

Terrain Engine 2D

A 2D Block Engine for Unity

Out now on the Unity Asset Store

BUY NOW!

FEATURES

DOCUMENTATION

API

FAQ

DEMO

EXAMPLE PROJECT

Terrain Engine 2D

User Manual - V1.20

INTRO ▾

GENERAL ▾

MAIN PROPERTIES ▾

Contents

Here is a list and explanation of all the main assets of Terrain Engine 2D. *Note that this does not include any extra files and example projects included in the asset.*

Table of Contents

- Graphics
- Objects
- Prefabs
- Scripts

Graphics Contains all the base textures and materials for the engine

Materials Contains all the base materials for the engine

- Lighting** Contains all the materials for the lighting
 - AmbientLight.mat** The material used for the ambient light mesh
 - BasicLight.mat** The material used for any basic light
 - BlockLighting.mat** The material used by the blocklighting mesh
 - FastBlur.mat** A material used by the Advanced Lighting system for blurring the light textures
 - LargeLight.mat** The material used for large light sources
 - LightBlend.mat** A material used by the Advanced Lighting system for blending light textures
 - Overlay.mat** A material used by the Advanced Lighting system for overlaying textures
 - SmallLight.mat** The material used for small light sources
- FluidMesh.mat** The material used for the mesh generated by the Fluid Chunks
- FluidTextured.mat** The material used for the FluidRenderer to render the fluid texture
- GridSelector.mat** The material used for the Grid Selector
- LightIcon.mat** The material used by the Grid Selector to show the lighting icon
- Sprite.mat** The material used for the Sprites that use Z-ordered layering

Shaders Contains all the shaders for the engine

- Lighting** Contains all the shaders for the lighting
 - AmbientLighting.shader** Shader used for rendering the ambient lighting
 - FastBlur.shader** Shader used for blurring light textures
 - LightBlend.shader** Shader used for blending light textures
 - LightSource.shader** Shader used on all light sources
- Fluid.shader** Shader used for rendering the fluid texture
- Sprite.shader** Replacement for the Default Sprite shader with ZWrite enabled
- Terrain.shader** Shader used for terrain textures
- UI-Default.shader** Built in Unity shader used by the Overlay material

Sprites Contains all the sprites for the engine

- Lighting** Contains all the Sprites for the lighting
 - Large_Light_Radial.png** Sprite used for large radial lights
 - Small_Light_Radial.png** Sprite used for small radial lights
- Circle_Fill.png** UI toggle button fill
- Circle_Outline.png** UI toggle button image outline
- Circle_Outline_2.png** UI color wheel outline image
- Color_Wheel.png** UI radial color wheel image for the fluid color picker
- Grid_Selector.png** The image of the Grid Selector
- Overlap_Block_Template.png** Template used for creating Overlap Blocks

Objects All the generic asset objects used by the engine

- FluidDensityInputValidator.asset** Object used by the OSD Fluid Density Input to ensure proper formatting of input values

Prefabs All the Prefabs used in the engine

- Lighting** Prefabs of light sources
 - AdvancedRaycastLight.prefab** An advanced Raycast light which shoots raycasts to the edges of the terrain to generate shadows
 - FlashLight.prefab** An advanced dynamic raycast light which rotates to face the cursor
 - FloodLight.prefab** A light source which floods the general area with light
 - RaycastLight.prefab** A light source which shoots Raycasts in a circular manor around the light to generate shadows
- Torch.prefab** An example block light with a simple texture and particle effects
- Single Instance** Prefabs of GameObjects which should only contain one instance per scene
 - GridSelector.prefab** The Grid Selector tool for modifying the generated world
 - OSD.prefab** The On Screen display for modifying the generated world
 - World.prefab** The World which controls all components of the engine
 - WorldCamera.prefab** The main Camera which displays the terrain, lighting and UI on the screen
- Chunk.prefab** The prefab used to generate the chunks which render the world
- LayerOption.prefab** A potential layer option for the OSD

Scripts All the source scripts used in the engine

Editor Custom Editor scripts

- ProjectEditor.cs** A collection of functions used to help update old projects to the latest version of TE2D
- ProjectStartup.cs** Any tasks that must run when the project starts to ensure TE2D works properly
- WorldCustomInspector.cs** This script controls the custom inspector for the World

