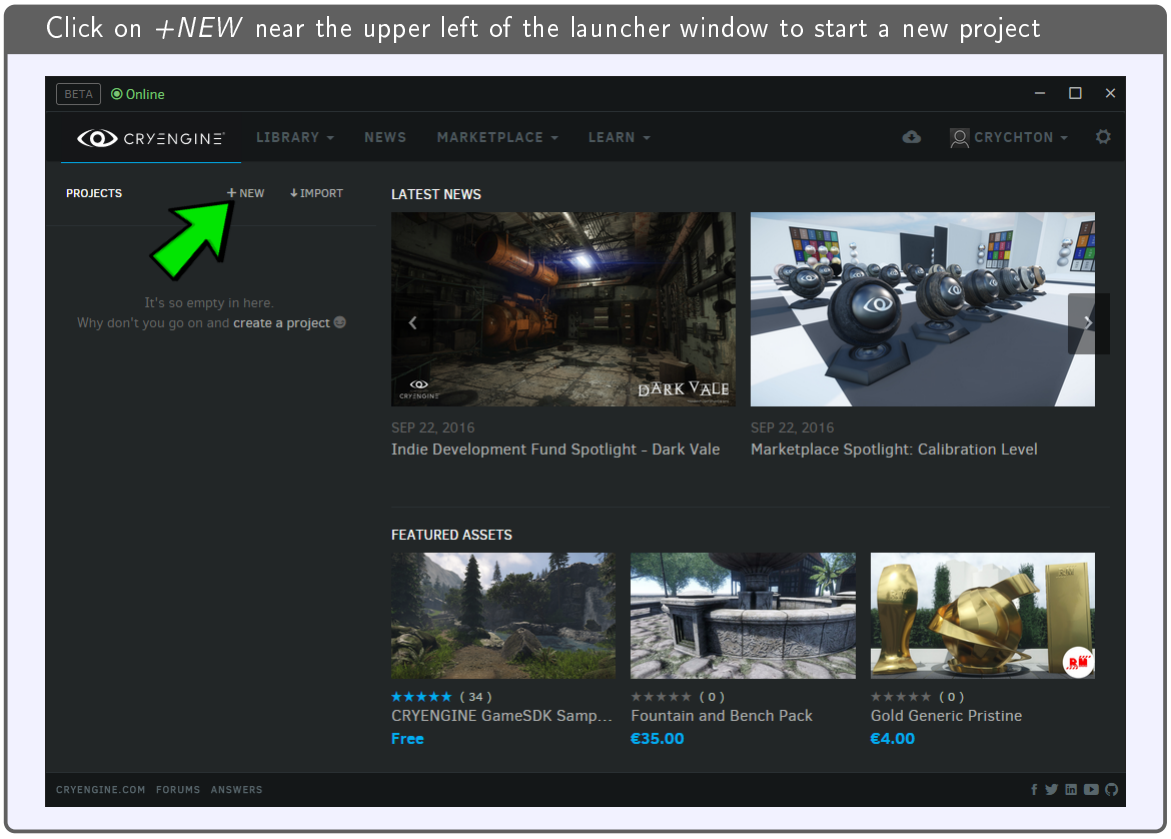
```
$ cd CRYENGINE_5.2
$ ./bin/linux_x64_gcc/GameSDK
```

This should run gamezero which allows you to fly around a set of spheres on a small island. You can skip to the section on installing the Game SDK to get the airfield demo running.

