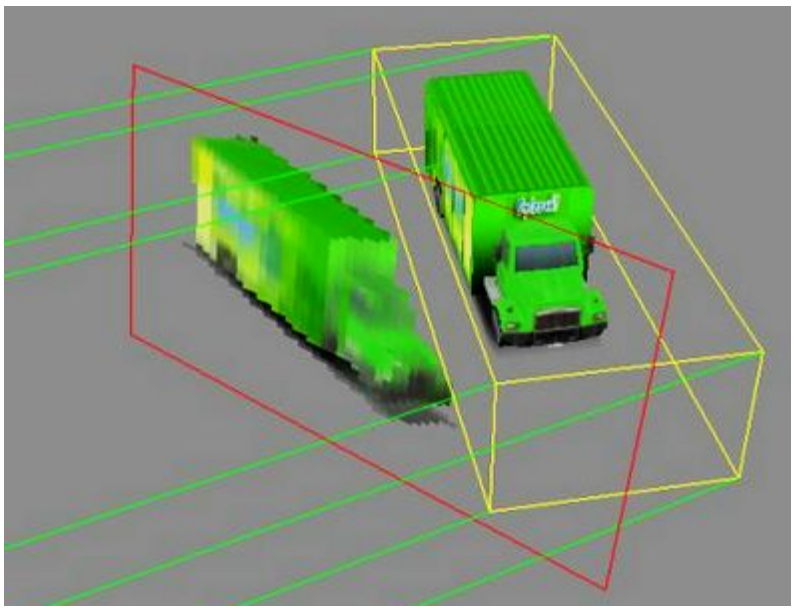
# Real Imposters LOD-System

## About

Imposter Rendering is an image-based LOD procedure. Instead of the actual 3D object, a billboard with a 2D image of the object. By doing so resources can be saved by using less polygons. In similar systems, the 2D images are calculated beforehand and saved from several viewpoints in advance. Generally that leads to very high memory requirements, as well as massive pre-processing time. In the case of “Real Imposters LOD System”, the texture of the billboard is dynamically generated during runtime, and updated depending on the current camera position. As a result the same image can be used for the Imposter over several thousand frames. Only when the camera’s position in relation to the object significantly changes, the Imposter is re-generated.



*3D model (yellow) and the Imposter (red) generated for the current camera angle (green)*
*http://www.gamasutra.com/view/feature/130911/dynamic_2d_imposters_a_simple_.php*

# Setup

1. Create an Imposter Manager by dragging the corresponding Prefab into the scene view or the hierarchy view.
2. Then adjust the vital parameters of the Imposter Manager:
   - First, select the **Imposter Layer**. It should not be used by any other Gameobject.
   - (optional) If the Imposter should shadow other Gameobjects, activate the option **Cast Shadow**. Furthermore:
     - Assign the directional light that you want to cast the shadow to the parameter **Directional Light**.
     - Ensure that this light and other are also active for the Imposter layer.
     - Add the Imposter camera (child object on the Imposter Manager) for illuminating relevant image effects of the main camera (such as SSAO)
   - If you main camera is tagged “maincamera”, you don’t have to make any further adjustments. Otherwise, remove the checkmark from the parameter **use Main Camera** and assign the corresponding main camera manually.
3. Add the Imposter script to your Gameobject or Prefab. The script works recursively and automatically includes child elements. That means these don’t need their own Imposter script.

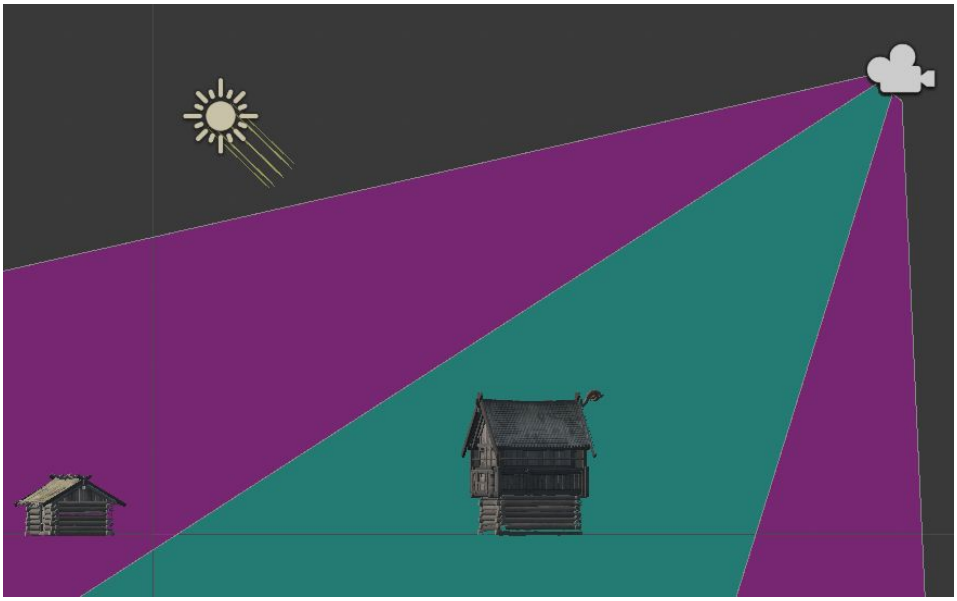
# Advanced Usage

The Imposter Manager script and the Imposter script both have their own default settings. These should be appropriate for 90% of possible applications. If you want to, you can adjust the parameters yourself:

## Imposter Manager

- **Active** - activates or deactivates the Imposter rendering
- **Imposter Layer** - the layer on which the objects are rendered into Imposter textures. Should not be used by any other Gameobject, and should be illuminated by all relevant lights.
- **Use MainCamera** - is the camera tagged “maincamera” to be used as the main camera?
- **Camera** - Main camera position for which the Imposters should be generated.
- **Texture Antialiasing** - Is anti-aliasing to be applied to the Imposter textures?
- **Cast Shadows** - Are the Imposters to cast shadows on other objects?
- **Directional Light** - Light to cast shadows.
- **Caching Behaviour** - What is to happen with non-visible Imposters?
  - **Discard Invisible Imposter** - The Imposters are destroyed to save memory.
  - **Cache Invisible Imposter** - The Imposters are temporarily saved and reused. This requires additionally memory.

