Part 8 – Platforms and Texture Repeating

Our playfield will contain a number of platforms. What have we got in the current tutorial asset folders over in the Orx project that would do for a platform?



`box.png` from `orx/tutorial/data/object`. That'll do nicely. Copy this file into our `data/texture` folder.

We'll eventually repeat this object in a line to produce a platform.

Start off with a single platform object:

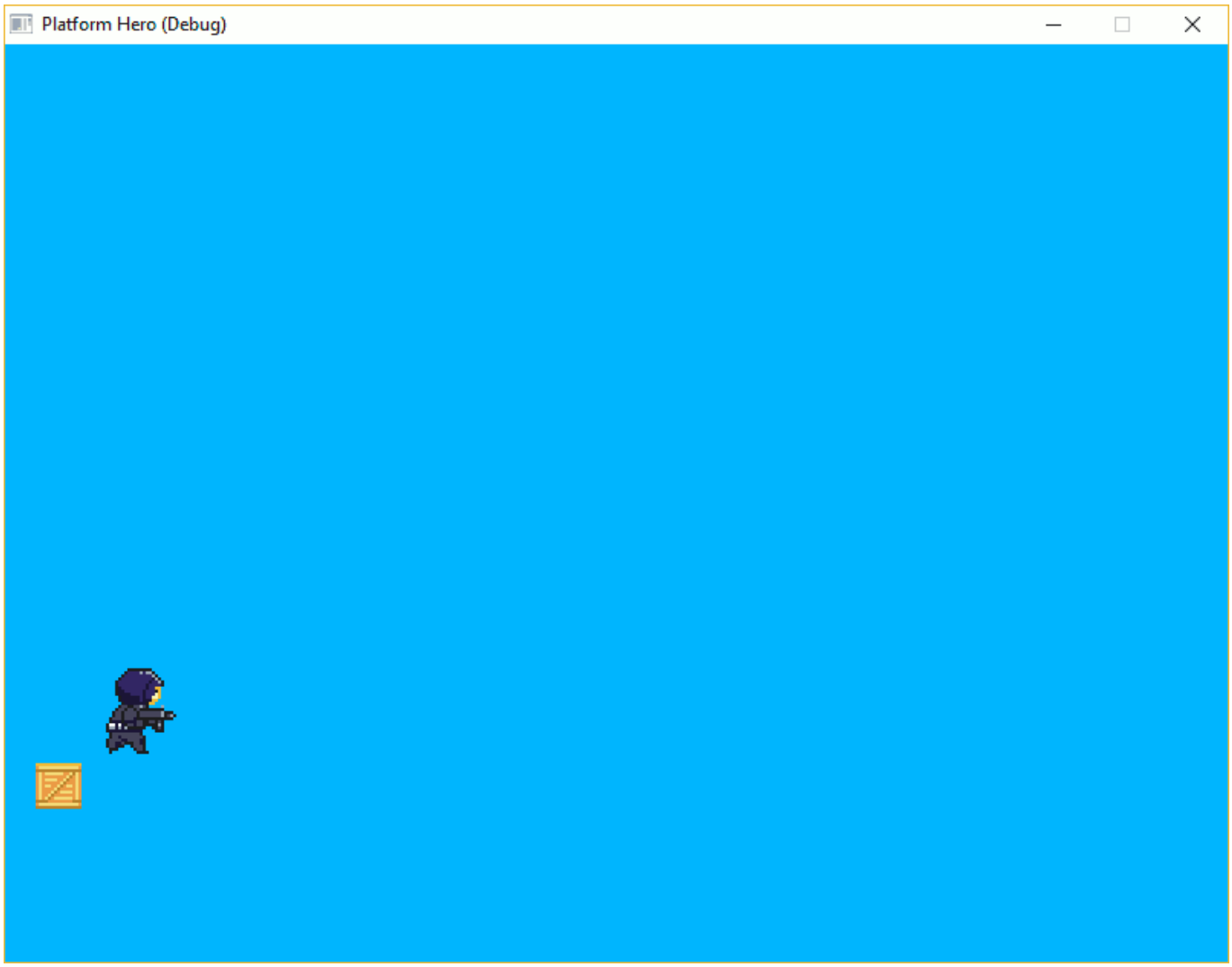
```
[PlatformGraphic]
Texture = box.png

[PlatformObject]
Graphic  = PlatformGraphic
Position = (-380, 170, 0)
Scale    = 2
```

And create it inside the Init() function with:

```
orxObject_CreateFromConfig("PlatformObject");
```

Compile and run. You'll get this:

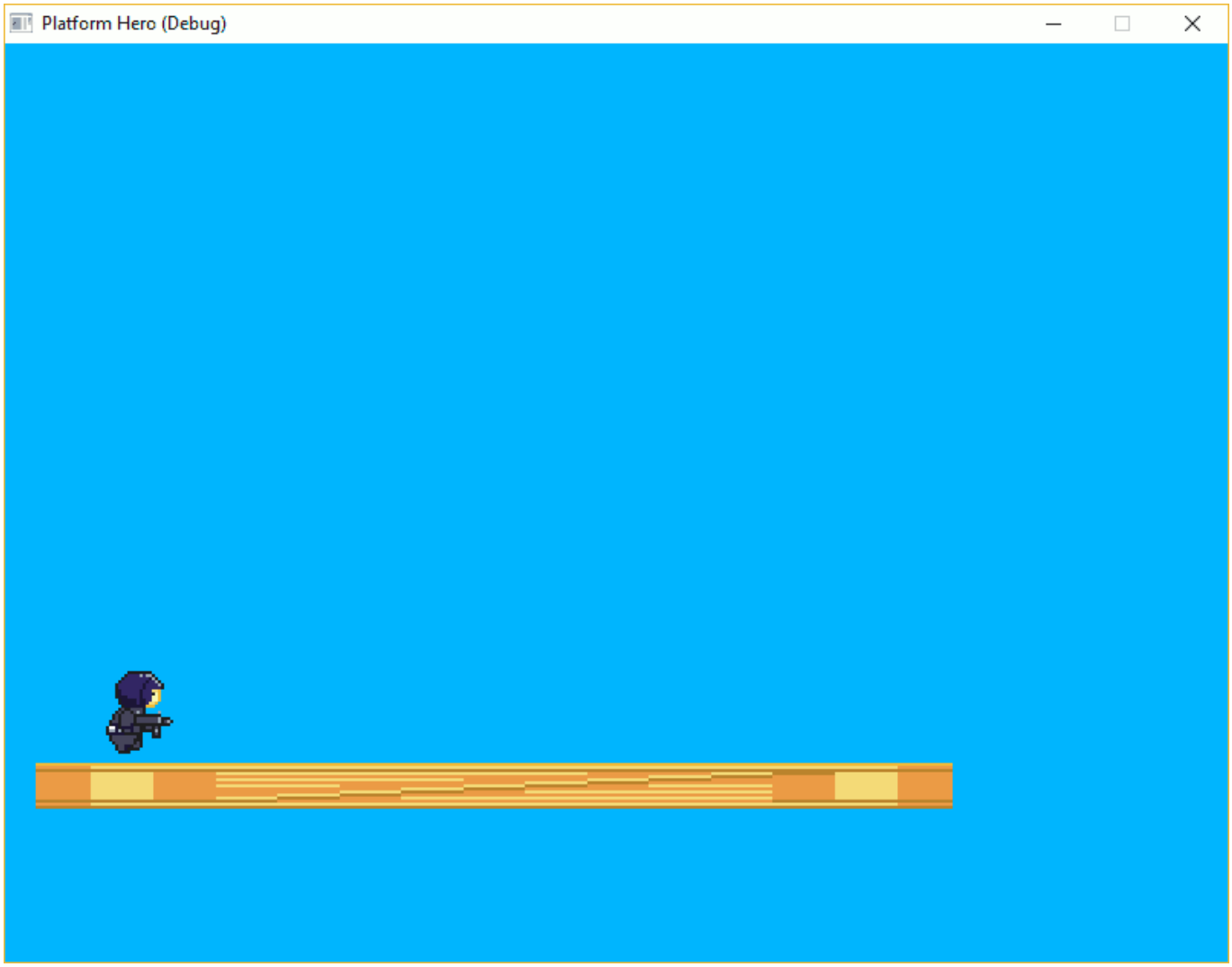


Great. Next, the platform needs to be stretched out so that it acts like a platform and not just a simple box. You can do this by changing the `Scale` property:

```
[PlatformObject]
Graphic  = PlatformGraphic
Position = (-380, 170, 0)
Scale    = (40, 2, 0)
```

So scale 40x along the X axis and 2x along the Y axis.

Run and you'll get:

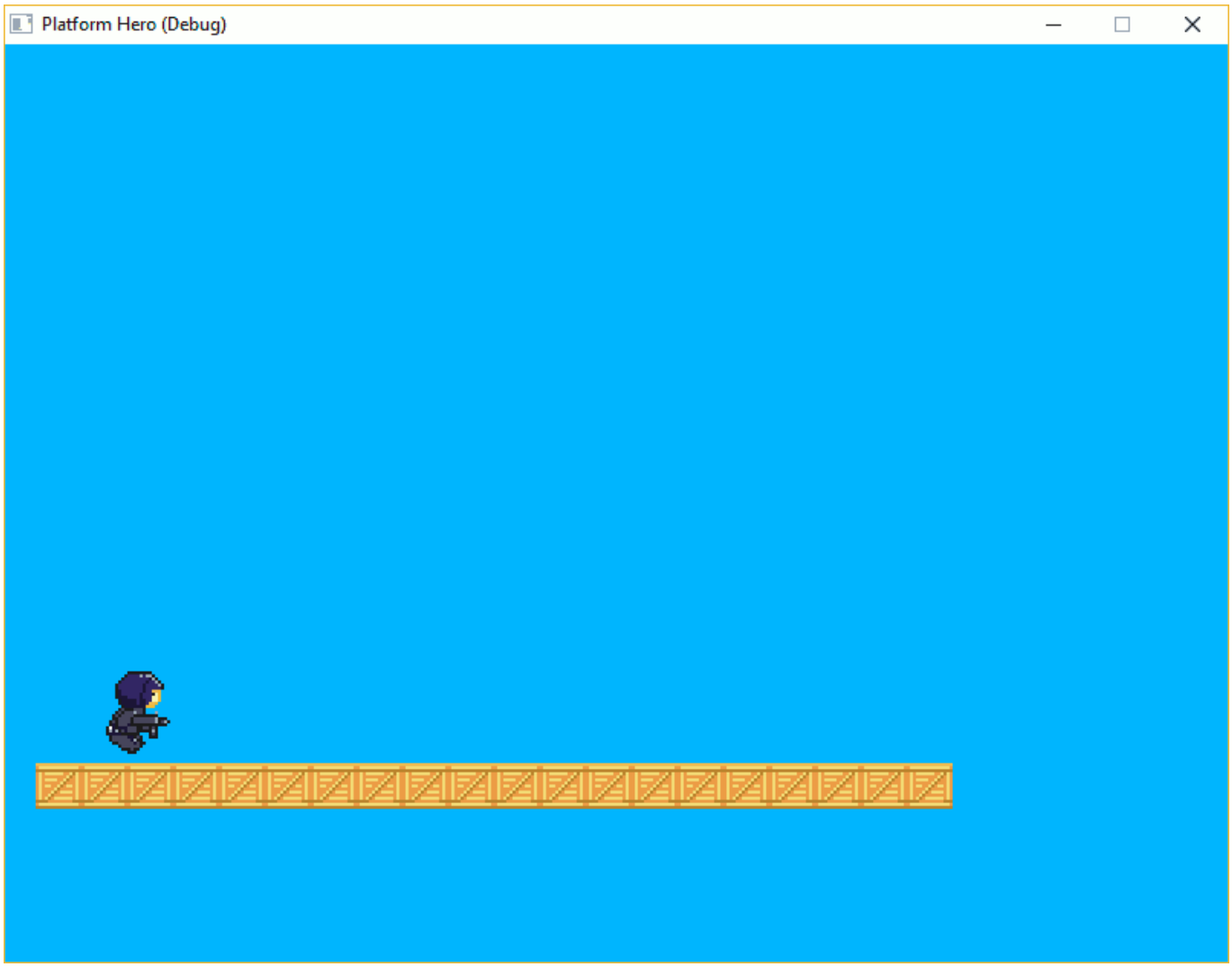


That's not quite what was expected. The object was scaled and the texture was scaled along with it.

We also need the Repeat property, so that the texture will repeat within the object:

```
[PlatformObject]
Graphic  = PlatformGraphic
Position = (-380, 170, 0)
Scale    = (40, 2, 0)
Repeat   = (20, 1, 0)
```

Run it and you'll see:



Great, that looks much nicer.

The Y value is 1, so that the texture is not repeated vertically, and X is 20 so that the texture repeats itself 20 times horizontally across the object.

Let's move the platform to the bottom of the screen and extend it to the whole screen width:

```
[PlatformObject]
Graphic  = PlatformGraphic
Position = (-400, 270, 0)
Scale    = (54, 2, 0)
Repeat   = (27, 1, 0)
```

We can use this as a template to create many platforms. But rather than create a few platforms in code, it would be much easier to create a single empty parent object with a few child objects under it.

Start by making the parent object called Scene to contain all the platform objects (and later other objects too):

```
[Scene]
ChildList = PlatformObject # MiddlePlatformObject # TopLeftPlatformObject
```

```
#TopPlatformObject #TopRightPlatformObject
```

Anything added to a `ChildList` will become a child under the object, and will move with the parent. Its co-ordinates will be local under the parent.

Notice that the `PlatformObject` is the only thing in the list that exists so far. Let's create others based on the settings of `PlatformObject`:

```
[MiddlePlatformObject@PlatformObject]
Position = (-150, 150, 0)
Scale    = (30, 2, 0)
Repeat   = (15, 1, 0)

[TopLeftPlatformObject@PlatformObject]
Position = (-400, 20, 0)
Scale    = (14, 2, 0)
Repeat   = (7, 1, 0)

[TopPlatformObject@TopLeftPlatformObject]
Position = (-100, -100, 0)

[TopRightPlatformObject@TopLeftPlatformObject]
Position = (200, -210, 0)
```

A quick explanation of the above syntax: `MiddlePlatformObject` and `TopLeftPlatformObject` inherit all four properties from `PlatformObject` and overrides three of them. `TopPlatformObject` and `TopRightPlatformObject` inherit properties from `TopLeftPlatformObject`, and the fourth property indirectly from `PlatformObject`.