1 Windows

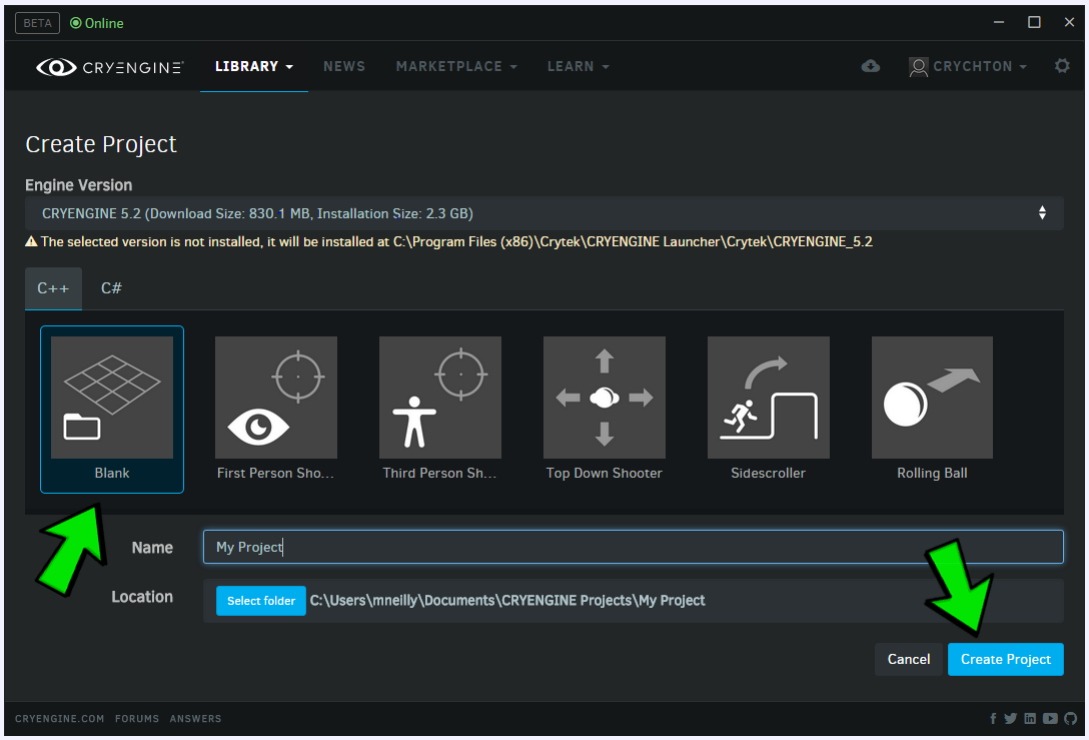
Download and install CryEngine from cryengine.com. You'll need some asset files from that download.

- Download the CryEngine launcher from `http://cryengine.com`
- Start the launcher
- Click on *+NEW* near the upper left of the launcher window to start a new project



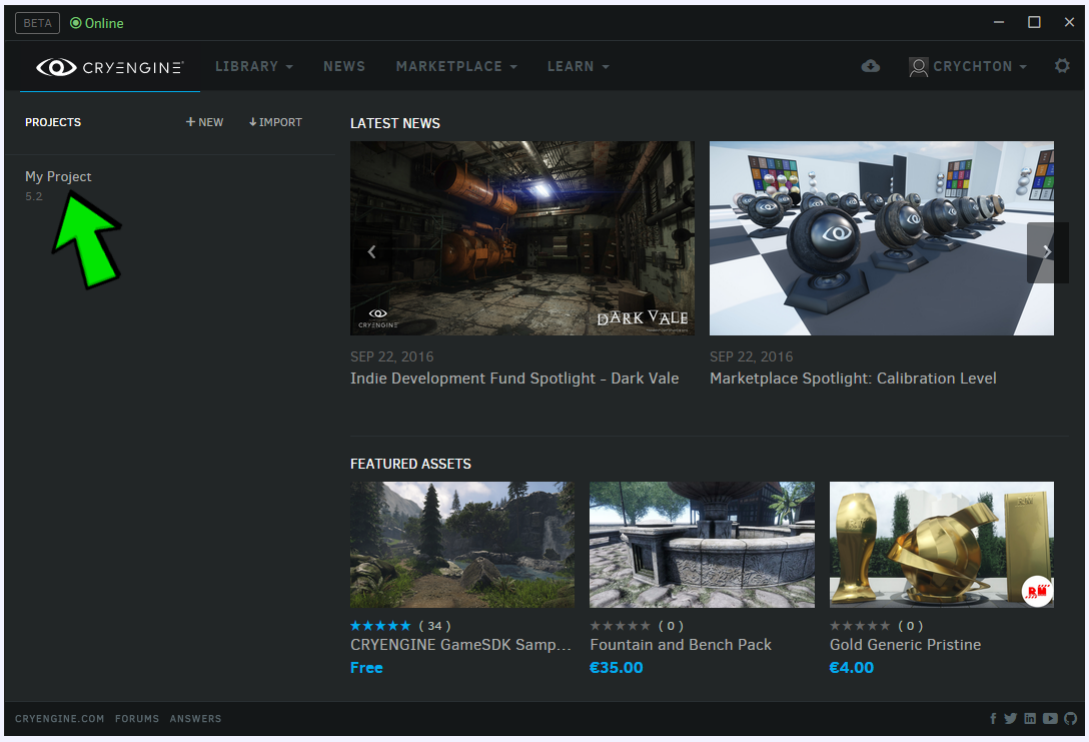
- Select the *Blank* project option and click create project

