

# Recommended way to build Sceneryobjects

## Table Of Contents

- [1 In General](#)
- [2 Polygon count, resolution and lots of details](#)
- [3 Drawcalls](#)
  - [3.1 Rules](#)
  - [3.2 Exceptions](#)
  - [3.3 Splines](#)
  - [3.4 Advanatges and disadvantages of loop textures](#)
  - [3.5 Important facts:](#)
- [4 Choosing a texture format](#)
- [5 LODs](#)

## 1 In General

The aim of this article is to get in touch with the performance aspect of LOTUS, not to describe the whole process behind LOTUS. We won't define strict values for textures or polygon counts. Each content creator can define them on their own depending on the target the content creator is aiming to. He may not keep an eye on people with less PC power or what so ever.

This article is written in february 2018 so there might be slight changes in the future.

## 2 Polygon count, resolution and lots of details

There is no reason not to keep an eye on the polygon count or the resolution of textures. Before chosing the amount of details on each object some things should be thought of:

Technically:

- Graphics cards these days need way more time to calculate materials than polygons
- The framerate won't get affected by the size of a texture. The renderer will always calcute with the pixel your monitor has. So the texture size doesn't really matter at all.
- The meshes and the textures are stored in the VRAM to get rendered by the graphics card. There is an option in LOTUS, where LOTUS only loads textures with a smaller size. This means textures with a high resolution will get a lower resolution. This is useful when having a not that good graphics card.
- Textures affect the VRAM more than the meshes do. A single 1024x1024 textures has over one million pixels, a vertex is like less than 10 pixels. So a mesh needs hundreds of thousands vertices to really affect the VRAM.
- Keep in mind that the loading time from the harddrive is still a thing, while the storage speaking of terabytes doesn't matter anymore.
- Each type of object is stored in the VRAM once at a time even there a several objects of this type in the near environment of the player. All objects are getting rendered which has no effect on the framerate even if there are different types of objects. On the other side this affects the VRAM a lot.

Strategy:

- Think about the distance between the user and the object. A tram stop will be seen a lot longer than some stuff across the street.
- Also think about the visibility of a polygon. Houses in a street with no public transport doesn't need to be that detailed. The backside of a building is also not often visible to the user, so it should be a plain wall in the best case.
- An object which blurs with the background or objects that are not so unique shouldn't be that detailed. Traffic lights are often seen by the user, traffic signs or other street equipment should be build to create the atmosphere of a complete city.
- The dimension of an object on the monitor defines the maximum value for the resolution of the texture. If a polygon is as height as 20 % of the height of the monitor, the polygon will be 216 px height on a full HD monitor. When having a 4K monitor the value gets doubled. So it would be nonsense to chose a texture bigger than 1024<sup>2</sup> px. Exceptions are made by buildings which are likely bigger than 20 % of the height of the monitor.  
If the question is to build a very detailed mesh, keep in mind that a cylindrical mesh with a dimension of 10 px should not have more than 8 corners. On the other hand the map editor splits the railtracks way more often, because in the drivers perspective non smooth curves are very frustrating to watch.

### 3 Drawcalls

A Drawcall is a basic part of the rendering, which renders a group of triangles with the same material settings. So when a single object has several material settings, the same amount of drawcalls are needed to display this object. A single drawcall with tons of triangles needs less time than more drawcalls with only one triangle.

A Drawcall only uses one texture and one material at a time, so there are at least as many drawcalls as textures. The conclusion of this is to reduce the amount of textures within one object.

#### 3.1 Rules

Only use a small amount of big textures and put several textures in one big texture.

#### 3.2 Exceptions

There are some exceptions, where advantages from other strategies count:

- If there are several buildings with the same roof texture, then there will be two advantages:
  - The roof texture will be once loaded to the VRAM. Keep in mind, that the roof texture shouldn't be part of the facade texture.
  - In this case LOTUS create one big object with the same texture, so the roofs are one drawcall.
- Loop textures may be cheap, but aren't suitable for a texture atlas. Read more in the next chapter.