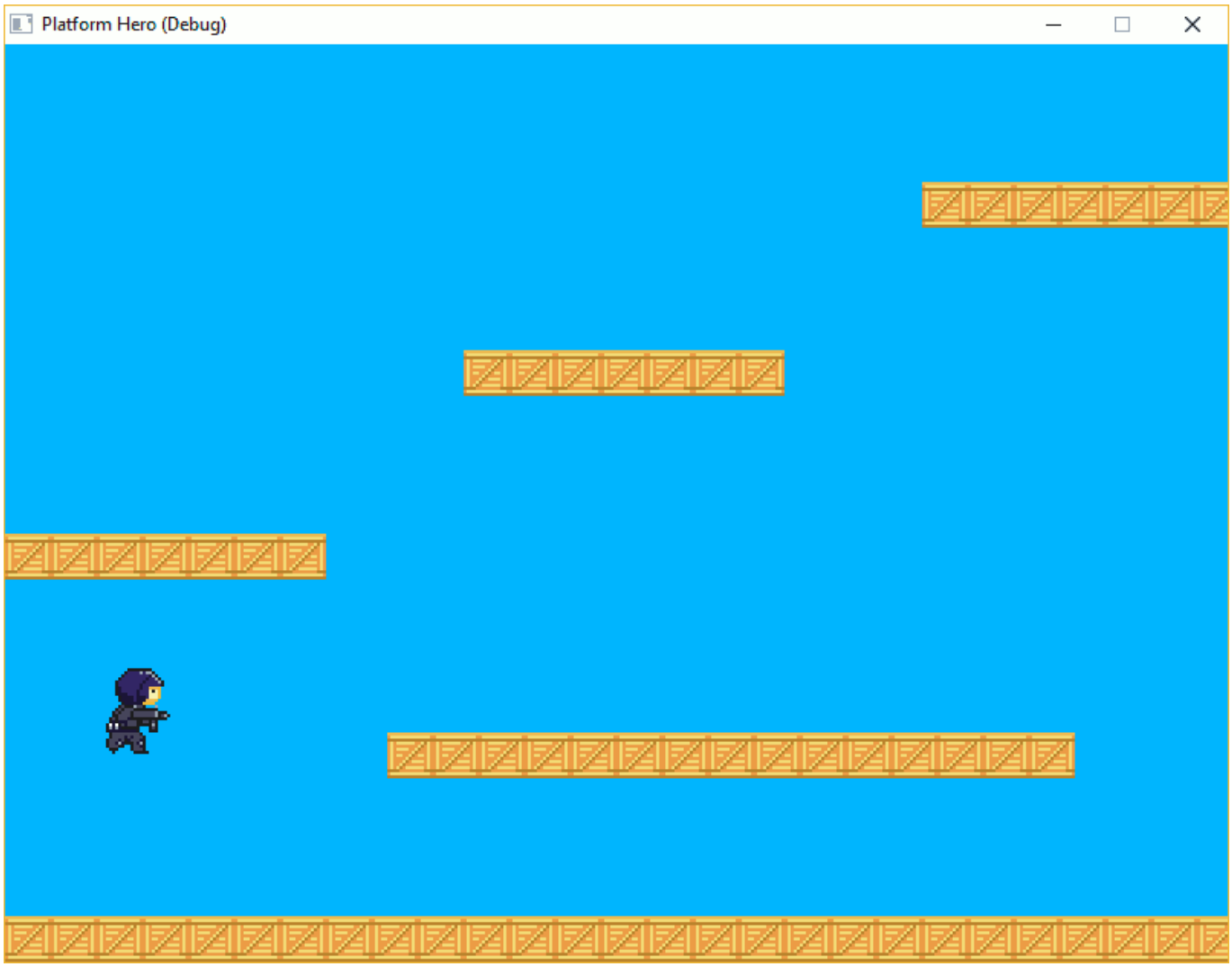
This is really to save on configuration, to be extensible, and to make things easier to change rather than copy and pasting the same details over and over.

Next change the code in the Init() function to create “Scene” instead of “PlatformObject”:

```
orxObject_CreateFromConfig("Scene");
```

Now every child under the Scene section will be created for us.

Compile and run. You should have a screen full of platforms:



Excellent!

Next, Part 9 - Physics.

- 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://orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://orx-project.org/wiki/en/guides/beginners/platforms_and_texture_repeating?rev=1550591618

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