# LightShafts 2

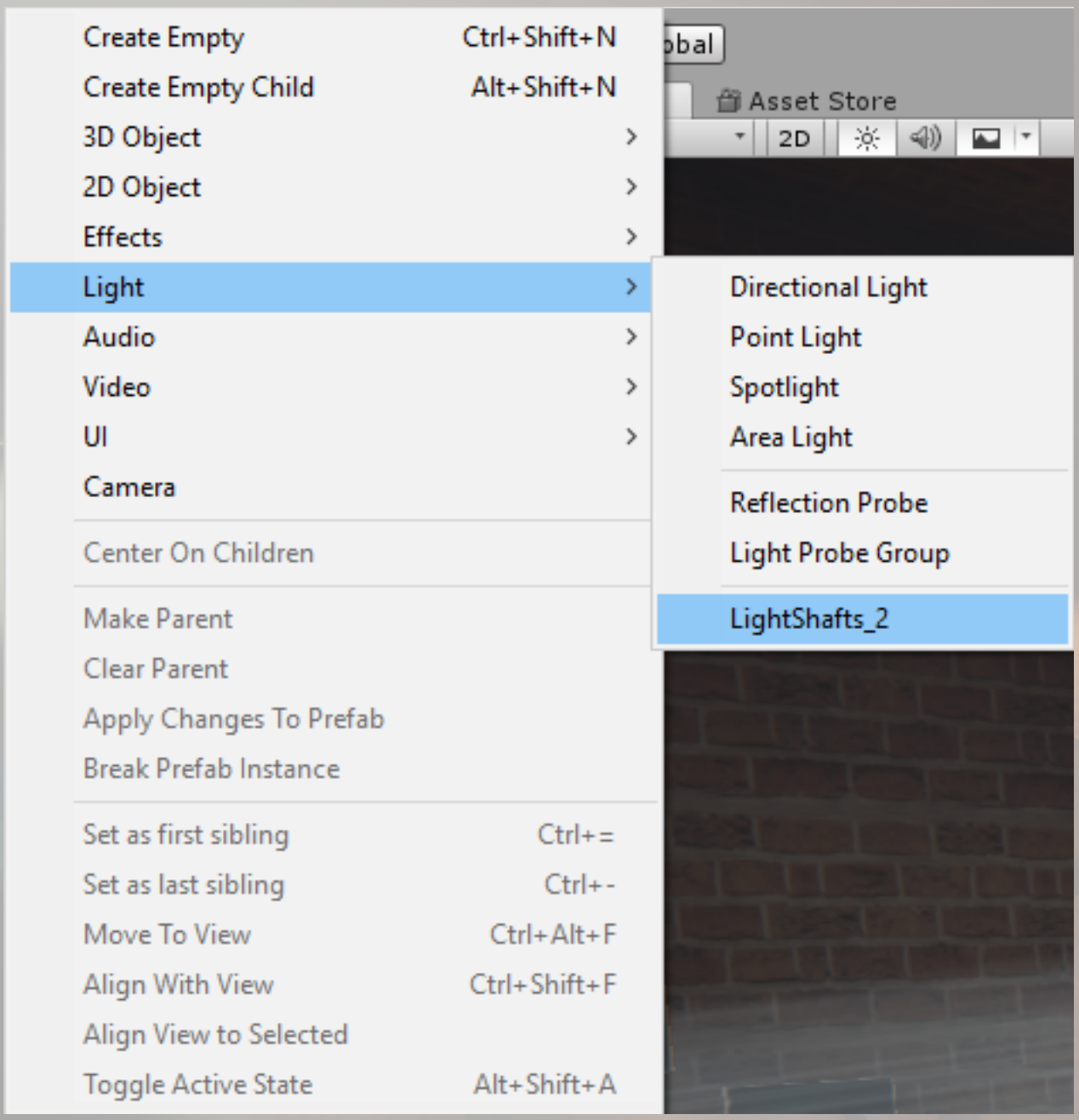
MARK DUISTERS

Getting started

System explanation

FAQ

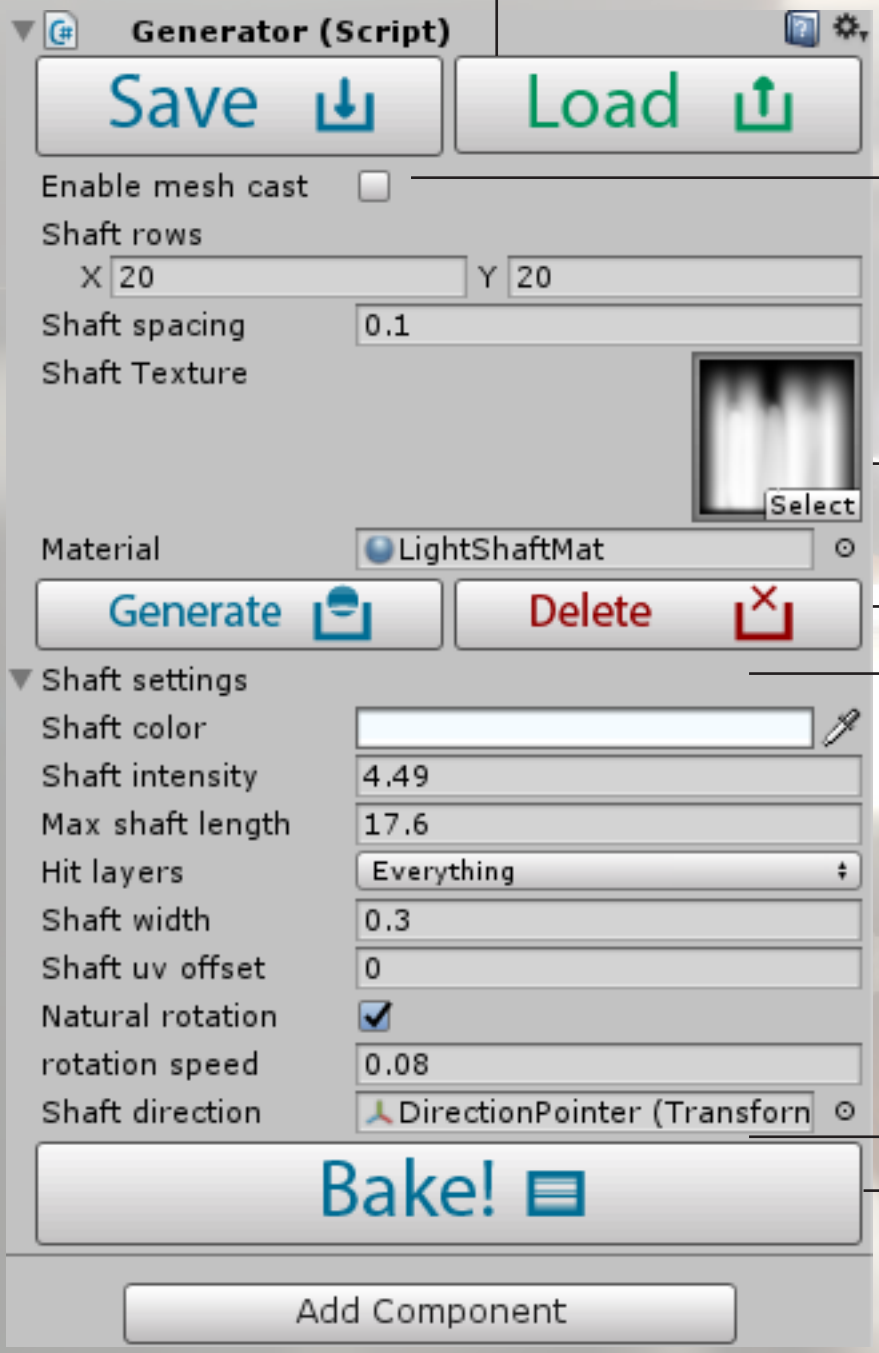
# Getting started




To place a LightShaft generator in the scene, go to: GameObject -> Light -> LightShafts_2. This will place a GameObject in the scene named “LightShaftGenerator”.

# Inspector overview

You can save or load settings to/ from a preset file.
Everything is saved except for "Shaft Texture", "Material" and "Shaft direction". Because they are Unity specific.




Generator inspector component.

"Enable mesh cast" switches between square or mesh cast mode. Square cast uses rows to spawn shafts (shafts = X * Y) and mesh cast uses each vertex as a spawn location. The actual mesh is never used.
"Spacing" sets the distance between shafts and the scale of a mesh cast.

Set a texture for each shaft to use. "Material" is a sharedmaterial. Generators with the same material will share settings.