Select the *Blank* project option and click *Create Project*



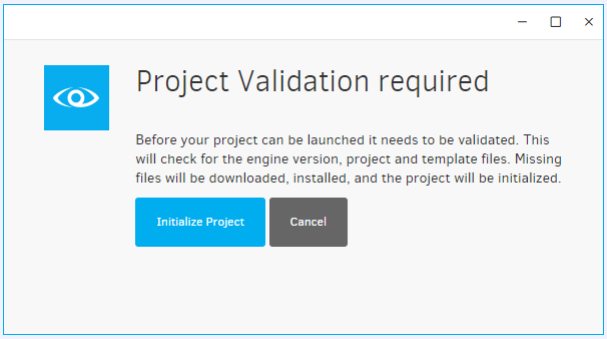
- Select your new project, *My Project*, from the project list on the left

Select your new project from the list on the left



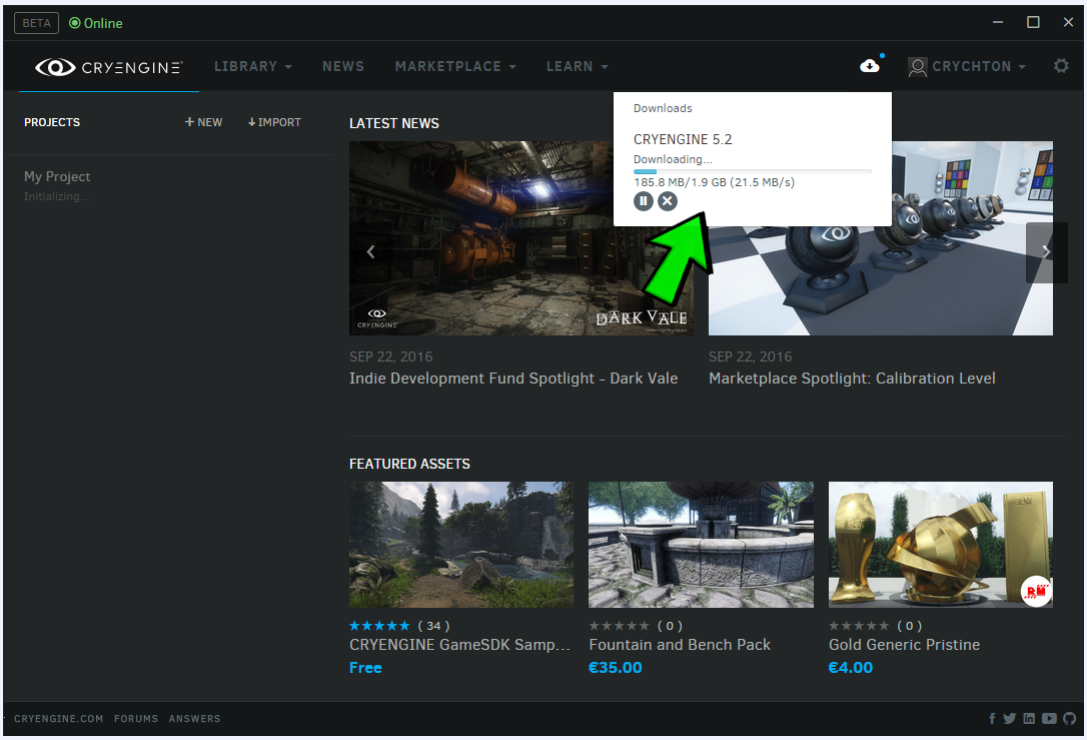
- Select *Initialize Project* in the following popup.

