Part 5 – Viewport and Camera

In order for anything to appear in your game, you will first need both a viewport and a camera.

A viewport is the rectangle that we can view the game world through. And a camera allows us to see all or part of that world at a time.

Take a look at your MyGame.ini and you'll see that a `[MainViewport]` section has already been specified for you.

```
[MainViewport]
Camera              = MainCamera
```

Let's give our game a blue sky as a background, by adding the `BackgroundColor` property to the config:

```
[MainViewport]
Camera            = MainCamera
BackgroundColor   = (0, 180, 255)
```

Secondly, notice the Camera parameter is set to `MainCamera`? This means it is set to the `[MainCamera]` section, which has also already been defined for you:

```
[MainCamera]
FrustumWidth    = 1280
FrustumHeight   = 720
FrustumFar      = 2.0
FrustumNear     = 0.0
Position        = (0.0, 0.0, -1.0) ; Objects with -1 <= Z <= 1 will be
visible
```

The `Position` is set to 0,0 which is in the center of the screen so that the whole screen can be viewed. The Z position is -1.0 which means the camera is set back a little from the action.

Let's change the screen width and height so that it makes working through the tutorial easier (and easier to take screenshots :)):

```
[MainCamera]
FrustumWidth    = 800
FrustumHeight   = 600
FrustumFar      = 2.0
FrustumNear     = 0.0
Position        = (0.0, 0.0, -1.0) ; Objects with -1 <= Z <= 1 will be
visible
```

A frustum width and height does not have to be the same size as your screen, but for our game, we want it to be. So the `ScreenWidth` and `ScreenHeight` values can be added to the `[Display]` section and are being inherited from the `FrustumWidth` and `FrustumHeight` values:

```
[Display]
ScreenWidth     = @MainCamera.FrustumWidth
ScreenHeight    = @MainCamera.FrustumHeight
...
```

The viewport needs to be created when our game runs, so let's take a look at the MyGame.c code (for a C++ project, look for `MyGame::Init` in MyGame.cpp):

```
orxSTATUS orxFASTCALL Init()
{

...
  orxConfig_PushSection("Main");

  for(orxS32 i = 0, iCount = orxConfig_GetListCount("ViewportList"); i <
iCount; i++)
  {
    orxViewport_CreateFromConfig(orxConfig_GetListString("ViewportList",
i));
  }
```

Here the "`ViewportList`" refers to this section of the config file:

```
[Main]
ViewportList    = MainViewport
```

You could also called `orxViewport_CreateFromConfig` on a static name:

```
orxSTATUS orxFASTCALL Init()
{

...

    orxViewport_CreateFromConfig("MainViewport");
```

The Init() function runs once when your game first executes, and this is where the viewport is created.

The "MainViewport" string parameter specified in that function is the same name as the `[MainViewport]` section in the config.

Because the camera was specified in the `[MainViewport]` section, we do not need to create a camera in code.

Final thing for this step is to name our game. This name will show in the window title. Set this with the `Title` property in the `[Display]` section. Also while you are here, set the `Smoothing` to false, and Decoration to true which means to make a standard window:

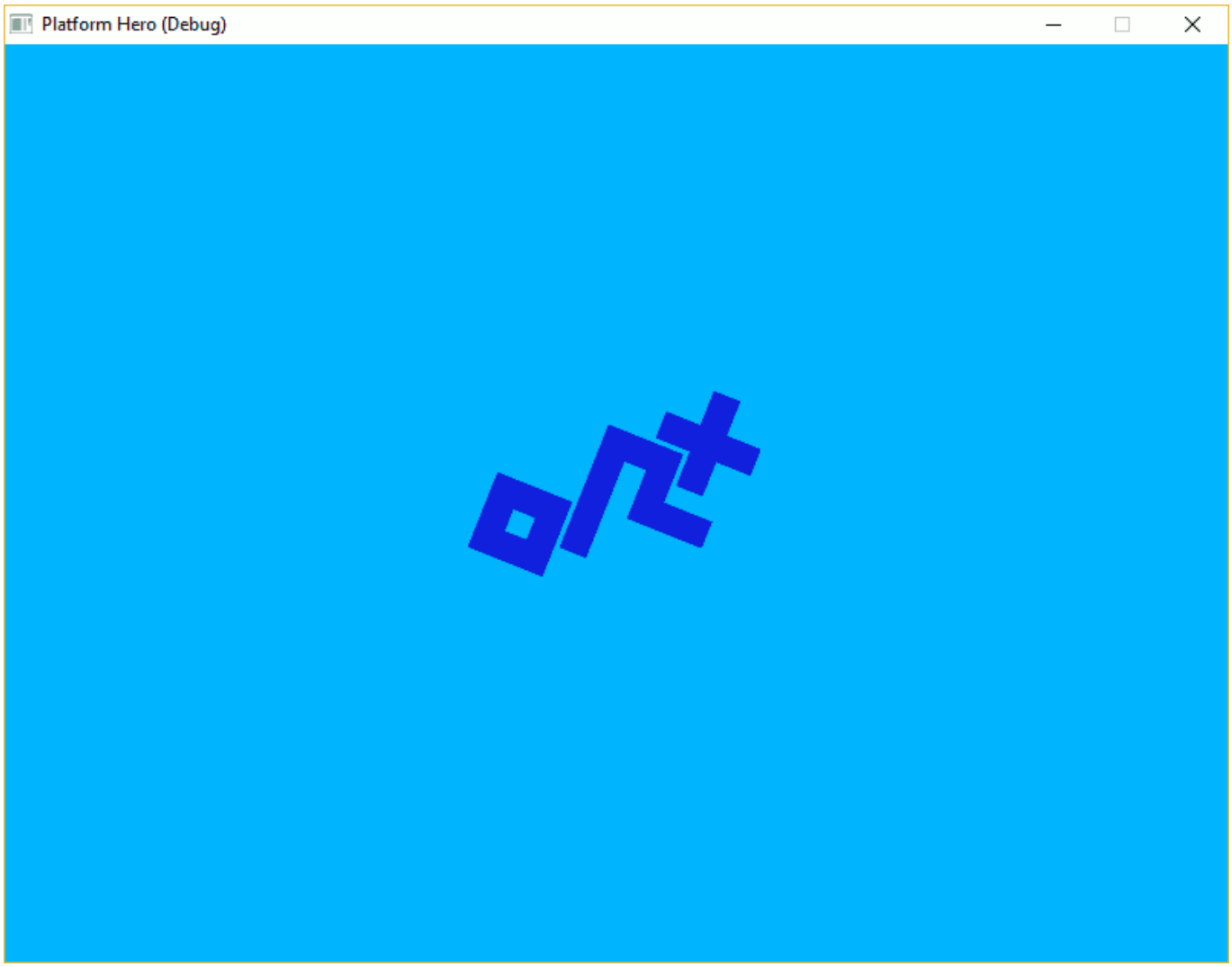
```
[Display]
```

```
ScreenWidth      = @MainCamera.FrustumWidth
ScreenHeight     = @MainCamera.FrustumHeight
Title            = Platform Hero
FullScreen       = false
Decoration       = true
Smoothing        = false
VSync            = true
```

This will change the name in the window title, but only if you are compiling in Release mode. You'll most likely be compiling in Debug mode right now. So you'll also need to change the title in the `MyGamed.ini` file:

```
[Display]
Title            = Platform Hero (Debug)
```

Compile and run to see your game window with a blue background and a default object in the center of the screen:



Next: Part 6 – Objects

- Part 1 – Downloading Orx
- Part 2 – How Orx works
- Part 3 – Setting up a new game project
- Part 4 – A tour of an Orx project
- Part 5 – Viewport and the camera
- Part 6 – Objects
- Part 7 – Spritesheets and Animation
- Part 8 – Platforms and Texture Repeating
- Part 9 – Physics
- Part 10 – Input Controls
- Part 11 – Running and Standing
- Part 12 – Changing Direction
- Part 13 – Getting our hero to shoot
- Part 14 – FX
- Part 15 – Collision Events.
- Part 16 – Jelly Monsters
- Part 17 – Timeline Tracks
- Part 18 – Exploding Monsters
- Part 19 – The Hero's survival.
- Part 20 – Text and Game Over

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/en/guides/beginners/viewport_and_camera?rev=1755708010

Last update: **2025/09/30 17:26 (14 months ago)**