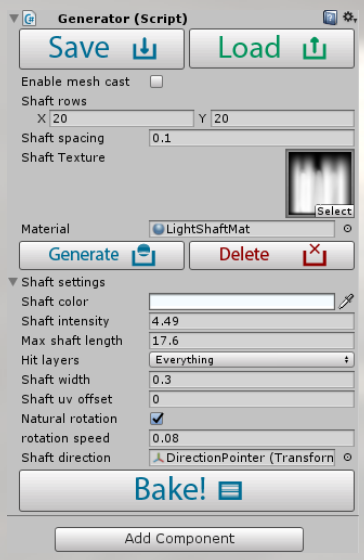
Generate will create shafts with above settings. Delete removes the current set. (Generate uses Delete before creating new shafts)

Adjusts the look of the generated shafts. These settings are updated real-time during edit mode.

For optimal performance (or for mobile use) generated shafts can be baked into a single static mesh. This also means your system is no longer dynamic. To tweak settings, simply click generate again and re-bake your shafts.

# System explanation

## Editor script




## Shaft behavior script

```
//all public variables will use Generator pointers as

public Color shaftColor= new Color(255,197,96,255);//s
public float shaftIntensity=22.0f;//sets the shader's

//public    Mesh shaftMesh;//users can set custom mesh
public  float maxLength= 50.0f;//used in the raycast,
public  LayerMask layerMask=1;//filter against what th

public float  shaftWidth = 1.0f;//set the shat's width
public  float shaftAdjustY = 1.0f;//set the Ytiling of
public Texture shaftTexture;//this texture will be pla


public bool  autoRotation=false;// enable or dissable
public float autoRotZSpeed=1.0f;//after Z is randomly

public  Transform shaftDirection;//will be used to ori

Material shaftMat; //helper variable to acces shared m


float randomRotZ;//this value is randomized once in th
```

All variables are adjusted by pointer values in the Generator script.

The Shaft behavior script is attached to each shaft and makes sure all settings can be updated in real-time.

## Generator script

```
/// local variables for the generation/instantiation: rows or t

public Vector2Int castRows = new Vector2Int(10,10);//each vect
public float    shaftSpacing;//distance between each shafts po
public bool meshCast = false;//toggle between castingmode.
public Mesh castMesh;//each vertex point will cast a shaft.

public GameObject lightShaftPrefab;//will be hidden for the us
public string savePath;//will be hidden for the user.
public bool shaftSettings;//will be hidden for the user. conta

public Material shaftMat; //helper variable to acces shared ma

//Pointer variables to which Shaft_behavior syncs its variable
////////////////////////////////////////////////////////////////
public Color pointer_shaftColor= new Color(255,197,96,255);//s
public float pointer_shaftIntensity=22.0f;//sets the shader's

//  public  Mesh pointer_shaftMesh;//users can set custom mesh
public  float pointer_maxLength= 50.0f;//used in the raycast,
public  LayerMask pointer_layerMask=0;//filter against what th

public float  pointer_shaftWidth = 1.0f;//set the shat's width
public  float pointer_shaftAdjustY = 1.0f;//set the Ytiling of
public Texture  pointer_shaftTexture;//this texture will be pl


public bool  pointer_autoRotation = false;// enable or dissabl
public float pointer_autoRotZSpeed = 1.0f;//after Z is randoml

public  Transform pointer_shaftDirection;//will be used to ori
```

The generator controls how shafts are spawned and to which shared-material + texture they belong.

It is also the container for the Shaft behavior pointer variables. Which in turn are made visible in the inspector through an editor script.

# FAQ

## How do I get started?
Simply go to GameObject->Light->LightShafts_2. This will place one generator in your scene.

## What is the performance?
LightShafts 2 has been optimized to use as much sharedmaterials/meshes as possible. This makes it possible to use hundreds of shaft per generator. However even more performance can be squeezed out by baking the shafts into a single mesh (sacrifising dynamic features such as realtime  collision). The latter is highly recommended for mobile/web platforms.

## I have two generators, but both use the settings of the first one. Can I use multiple generators using different colors/textures?
The system uses sharedmaterials. Meaning if two generators use the same materials, they will get batched based on the first generator using said material.

In order for a generator to use a different color/texture, simply create a new material, assign the lightshaft shader and place this material in the material slot (see inspector overview).

## Where is that cool LightShaft demo using the Adam movie interior?
The demo itself is not included in the asset package because it is a full project. Which means it will adjust all your engine settings.

~~There only exist a Windows 64bit build of this scene, because the project folder got corrupted during build. The 64bit was the first only only build to succeed. I might re-do this scene in the future for other platform builds. But for now, the school room will have to do.~~

## Can I order a pizza?
No, but you are always free to contact me for help or other asset related questions!
http://markduisters.com/contact/