v1.1.1

2D Terrain Editor Documentation

Description

This package contains a simple tool for Unity that allows you to create, edit or randomly generate 2D terrain. Create terrain for your 2D game with one click or generate it in runtime. Feels great on mobile and takes only 1-2 drawcalls. Perfect solution for 2D racing games with Physics2D components.

Features:

- Terrain brush for quick editing
- Random terrain generator
- API for creating 2D terrains at runtime
- Mobile-friendly

Quick start

1. Import package and switch scene view camera to 2D or orthographic mode. Go to GameObject > Create other > Terrain2D and select created **New Terrain2D** object on the scene.

2. Open Terrain Settings in Terrain Editor 2D component. Configure size and resolution of your terrain and click "Apply".

3. Assign Main material and set Texture size**.**

4. Take a look at the scene view. When **New Terrain2D** is selected you can enter to *EDIT MODE* using [Shift+E] combination. You will see terrain brush under mouse position (green circle). Raise your terrain using [Left Mouse Button] or lower [D + Left Mouse Button].

5. Now you can make a prefab of **New Terrain2D** object and use it lately.

Runtime generation

Examples:

1. Creating simple terrain at runtime

```
void Start()
    {
        CreateSimpleTerrain();
    }

void CreateSimpleTerrain()
    {
    //STEP 1. Creating default terrain
    GameObject newTerrain = TerrainEditor2D.InstantiateTerrain2D(new Vector3(0, 0, 0));

    //STEP 2. Assign material to generated terrain
    newTerrain.GetComponent<TerrainEditor2D>().MainMaterial = Terrain2DMaterial;

    }
```

2. Creating randomly generated terrain at runtime

```
void Start()
    {
        CreateRandomTerrain();
    }

void CreateRandomTerrain()
    {
        //STEP 1. Creating default terrain
        GameObject newTerrain = TerrainEditor2D.InstantiateTerrain2D(new Vector3(0, 0, 0));

        TerrainEditor2D myTerrain = newTerrain.GetComponent<TerrainEditor2D>();

        //STEP 2. Assign material to generated terrain
        myTerrain.MainMaterial = Terrain2DMaterial;

        //STEP 3. Randomize terrain
        myTerrain.RandomizeTerrain();

        //STEP 4. Recalculate terrain collider
        myTerrain.UpdateCollider();
    }
```

3. Creating randomly generated terrain with cap, fixed sides and 3D collider

```
void Start()
    {
        CreateRandomTerrain(new Vector2(10, 0));
    }

void CreateRandomTerrain(Vector2 pos)
    {
        //STEP 1. Creating default terrain with width=100, height=50 and resolution=4
        GameObject newTerrain = TerrainEditor2D.InstantiateTerrain2D(pos, 100, 50, 4);

        TerrainEditor2D myTerrain = newTerrain.GetComponent<TerrainEditor2D>();

        //STEP 2. Cofigure generated terrain
        myTerrain.MainMaterial = Terrain2DMaterial;
        myTerrain.CapMaterial = TerrainCapMaterial;

        //Configure cap
        myTerrain.CapHeight = 0.75f;
        myTerrain.CapOffset = 0.1f;
        myTerrain.CreateCap = true;

        //Confire fixed sides
        myTerrain.LeftFixedPoint = 25;
        myTerrain.RightFixedPoint = 25;
        myTerrain.FixSides = true;

        //Configure 3D collider
        myTerrain.Collider3DWidth = 5;
        myTerrain.Create3DCollider = true;

        //STEP 3. Configure and randomize terrain
        myTerrain.RndAmplitude = Random.Range(2, 8);
        myTerrain.RndHeight = Random.Range(5, 15);
        myTerrain.RndHillsCount = Random.Range(4, 8);
        myTerrain.RandomizeTerrain();

        //STEP 4. Recalculate terrain collider
        myTerrain.UpdateCollider();
    }
```

Check out **RuntimeGenerationExample** scene and code comments in **Terrain2DGenerator.cs** script that placed in /2DTerrainEditor/Example/Scripts/ to learn how runtime generation really works.

NOTE: One time created terrain and saved as prefab will not be generate mesh again. All created 2D terrain meshes in editor mode will be placed in /2DTerrainEditor/SavedMeshes/ folder.