- **Preload And Cache Invisible Imposter** - The Imposters are temporarily saved and actively pre-loaded in order to accelerate the rendering during the camera movement. This requires even more memory.



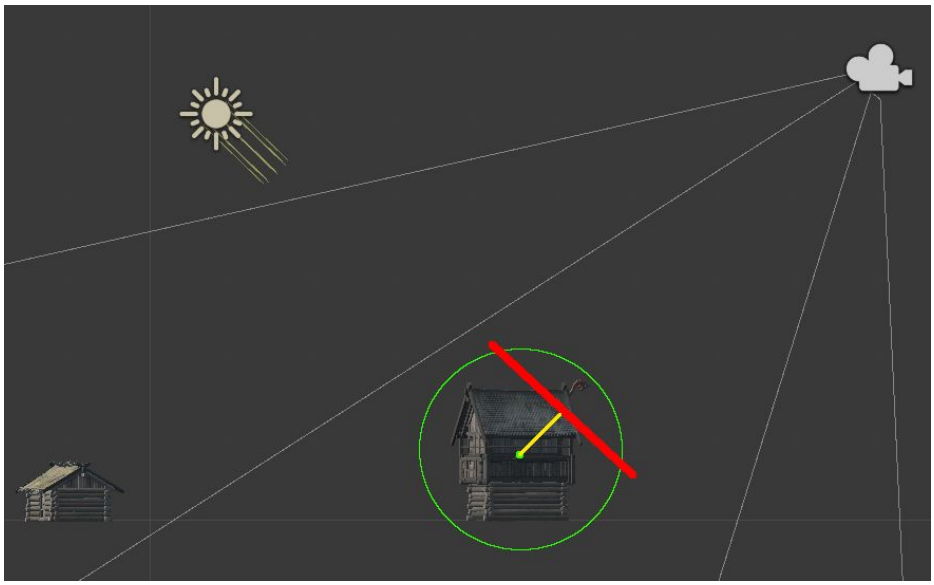
*Area of view of the main camera (teal) and the area in which the Imposters are pre-cached or pre-loaded (pink) for Caching FOV Factor = 2.*

- **Caching FOV Factor** - In which area around the field of view of the main camera should Imposters be pre-cached or pre-loaded? A value of 2 means that this area is twice as big as the visible area of the main camera.
- **Preloads per Frame** - How many Imposters should be pre-loaded per frame at maximum?

## Imposter

- **Lod Metric** -How should it be determined, when the switch between the actual Gameobject and the Imposter is done?
  - **Distance** - At a certain distance, the Gameobject is switched for the Imposter.
  - **Screensize** - At a certain size of the Gameobject on the screen, it is switched for an Imposter.
- **Distance** - At what distance shall the switch to the Imposter be done?
- **Texture Size** - What resolution shall the Imposter texture have?
- **Z Offset** - How far in front of the center of the object should the Imposter be displayed? The value is in relation to the total size of the object. If the value is too small, the Imposter might clip through the ground. If it is too big, it might cover other objects.
- **Cast Shadow** - Should the Imposter be able to cast a shadow on other objects? If yes, an additional shadow imposter is added at runtime.
- **Max Shadow Distance** - Maximum length of the shadow cast by the Imposter. This is important to determine when the shadow imposter is no longer displayed.
- **Shadow Downsampling** - How much lower, in relation to the Imposter texture, should the resolution of the shadow imposter texture be?

- **Shadow Offset** - Distance of the shadow imposter from the center of the object analogue to the Z offset.
- **Max Angle Error** - Starting from what change of angle of view of the camera in relation to the imposter in degrees should the imposter be updated.
- **Max Distance Error** - Starting from what change of the distance from the camera in percent should the Imposter be updated. For example: If the distance from the camera, since the last update, is at 100 units, that translates into a value of 15, so the next update will occur at a distance of > 115 or < 85. This value should be higher the higher the field of view of the camera is in order to compensate for the perspective-based distortion.



*Maximum size of the object (green), the generated Imposter (red) and the Z offset (yellow).*

## Troubleshooting

- **The shadow of the Imposter is only partly visible!**
  - ➢ increase the shadow offset
- **On the edge of the visible area of the camera, the shadow of the Imposter sometimes disappears!**
  - ➢ increase the Max Shadow Size
- **The Imposter clips through the ground!**
  - increase the Z offset
- **The Imposter is illuminated differently from the actual Gameobject!**
  - ➢ ensure that all lights also affect the Imposter layer
  - ➢ Possibly add effects of the main camera (SSAO) to the Imposter camera (in the Imposter Manager)
- **No shadow is displayed!**
  - ➢ Enter the the light that casts the shadow into the ImposterManager
  - ➢ The light must be active and the shadow type must be set to hard or soft
  - ➢ The option Cast Shadow must be activated in the Imposter Manager and in the Imposter