ChunkMaster Tutorial

Return to Documentation

ChunkMaster

Saving & Loading

Index

1. Saving your world

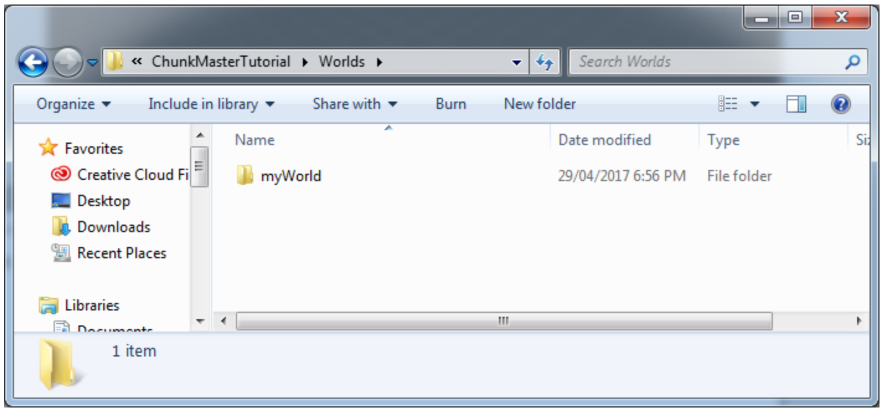
a) The SaveWorld method

Saving a world is just how you might expect it to be. We will need a method and a path to our save location. Consider the code below. Also lets remove the ClearBlock statement. We would need to see if our block has been saved in the future.

```
myWorld.SaveWorld("E:\\ChunkMasterTutorial\\Worlds");
```

Finally the result will be the following in your chosen directory.

See Figure - 1

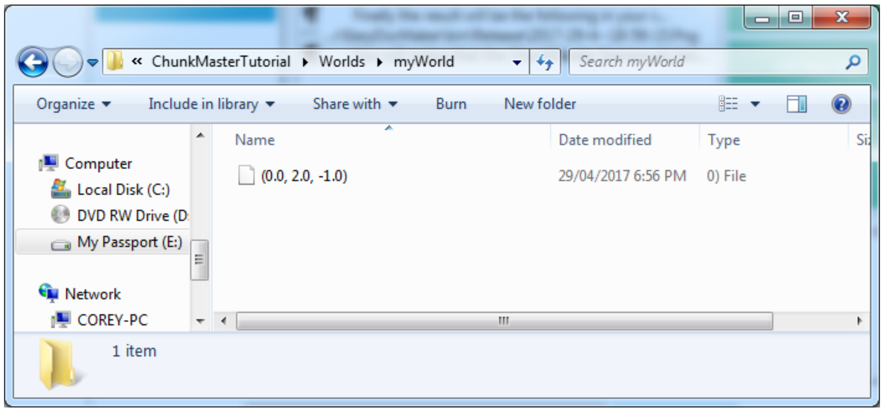


Name	Date modified	Type
myWorld	29/04/2017 6:56 PM	File folder

Figure - 1

You will notice that the ChunkMaster framework has created a directory "myWorld", this is normal as it takes the current world being manipulated and saves all the sectors to the designated directory.

See Figure - 2



Name	Date modified	Type
(0.0, 2.0, -1.0)	29/04/2017 6:56 PM	0) File

Figure - 2

In the figure above you can see that inside the "myWorld" folder we will have a file with some strange notation. ChunkMaster has simply taken the edited coordinates and saved the changes to a file. Even if we don't have any blocks in our world, it will still save the file.

b) Understanding the save file notation

You may be confused to see a file being generated with the name "(0.0, 2.0, -1.0)", how could this be since we've edited inside the coordinates (15.4f, 64.4f, -12f). What the ChunkMaster has done is take the "(15.4f, 64.4f, -12f)" coordinates, round the values off to the nearest 32 block division, divide by the number of chunks per sector, divided by the number of chunks per sector division, and finally the result will be the file name directory.

Example: (15.4f, 64.4f, -12f) will be rounded off to the nearest 32 division resulting in (0, 64, -32), and divided by (chunkSize(8) * sectorSize(4)) which will in tern be equal to (0.0, 2.0, -1.0).

Just to keep us on track, here is what your code should look like.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

using ChunkMaster.Unity.Framework;

public class WorldController : MonoBehaviour {
    public static UnityWorld myWorld;

    // Use this for initialization
    void Start ()
    {
        // Creating a myWorld instance object
        myWorld = new UnityWorld(transform);

        // Declaring a random point in space
        Vector3 worldPoint = new Vector3(15.4f, 64.4f, -12f);

        // Set a block in world space
        myWorld.SetBlock(worldPoint, 0);

        // Checking if the space is occupied
        print(myWorld.IsOccupied(worldPoint));

        // Saving the world instance to your desired file loca
        myWorld.SaveWorld("E:\\ChunkMasterTutorial\\Worlds");
    }
}
```

I Completed This Section

2. Loading your world

a) The LoadSector method

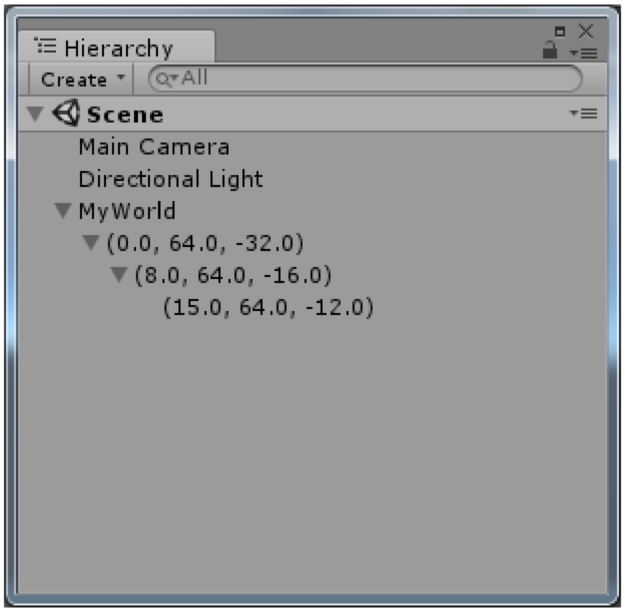
Let's imagine that you have a world with millions of blocks at unique locations. You've saved your world and closed your application, you are now ready to reload your data. It wouldn't be very smart to load a million blocks at one time, this is why we've created the LoadSector statement. You can load a portion of blocks by using a desired point in world space, which will then load the blocks for that coordinate segment. Consider the code below.

```
            // Declaring a random point in space
            Vector3 worldPoint = new Vector3(15.4f, 64.4f, -12f);

            // Loading a sector
            myWorld.LoadSector(worldPoint, "E:\\ChunkMasterTutori
```

The code above looks kind of odd. We are specifying a point in world space with the directory of our world location. If the blocks exist in storage for that location, then the blocks will be loaded in, otherwise there will be a DirectoryNotFoundException. The output will be the following.

See Figure - 3



```
Scene
  Main Camera
  Directional Light
  MyWorld
    (0.0, 64.0, -32.0)
      (8.0, 64.0, -16.0)
        (15.0, 64.0, -12.0)
```

Figure - 3
Success, we've loaded our blocks for that location!

I Completed This Section

Up Next Rendering Blocks->

Developed by Corey St-Jacques
Questions please contact Corey_stjacques@hotmail.com