

# 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 - V1.10

GENERAL

BASIC

ADVANCED

## Lighting

This page explains how the lighting works in Terrain Engine 2D.

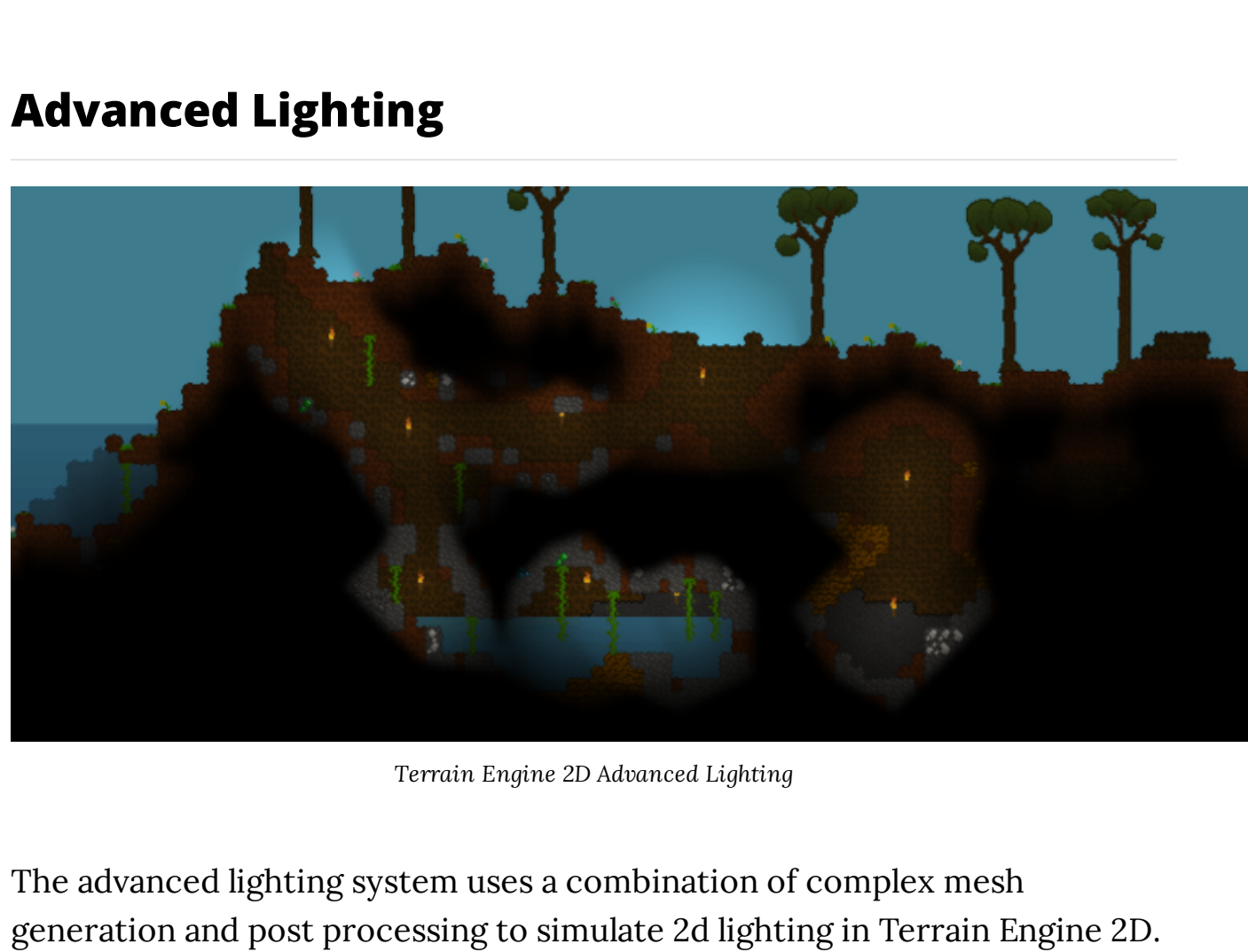
### Table of Contents

- General
- Basic Lighting
- Advanced Lighting

### General

Terrain Engine 2D contains a full featured 2d lighting system with many options to suit the needs of your unique game. The lighting in Terrain Engine 2D consists of two different lighting systems; basic and advanced. The basic lighting system is meant for games that do not require a full lighting solution. It is more performant than the advanced lighting system, but lacks many of the advanced features. The advanced lighting system has ambient lighting, complex light sources, a day/night cycle, and options for post processing.