Click *Initialize Project*.



- CryEngine will be downloaded. While it is downloading you can move on to the Linux section.

CryEngine downloading.



The actual engine is only downloaded when you start a project and the launcher recognizes that you have not already installed an engine. There is no direct download for the engine. As new engines and new launchers are released you will be able to use the new engine for new projects while the older engines will remain available for projects using the older engines.

2 Linux

Download the CryEngine source from GitHub:

```
$ git clone https://github.com/CRYTEK-CRYENGINE/CRYENGINE.git
```

If the git repository has advanced to a newer version of CryEngine and you still want to build 5.2.1 or 5.2.2, you can checkout the specific tag after the clone:

```
$ git checkout 5.2.1
```

Install the required SDKs. It appears that waf would normally download and install the required SDKs tarball. However, there appears to be a disconnect between the format of the SDKs file and waf. Although the file is suffixed as .tar.gz, tar is unable to read the file. Waf will fail in the next step but then you can use 7zip to manually install the SDKs.

```
$ cd CRYENGINE
$ ./cry_waf.sh -p gamesdk build_linux_x64_gcc_release
```

A popup will appear with gamezero selected. Click the confirm button. Then, waf will fail due to an issue with the SDK tarball. Manually extract the SDKs:

```
$ (mkdir -p Code/SDKs; cd Code/SDKs; 7z x ../../CRYENGINE_v5.2.0_SDKs.tar.gz)
```

The following one line change is one method to avoid a SEGV due to stack overflow. It is the easy method but not the best approach. However, it is probably fine unless you get to the point where CryEngine is fully functioning and you are low on memory. Presumably, a subsequent release will fix this issue. An alternative method is mentioned at the end of the document.

Hack to increase stack frame size

```
diff --git a/Code/CryEngine/CrySystem/CryThreadUtil_posix.h b/Code/CryEngine/CrySystem/CryThreadUtil_posix.h
index 4d0cca2..f9930df 100644
--- a/Code/CryEngine/CrySystem/CryThreadUtil_posix.h
+++ b/Code/CryEngine/CrySystem/CryThreadUtil_posix.h
@@ -162,7 +162,8 @@ void CrySetThreadPriorityBoost(TThreadHandle pThreadHandle, bool bEnabled)
//////////////////////////////////////////////////////////////////////////
bool CryCreateThread(TThreadHandle* pThreadHandle, const SThreadCreationDesc& threadDesc)
{
-	uint32 nStackSize = threadDesc.nStackSizeInBytes != 0 ?  threadDesc.nStackSizeInBytes :  DEFAULT_THREAD_STACK_SIZE_KB * 1024;
+	//uint32 nStackSize = threadDesc.nStackSizeInBytes != 0 ? threadDesc.nStackSizeInBytes : DEFAULT_THREAD_STACK_SIZE_KB * 1024;
+	uint32 nStackSize = 256*1024;

	assert(pThreadHandle != THREADID_NULL);
	pthread_attr_t threadAttr;
```

Build CryEngine:

```
$ ./cry_waf.sh -p gamesdk build_linux_x64_gcc_release
$ ./cry_waf.sh -p gamezero build_linux_x64_gcc_release
```

There will be errors about missing audio libraries. For now, they can be ignored.

Copy the entire *gamezero* folder and the *system.cfg* file from Windows:

```
C:\Program Files (x86)\Crytek\CRYENGINE Launcher\Crytek\CRYENGINE_5.2\gamezero
C:\Program Files (x86)\Crytek\CRYENGINE Launcher\Crytek\CRYENGINE_5.2\system.cfg
```

to your Linux *CRYENGINE* directory.