Extras Custom Editor scripts

- CursorFollower.cs** This class causes its GameObject to follow the cursor
- FaceCursor.cs** This class causes a 2D GameObject to rotate to face the cursor
- MonoBehaviourSingleton.cs** This abstract class is used as a base for all scripts that should act as Singletons
- TexturedMesh.cs** Generates a custom texture and renders it to a mesh

Fluid Dynamics The scripts used to simulate the Fluid Dynamics System

- Advanced Fluid Dynamics** The scripts used by the Advanced Fluid Dynamics system
 - AdvancedFluidBlock.cs** This class stores the information for blocks of fluid of the advanced system
 - AdvancedFluidDynamics.cs** This class simulates the advanced fluid physics
 - FluidType.cs** The type of fluid, used by the Advanced Fluid Dynamics system
- FluidBlock.cs** This class stores the information of a single block of fluid
- FluidChunk.cs** This class generates the fluid mesh for a single chunk
- FluidDynamics.cs** This class simulates the fluid physics
- FluidRenderer.cs** This class renders the fluid simulation in a texture

Lighting The scripts used for the Lighting

- Advanced Lighting** The scripts used for the Advanced Lighting system
 - Block Lighting** This class controls the ambient lighting
 - AmbientLight.cs** This class controls the ambient lighting
 - BlockLighting.cs** This class controls the block lighting system
 - BlockLightMesh.cs** Generates a textured mesh used to render the block lighting
 - BlockLightNode.cs** A node of light used by the Block Lighting system
 - BlockLightSource.cs** A source of light for the Block Lighting system
 - Mesh Lights** The scripts used for the mesh light sources
 - AdvancedRaycastLight.cs** The script that controls the AdvancedRaycastLight light source
 - FloodLight.cs** The script that controls the FloodLight light source
 - MeshLight.cs** The script for a source of light that generates a mesh
 - RaycastLight.cs** The script that controls the RaycastLight light source
 - AdvancedLightSystem.cs** This class controls the advanced 2d lighting system
 - LightRenderer.cs** Renders the advanced lighting
 - LightSource.cs** The base light source script
- LightSystem.cs** This class controls the basic world lighting

Physics The scripts used for object physics in the engine

- PhysicsObject.cs** A custom physics script for objects used with the engine

Serialization The scripts used for file I/O and serialization of data

- AdvancedFluidData.cs** Serializable script for saving advanced fluid data
- BaseData.cs** Base serialization script for any save data
- BlockData.cs** Serializable script for saving block data
- FluidData.cs** Serializable script for saving fluid block data
- SaveData.cs** Script holding any data to be saved to a file
- Serialization.cs** Static class for saving and loading data
- SerializationHelper.cs** Static class of helpful functions for serialization
- WorldData.cs** ScriptableObject holding all preferential data (world inspector data) for the world

Terrain The scripts responsible for generating, modifying and controlling the terrain

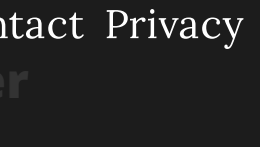
- BlockGridMesh.cs** The class is used to create a 2D mesh made up of blocks
- BlockInfo.cs** This class stores information of a single block type
- BlockLayer.cs** This class holds block layer data and information
- Chunk.cs** The class controls a single chunk
- ChunkLoader.cs** This class controls loading and unloading of chunks
- ColliderGenerator.cs** The class generates the colliders for a chunk
- FallingBlockSimulation.cs** The class controls the Falling Block Simulation
- TerrainGenerator.cs** This class is meant to be expanded upon, it contains the framework for generating the terrain
- TerrainGeneratorTemplate.cs** This is the template for creating a TerrainGenerator script
- World.cs** This is the main World class which holds all block layers and other important information for controlling the terrain

Tools The tool scripts

- CameraController.cs** This class handles input and controls the camera
- ChildCameraController.cs** This class maintains the orthographic size of a child camera with the parent
- GridSelectorImageSetter.cs** This class changes the image of the Grid Selector
- OSDController.cs** This class controls the OSD
- WorldInputHandler.cs** This class handles user input
- WorldModifier.cs** This class contains functions for modifying the terrain

UI The scripts used by the user interface

- ColorPicker.cs** The script used to control the color picker of the OSD
- FluidDensityInputValidator.cs** The script used to create Fluid Density Input Validator objects
- LinkBox.cs** A component attached to the World GameObject showing relevant links to help users of the engine



Copyright © 2020 Matthew Wilson. All Rights Reserved.

Contact Privacy Top

Help support the developer

DONATE