### Basic Lighting



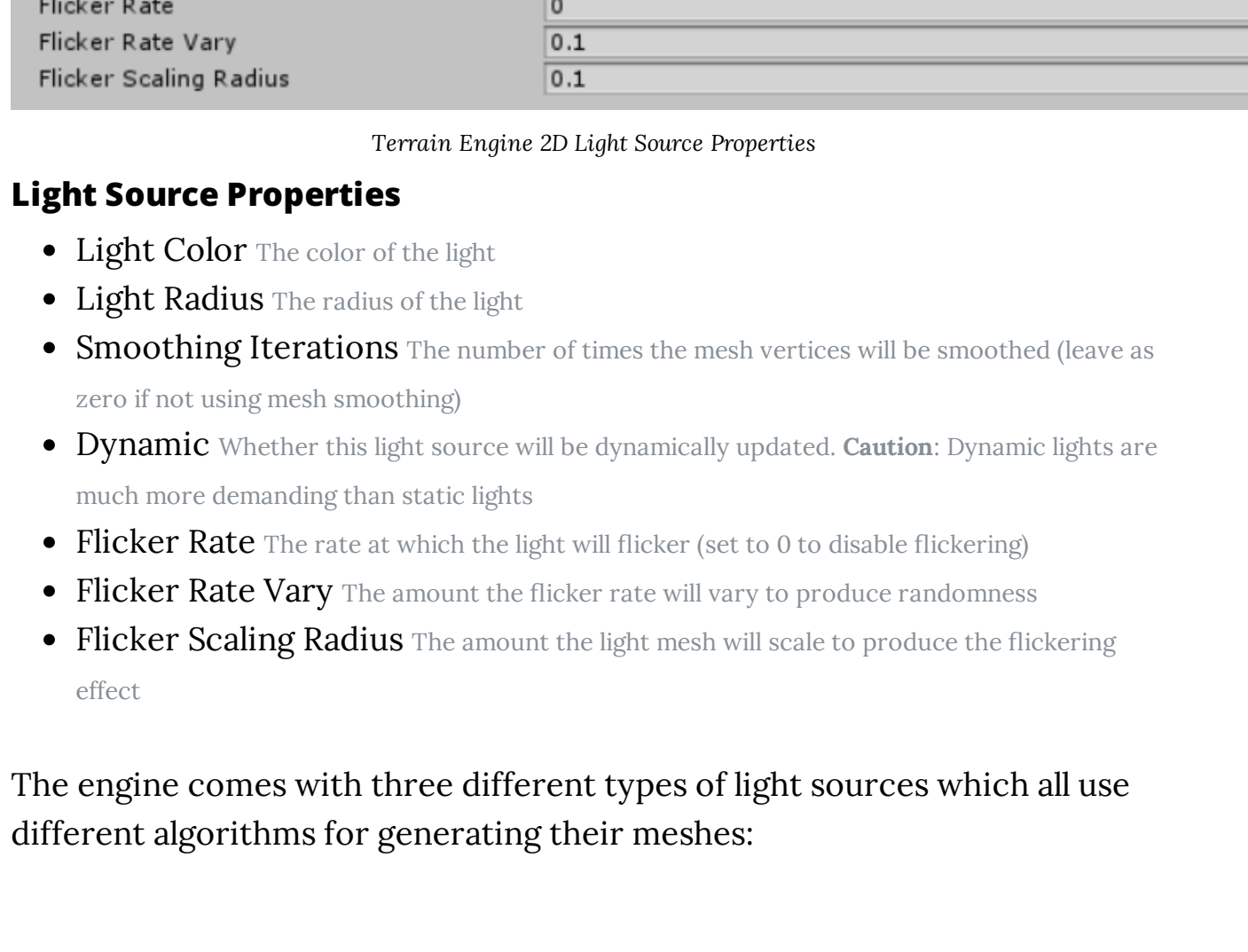
Terrain Engine 2D Basic Lighting

The basic lighting system in Terrain Engine 2D is highly optimized and does not use any post processing making it a good option for those who do not need a complex lighting system and would like to maximize the performance of their game.

The lighting system renders a texture the size of the current loaded world/chunks. The pixels of the texture are set based on the terrain blocks of the selected Light Layer. If there is a terrain block in the corresponding texture position then the color of that pixel is set based on the Edge Light Bleed value and that block's distance from the edge. The closer the block is from the edge the more visible it will be. Blocks that are a distance from the edge that is greater than the Edge Light Bleed value have a corresponding pixel color that is black. If there is no terrain block in the correspond texture position that pixel is set to white. In this fashion a shadow mask is generated which will hide the inner portions of the terrain.

All light data is stored in memory for fast access which means that using the basic lighting system will cause your game to use more memory than if you opt to not use any lighting.

### Advanced Lighting



Terrain Engine 2D Advanced Lighting

The advanced lighting system uses a combination of complex mesh generation and post processing to simulate 2d lighting in Terrain Engine 2D. Each source of light in the system (including the ambient lighting) generates a mesh to represent the illuminated area around the light. A separate camera is used to render all of these light sources into a texture using a variety of custom shaders, this texture is then blurred and then blended with the texture of the main camera. After that a separate overlay camera which captures all of the UI and other graphics that should render above the lighting has its texture copied onto the main texture. The main texture is then output to the screen with the added lighting and overlay.

### Layers

There are two special layers that are required to be added to your project in order to use the advanced lighting system. These layers are used by the additional cameras to render the lighting and UI.