Copy all of the pak files from the Windows *engine* folder:

```
C:\Program Files (x86)\Crytek\CRYENGINE Launcher\Crytek\CRYENGINE_5.2/engine/*.pak
```

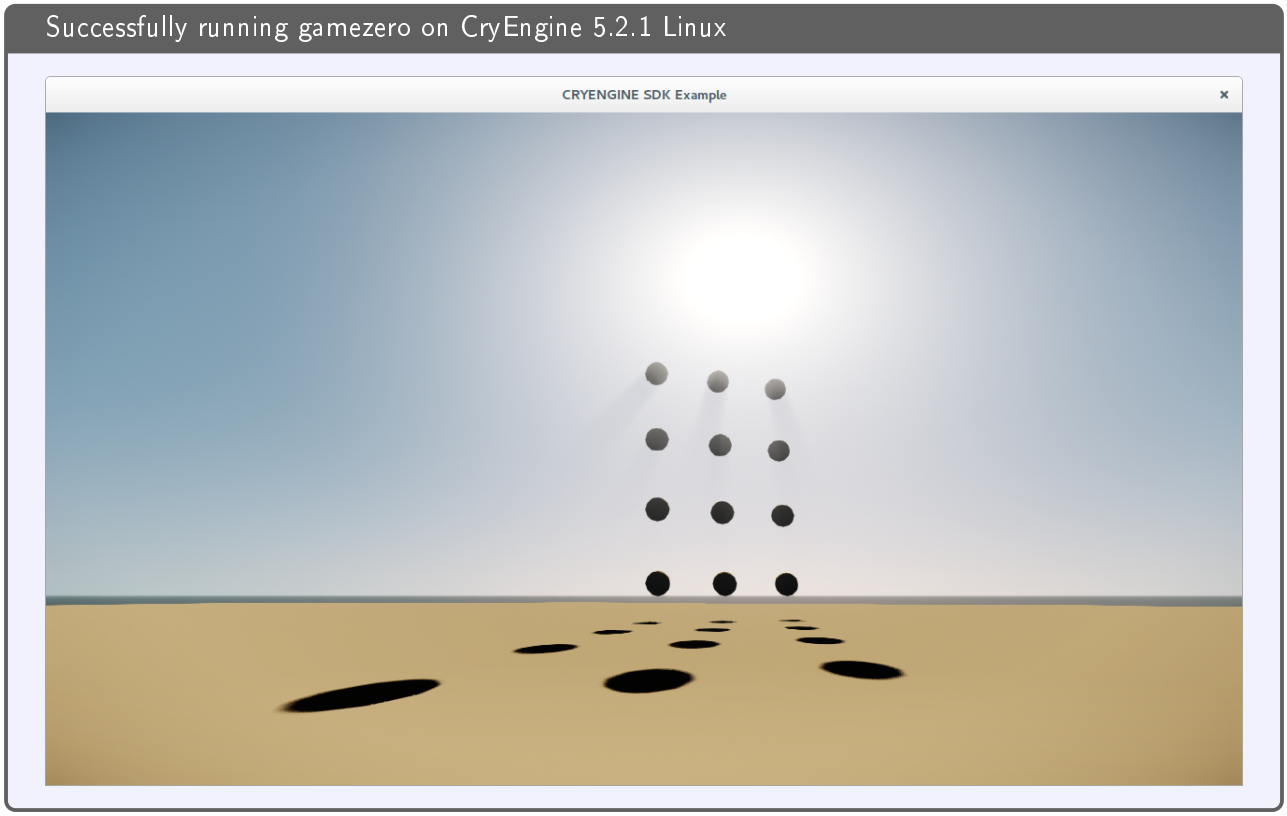
to your Linux *CRYENGINE/Engine* directory.

Run the engine:

```
$ ./bin/linux_x64_gcc_release/GameSDK
```

At this point you should have a working gamezero and see something like the following:

Successfully running gamezero on CryEngine 5.2.1 Linux



As mentioned earlier, there appear to be issues with CryEngine on Linux. I get different results each time I run. Sometimes the screen remains black, sometimes the spheres are missing. Sometimes the color map makes the whole thing look psychedelic. But it runs...