#### 3.3 Splines

Splines won't be display based on the raw data. For each tile and Spline class (normal, surface, marking) all splines get combined.

Splines are made in Blender, so there is no need to look for an atlas texturing except of the material settings. If the basic model is valid in Blender, then the Spline is also valid in LOTUS.

The conclusion is to use the same texture(s) for the splines with the same class and the same material settings. So the curbstones and the sidewalk splines should use one texture.

### **3.4 Advanatges and disadvantages of loop textures**

Loop textures are smaller textures, which repeat on one or more polygons. An example is a roof texture with a small amount of roof tiles, which repeats over the whole roof top. The benefit is to map bigger polygons with a small texture in a high definition. On the other hand, this doesn't match the concept of a texture atlas.

A good soluation is the tiling of the roof with a simple mapping. In this case the amount of polygons increase a lot, but the costs of the rendering are in these days highly influenced by the material settings, the amount of pixels on the screen and the drawcalls. The complexity of the mesh has no real impact, beacuse the polygon count of a building is not that high.

This method allows us to "fake repeat" the texture atlas and decreases the amount of drawcalls.

As an example we use for our row houses one big texture atlas with two backside textures, two side textures with no windows on it, four roof tile textures, a roof cardboard texture and several textures for objects on the roof top. So every house, that matches this pattern has two drawcalls for the whole object.

#### **Keep in mind:**

When choosing this method to map the textures keep an eye on the night textures! Texturing a prefabricated building from the former eastern germany with a single tile would light all windows at once. The repetition of a small amount of windows will also lead to bad looking pattern at night.

A possible solution is to choose a bigger part of the building (may be 4 floors and 8 windows in a row). There might be a pattern afterwards, but it is not that showy.

In case the windows are on seperate textures/polygons it is possible to use one window for each version of the window (enlighted or not enlighted at night) on the same texture. The windows could be mapped in a random order on the bulding, so there won't be a pattern.

### **3.5 Important facts:**

The material settings should be all the same to get an effect out of the optimizations. Within the object the same material should be used, in a bunch of objects (e.g. Splines) the focus is on the material type and its parameters. Only in this case LOTUS can simplify the amount of drawcalls.

## 4 Choosing a texture format

LOTUS can use bitmaps (\*.bmp) or DXT-compressed textures (\*.dds).

DXT-compressed textures have the advantage to get stored on the harddrive and in the VRAM in a compressed format, so they won't occupy that much space. A disadvantage is that the compression will cause a data loss and the alpha channel only has 16 different values than 256 in a bitmap.

Bitmaps will get compressed by the Content tool, but they have to be decompressed in the VRAM.

JPG files are not supported as they will be decompressed in the VRAM anyway, so they only have the advantage of not using too much hard drive space, which isn't a problem anymore. The usage of the VRAM has no difference to a normal bitmap.

When preparing the textures for an upcoming 3d modell keep in mind that JPG files are easier to handle as long as they are not imported into LOTUS. Afterwards there is no advantage using a JPG file.

## 5 LODs

LOD ("Level of Detail") is a method to optimize the display of objects in a far distance. At first the object will be created as usual, then there should be lightweight versions of the object too. To achieve this:

- the amount of textures or the resolution of a texture can be smaller (but uses more VRAM),
- the complexity or the amount of meshes can be reduced,
- or the complexity or the amount of materials can be reduced.

Because of the rendering the reduction of the amount of textures should be a thing. Afterwards the mesh could be checked for unnecessary things. If the object is not part of the distant tiles or the reduced mesh won't differ that much from the original mesh, then the effect can't be measured. As said before the vertices count has only a small influence on the performance.

If the object is part of the distant tiles and will get combined to a single mesh, then the effect should be very high. The problematic storage of these combined meshes will be reduced very good, especially when these objects are placed very often.