### Lighting

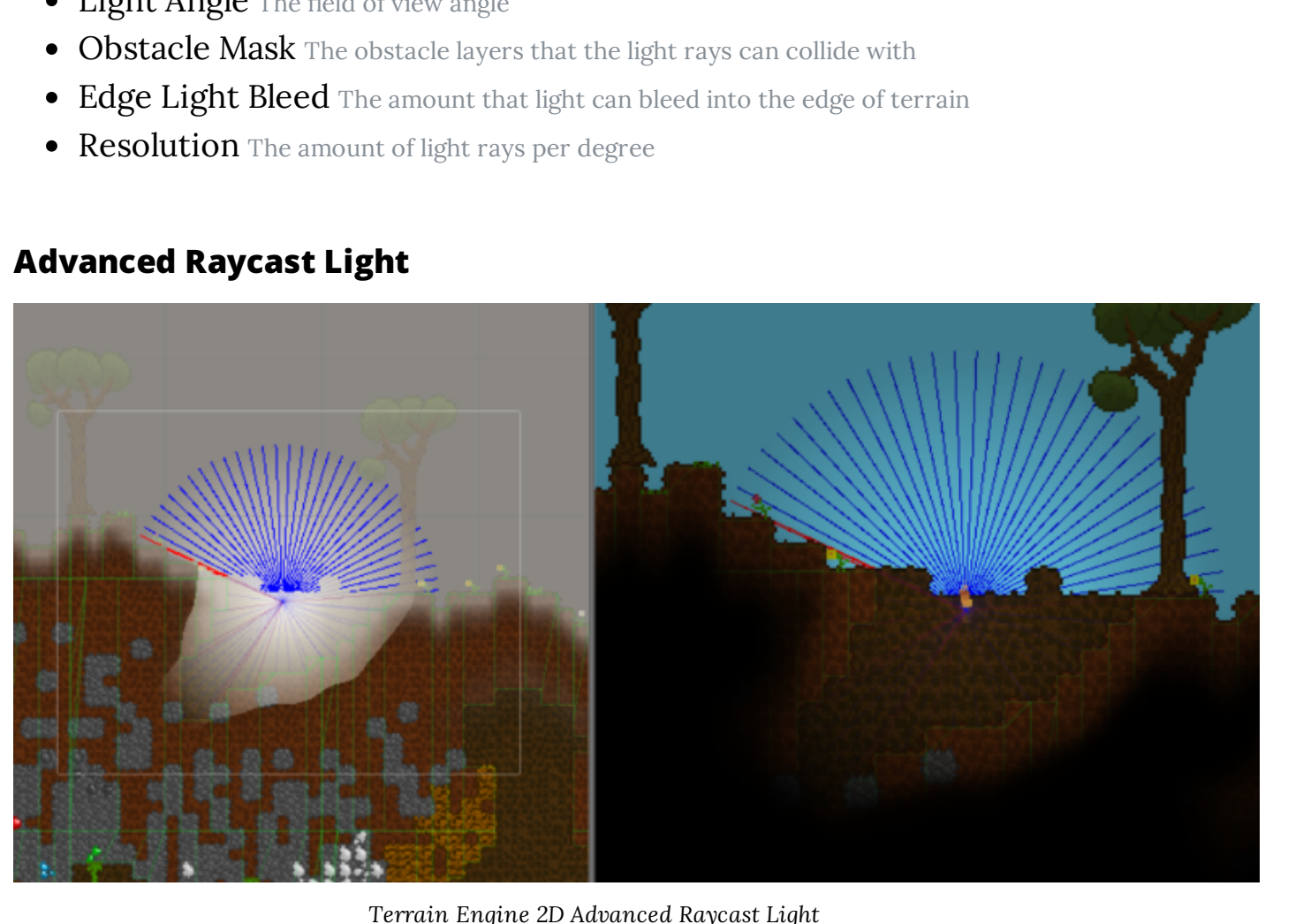
The Lighting layer is to be used by all light sources that generate a mesh. This is used so the Light Camera knows which objects to render. No other objects should use this layer.

### Ignore Lighting

The Ignore Lighting layer is to be used by all objects which should render on top of the lighting graphics but are not considered UI objects and shouldn't be using the UI layer.

### Light Sources

In the advanced lighting system a light source in its simplest form is a generated mesh. The generation of the mesh is controlled by the LightSource script which has a number of properties for manipulating how the light will look. Each Light Source also must have a material which will contain the texture and shader used to render the light. Both the texture and shader can be modified in order to produce different light shapes and effects.