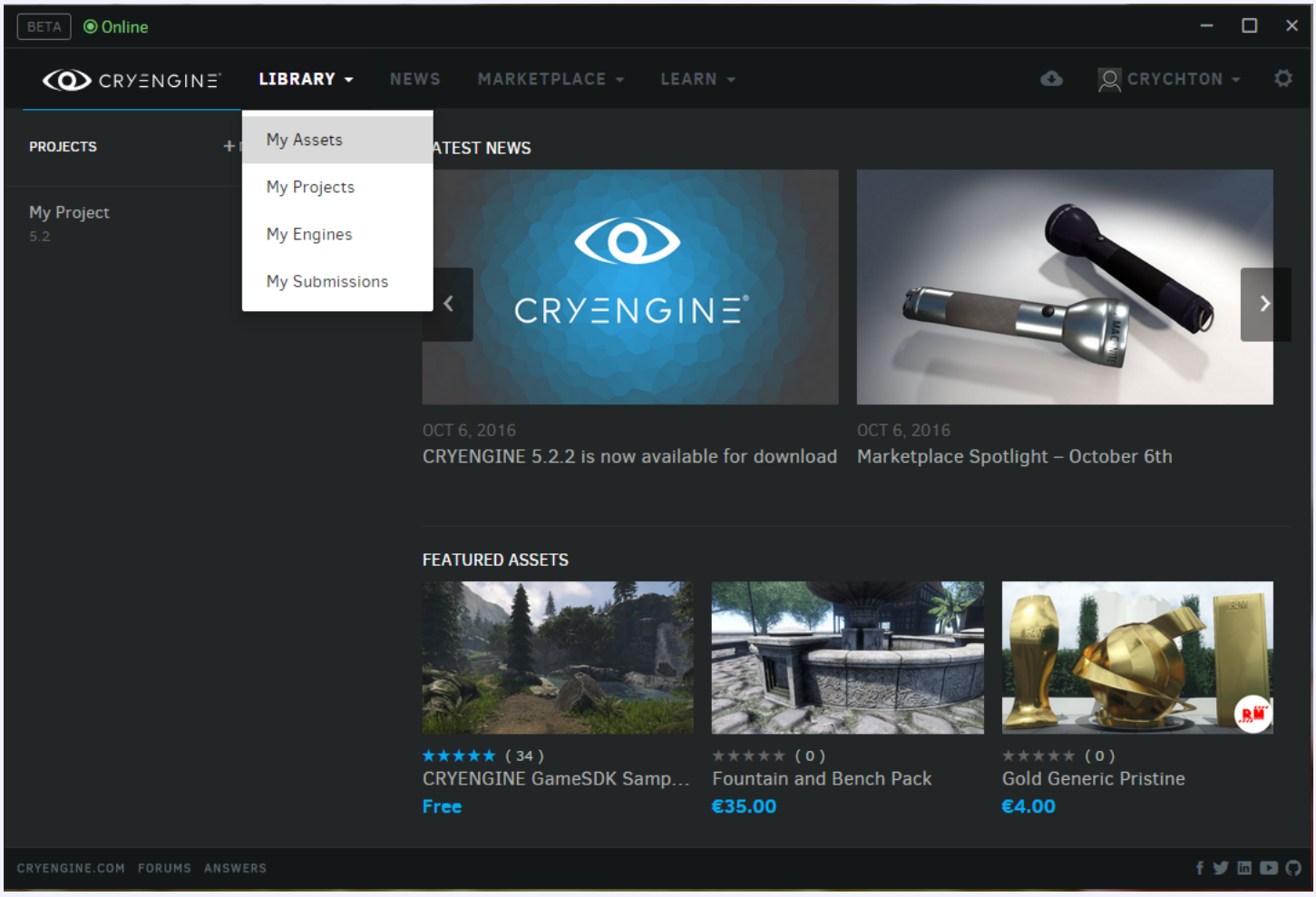
3 Airfield

On Windows, "purchase" the free Game SDK from the following link:

https://www.cryengine.com/marketplace/product/cryengine-gamesdk-sample-project

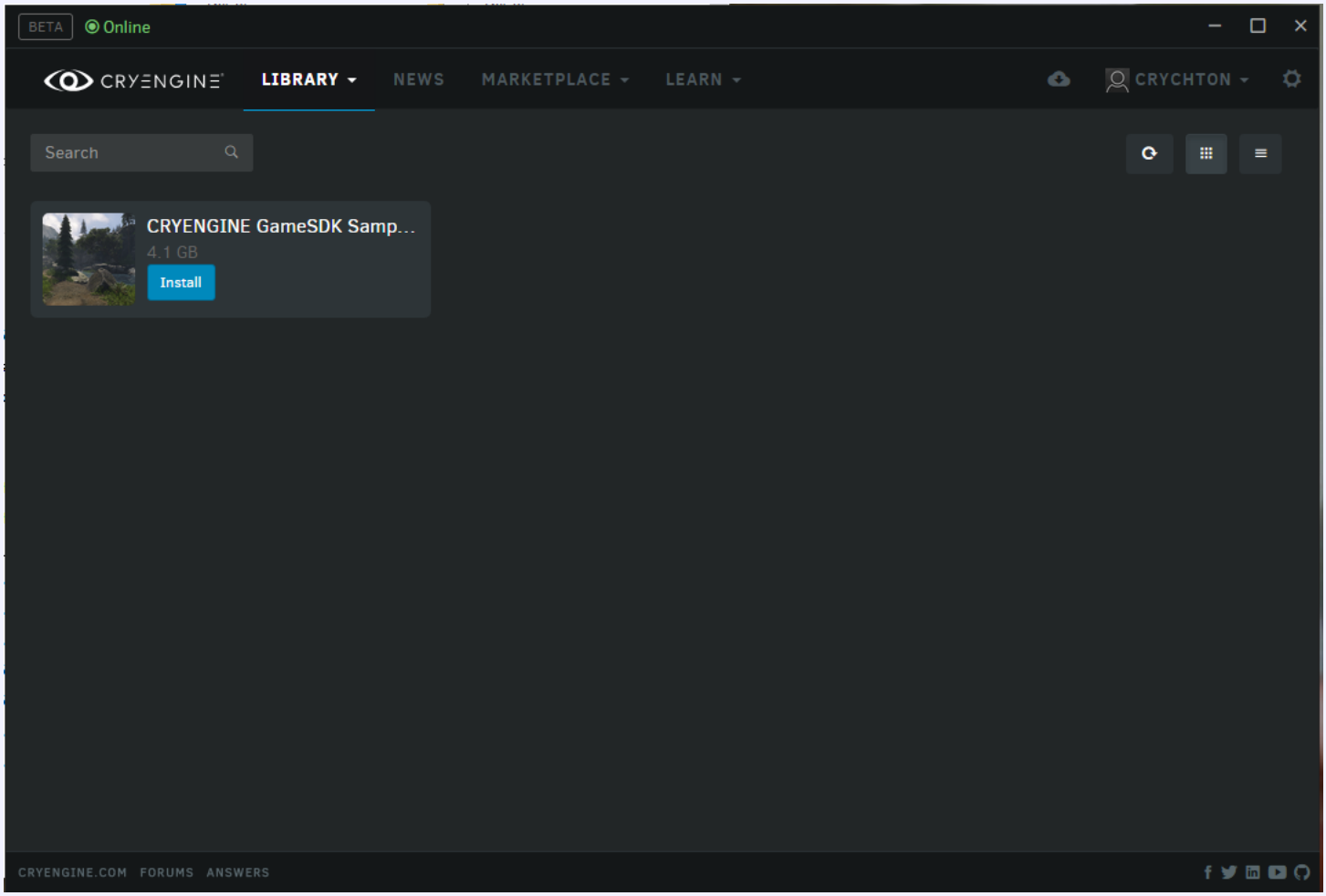
Then, in the launcher select the Library menu and My Assets:

Select *Library* then *My Assets*



Click install for the Game SDK:

Click on *Install*



After the SDK is installed copy the SDK from the following Windows location:

```
C:\Program Files (x86)\Crytek\CRYENGINE Launcher\Crytek\gamesdk_5.2
```

To the same directory that contains your *CRYENGINE* directory in Linux.