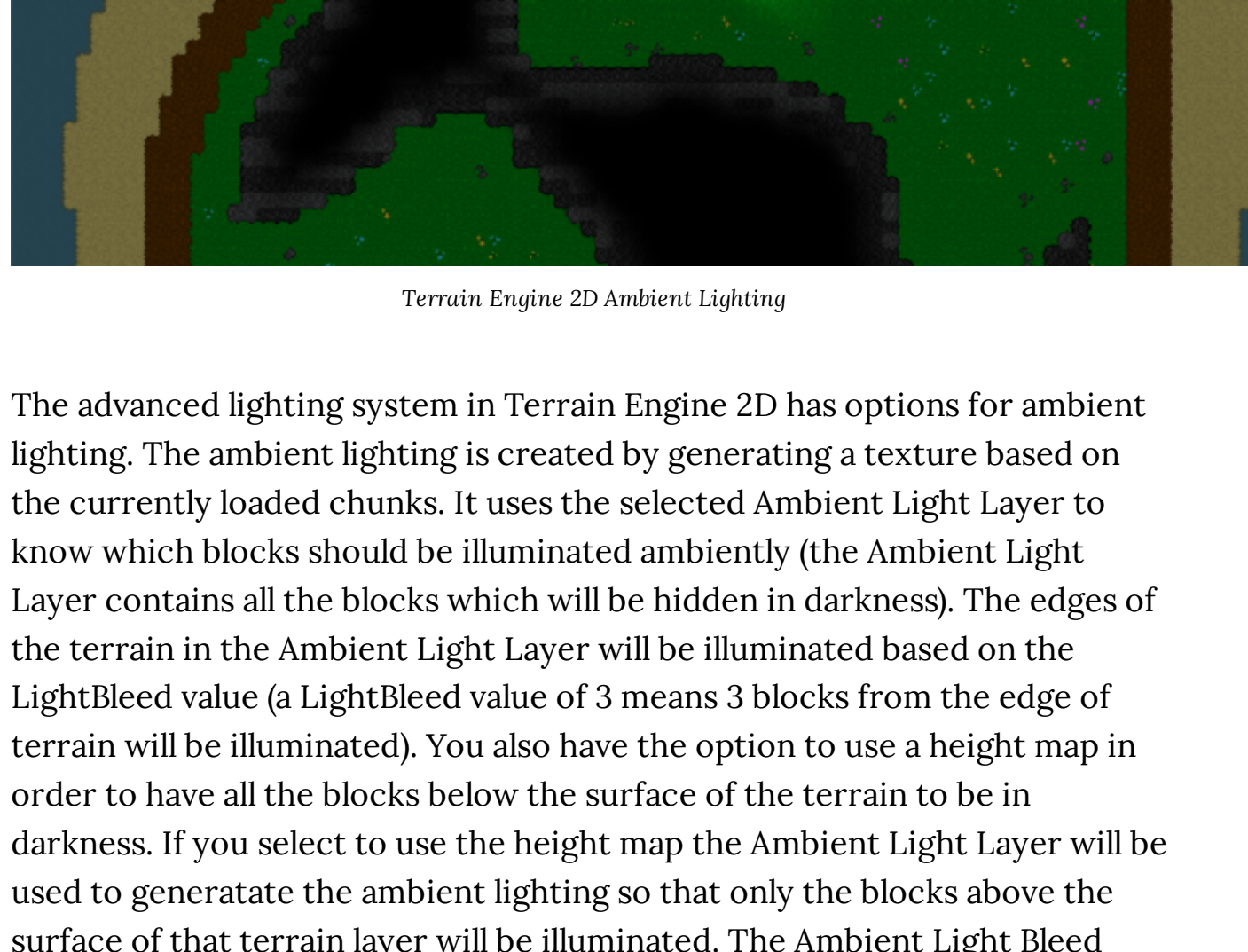
Terrain Engine 2D Light Source Properties

### Light Source Properties

- Light Color** The color of the light
- Light Radius** The radius of the light
- Smoothing Iterations** The number of times the mesh vertices will be smoothed (leave as zero if not using mesh smoothing)
- Dynamic** Whether this light source will be dynamically updated. **Caution:** Dynamic lights are much more demanding than static lights
- Flicker Rate** The rate at which the light will flicker (set to 0 to disable flickering)
- Flicker Rate Vary** The amount the flicker rate will vary to produce randomness
- Flicker Scaling Radius** The amount the light mesh will scale to produce the flickering effect

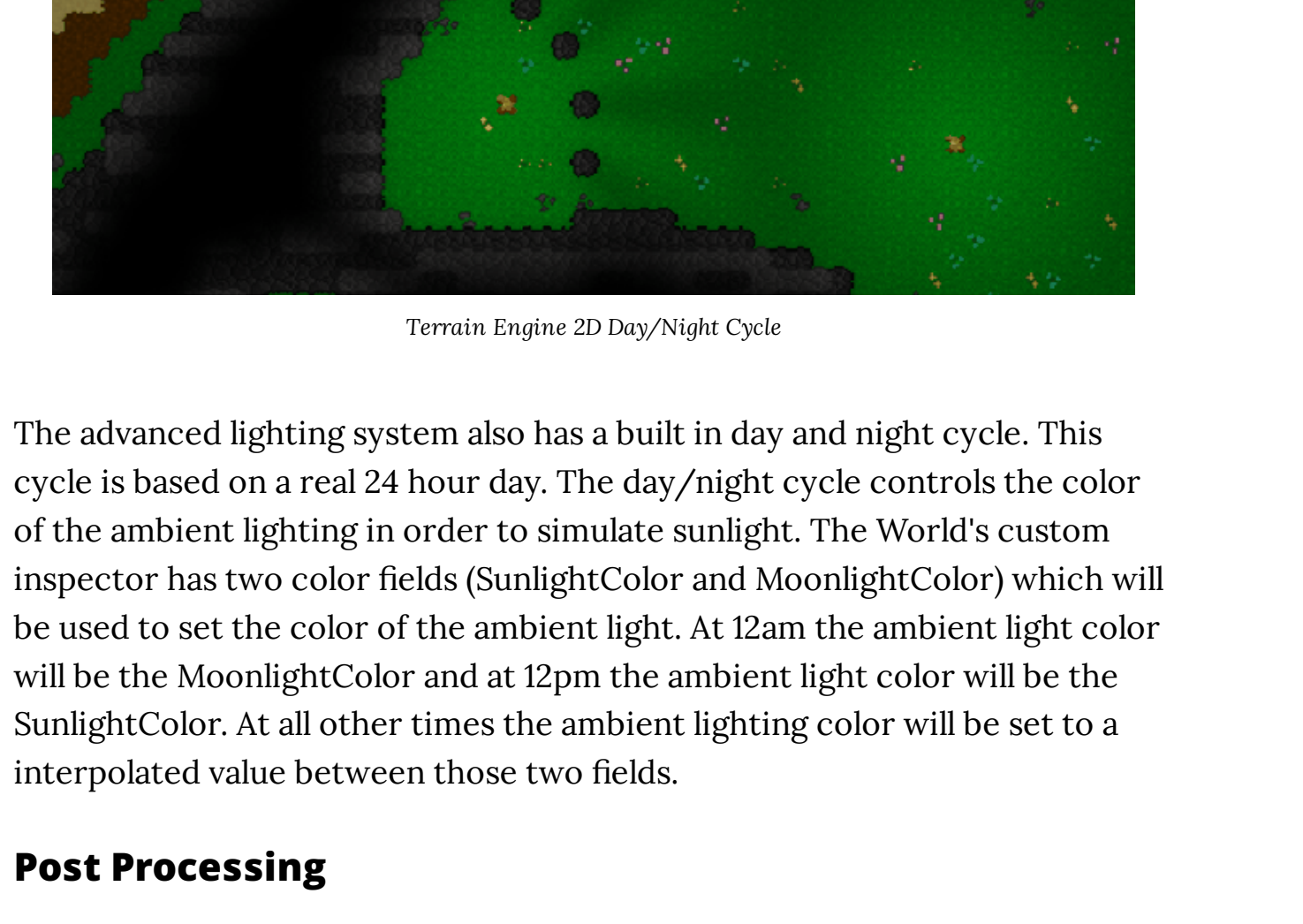
The engine comes with three different types of light sources which all use different algorithms for generating their meshes:

### Flood Light



Terrain Engine 2D FloodLight

The FloodLight is a light source which generates its mesh vertices by looping through the blocks in the Light Layer in 8 directions (north, north-east, east, south-east, south, south-west, west, north-west) straight out from the center of the light. Light can propagate through a certain number of blocks as determined by the BlockLightTransfer value. This results in 8 mesh vertices that make up the shape of the mesh of this light source. This type of light source is good to use for lights that should illuminate large areas as well as terrain blocks of the Light Layer.