Edit *CRYENGINE/gamesdk_5.2/GameSDK/Assets/autoexec.cfg* to contain the following line:

```
map singleplayer/airfield
```

Edit *CRYENGINE/system.cfg* to contain the following line:

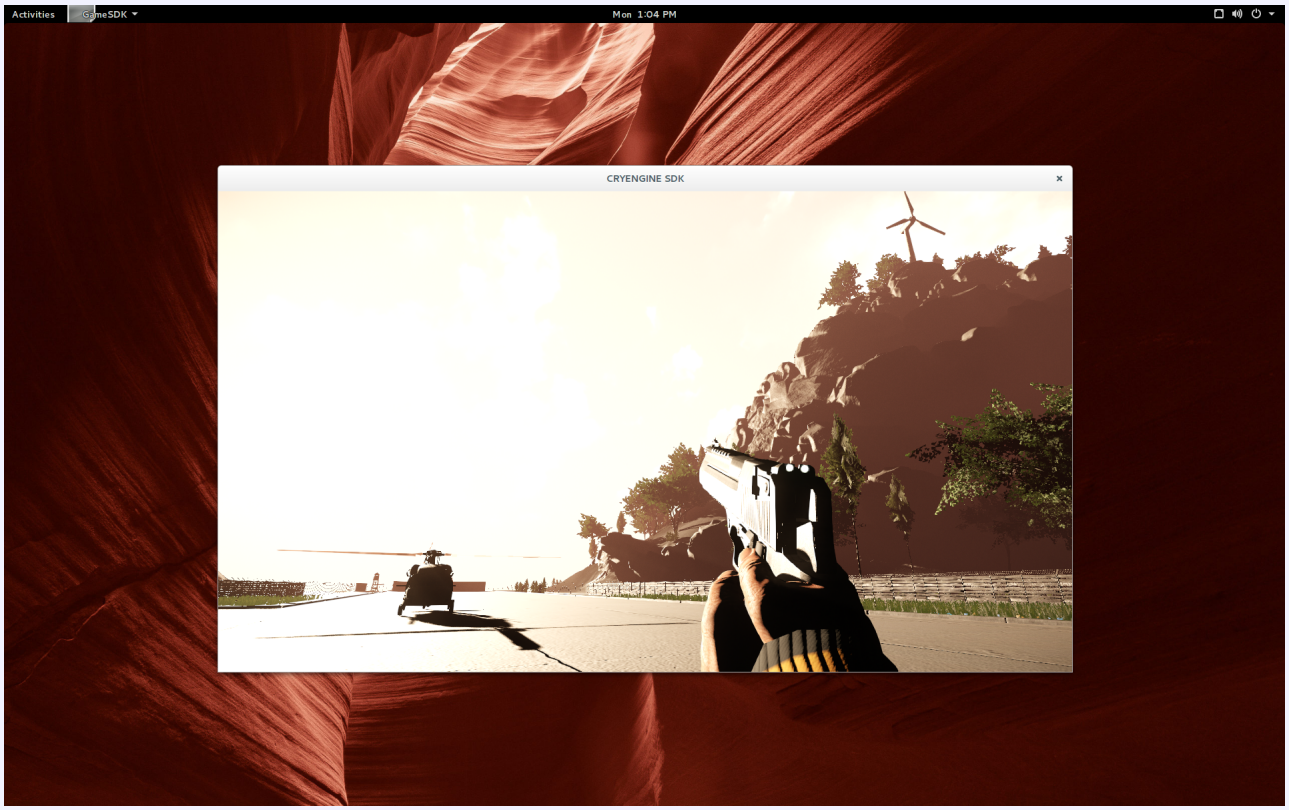
```
sys_game_folder="../gamesdk_5.2/GameSDK/Assets"
```

Then run the engine with:

```
$ ./bin/linux_x64_gcc/GameSDK
```

You should see something like the following:

Click on *Airfield*



As an alternative to modifying the source as mentioned above, you can unpack the *Engine/engine.pak* file, modify the *Engine/Config/engine_core.thread_config* file and repak it. This is the better solution since it limits the stack frame size change to the effected code. The above hack may eventually cause you to run out of memory because it increases the stack frame size for all threads not just RenderDLL. Since the function that requires this (Job_Plan) uses slightly more that 128KB you may get away with less than 256KB. GCC reports that Job_Plan uses 131248 static bytes.

In *Engine/Config/engine_core.thread_config*, find the section for Linux and change *StackSizeKB="128"* to *StackSizeKB="256"*:

```
<!– ============ –>
<Platform name="LINUX_common">
<!– ============ –>
<!– [RenderDLL] –>
<Thread name ="RenderThread" Priority="Above_normal" StackSizeKB="128"/>
```