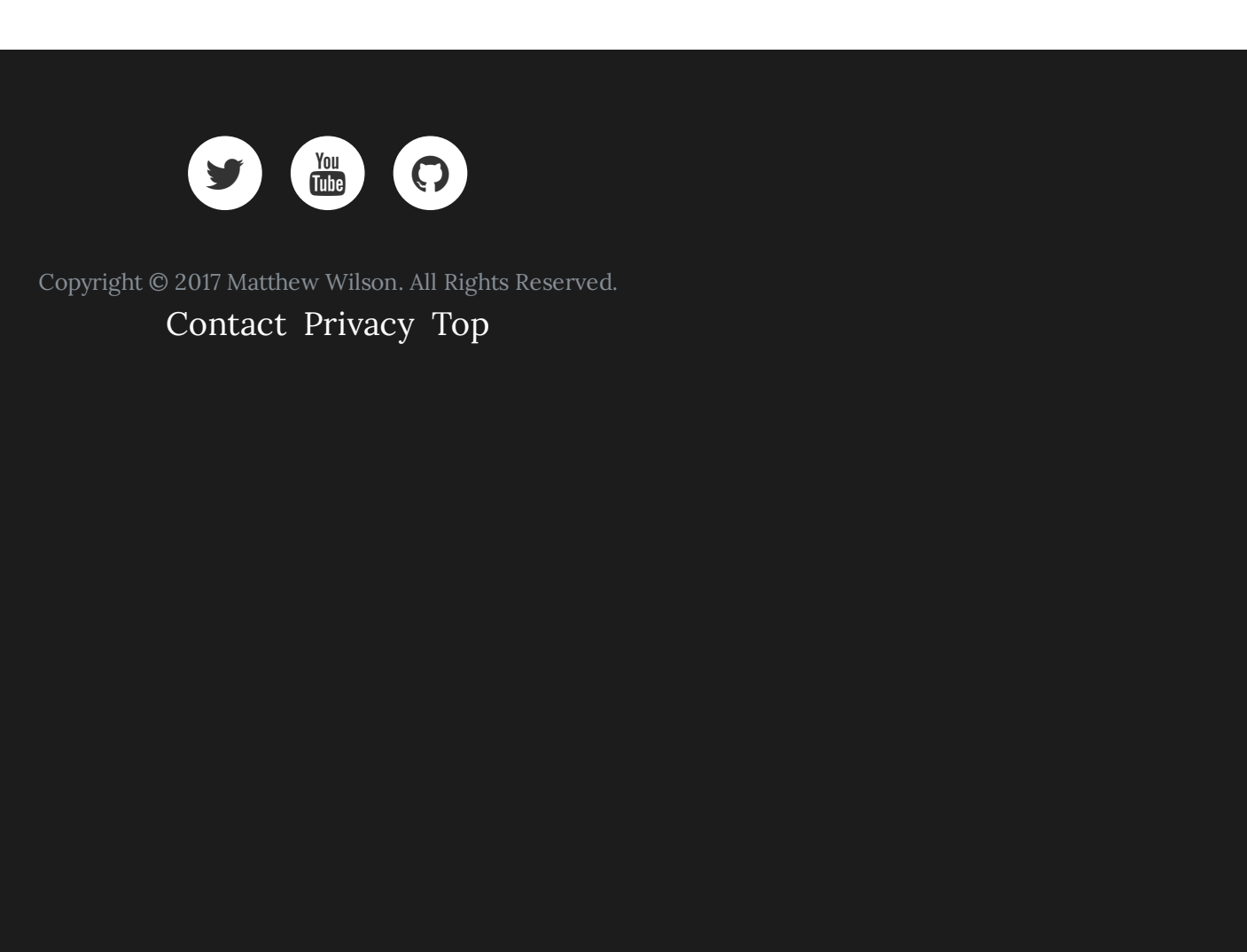


Terrain Engine 2D FloodLight Properties

### FloodLight Properties

- Block Light Transfer** The amount of blocks that light can pass through

### Raycast Light



Terrain Engine 2D Raycast Light

The Raycast Light is a light source which generates its mesh vertices by raycasting out from the center of the light. Rays are cast in a clockwise manner based on the Resolution and Light Angle. If there is a collision with an obstacle the collision point is used as a mesh vertice, else the vertice is set to the furthest point the ray will reach as determined by the Light Radius. This type of light source is good for static lights that should cast shadows on certain objects.



Terrain Engine 2D Advanced Raycast Light Properties

### Raycast Light Properties

- Light Angle** The field of view angle
- Obstacle Mask** The obstacle layers that the light rays can collide with
- Edge Light Bleed** The amount that light can bleed into the edge of terrain
- Resolution** The amount of light rays per degree

### Advanced Raycast Light



Terrain Engine 2D Advanced Raycast Light

The Advanced Raycast Light is a light source similar to the Raycast Light but is quite a bit more complex. Instead of just Raycasting out from the center in no particular direction, this light finds all the nearby terrain chunks and Raycasts to the points in the PolygonCollider2Ds (the corners of the terrain). This produces more accurate shadows and is great for dynamic moving lights that should cast shadows.



Terrain Engine 2D Advanced Raycast Light Properties

### Advanced Raycast Light Properties

- Raycast Offset** The amount the light rays are offset from terrain points.
- Raycast Hit Threshold** The threshold of whether the correct point was hit. Light points below this threshold are ignored.

### Ambient Lighting



Terrain Engine 2D Ambient Lighting

The advanced lighting system in Terrain Engine 2D has options for ambient lighting. The ambient lighting is created by generating a texture based on the currently loaded chunks. It uses the selected Ambient Light Layer to know which blocks should be illuminated ambiently (the Ambient Light Layer contains all the blocks which will be hidden in darkness). The edges of the terrain in the Ambient Light Layer will be illuminated based on the LightBleed value (a LightBleed value of 3 means 3 blocks from the edge of terrain will be illuminated). You also have the option to use a height map in order to have all the blocks below the surface of the terrain to be in darkness. If you select to use the height map the Ambient Light Layer will be used to generate the ambient lighting so that only the blocks above the surface of that terrain layer will be illuminated. The Ambient Light Bleed value will then be used to set the lighting of terrain blocks below the surface. This is used to mimic sunlight shining down from the sky into the terrain (because of this it does not work with top-down style games).

The ambient lighting is updated dynamically, which means if blocks are added or destroyed the ambient lighting texture will be regenerated.

### Day and Night Cycle



Terrain Engine 2D Day/Night Cycle

The advanced lighting system also has a built in day and night cycle. This cycle is based on a real 24 hour day. The day/night cycle controls the color of the ambient lighting in order to simulate sunlight. The World's custom inspector has two color fields (SunlightColor and MoonlightColor) which will be used to set the color of the ambient light. At 12am the ambient light color will be the MoonlightColor and at 12pm the ambient light color will be the SunlightColor. At all other times the ambient lighting color will be set to an interpolated value between those two fields.

### Post Processing

The advanced lighting system renders the lighting separately and then blends it into the main graphics after. In this way, post processing effects can be applied to the light texture in order to create more smooth and beautiful looking lighting. During the post processing the lighting texture is downsized and then blurred a number of times in order to smooth the hard edges of the generated meshes. You can control how many times the texture is downsized and blurred in the World's custom inspector. Downsizing is the fastest and easiest way to blur the texture, but it does not produce the best result and can give undesirable effects. Blurring the texture looks nice but can slow down performance as this is done every single frame. Try and find a good combination of downsizing and blurring the lighting for your game.



Copyright © 2017 Matthew Wilson. All Rights Reserved.

[Contact](#) [Privacy](#) [Top](#)