Content Design Guidelines

Embedded 2D/3D hardware is optimized for low power consumptions and reliable operation in industrial environments (e.g. very low or very high temperature). The processing power of embedded devices thus can’t be compared to desktop graphics hardware which has significantly less limitations in computing power, memory, power consumption, and cooling requirements.

Generally, 3D models need to be optimized to make use of the strengths, and to minimize the impact of limitations set by the embedded graphics hardware. With 3D models tailored specifically to the capabilities of the embedded hardware, it is possible to achieve a high visual appeal and also a high frame rate.

This chapter explains, what has to be considered in the very early design phases regarding the graphical composition of an application, to meet the requirements of an embedded 2D/3D system.

Content Design in the Planning Phase

Optimized Screen Updates

If possible, avoid effects/animations etc. that would change all content of the screen at the same time. This allows updating only parts of the screen that really changed. Partial screen updates will improve performance compared to full screen updates.

Define 2D and 3D Content

Consider which parts of the content have to be a fully modeled 3D mesh and which can be flat 2D/3D-Surfaces. If it is possible to use a dedicated 2D-Core, then this would be the preferred way to realize 2D or flat surfaces. If only a 3D-Core is available, it is often preferable to use flat meshes with textures instead of complex 3D meshes. Good candidates for flat surfaces are objects that are only seen and lit from one direction.

Anti-Aliasing Requirements

Anti-Aliasing is a costly technique and should therefore only be used sparsely. If the platform supports layers, it is advised to make the layers with content which needs to be anti-aliased as small as possible. The need for anti-aliasing can also be avoided if textures are used that are fully transparent on the edges of a mesh which can be easily done for most flat objects.

Content Separation by Dynamics

Enable separation of content regarding its dynamic during runtime. Less dynamic content can be rendered less frequently which leaves more performance headroom for rendering high dynamic content.
If the target platform supports (hardware) layers, it’s advised to put content like background images on separate layers. E.g. background content doesn’t need to be heavily optimized as much if it can be rendered only once and never updated again.

Zooming Requirements (Level of Detail)

If objects are seen from varying distances additional low-poly variants can be used when the object is far away. A high-poly version should only be used when the object is close to the camera.

Content Creation Guidelines

Simplify Meshes

Often high-Polygon versions of meshes are reduced with the help of automated tools. While this gets fast results the reduced meshes are also often sub optimal. A low poly mesh created from scratch often has fewer polygons than one made with an optimizer while still having the same visual appearance.
Small geometric details of meshes can also be imitated with textures to reduce the polygon count further.

Optimized Billboards

To avoid drawing fully transparent pixels, create special meshes that have no geometry where the texture would be transparent. These meshes don’t have to fit the texture exactly since they should only have very few polygons.
Refer to the following example: Instead of a single Billboard with 2 triangles this mesh uses 40, but also avoids a lot of transparent pixels in the center that otherwise would be drawn:

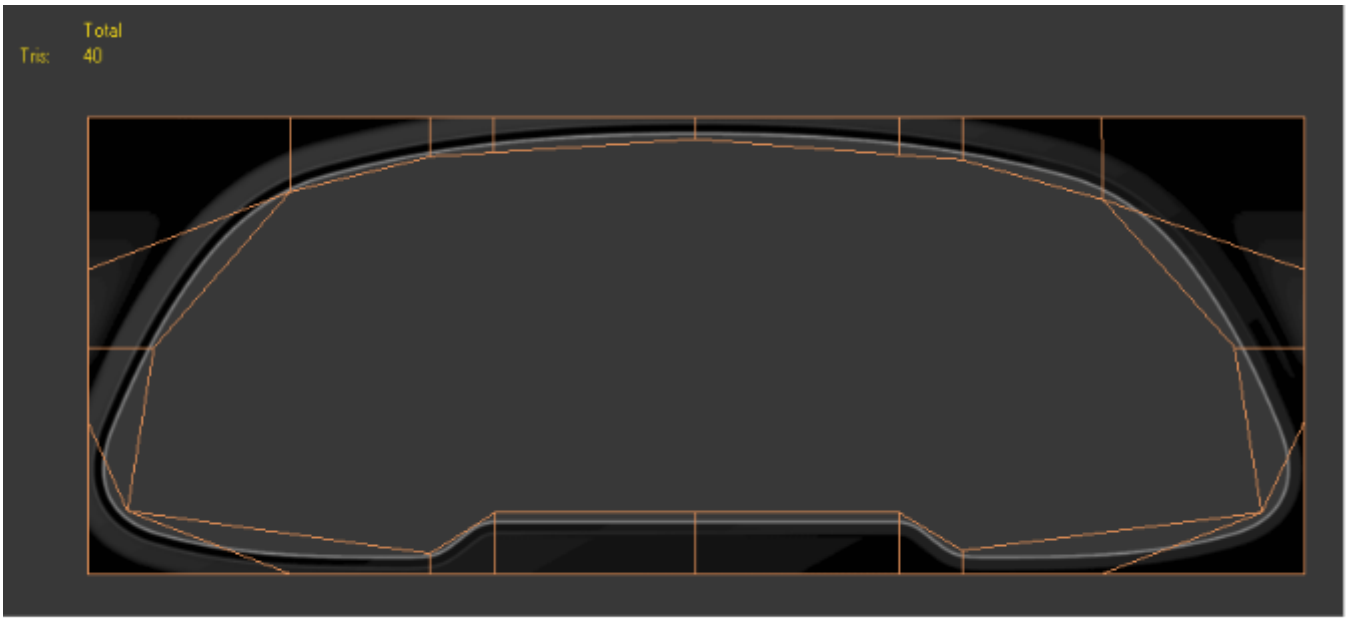


Figure 1 - No Geometry for Transparent Areas

Optimize Number of Nodes

To reduce the number of draw calls object parts that share the same material and are not transparent can be combined into one single object.

Use Level of Detail (LoD)

For the best possible results, it’s also advised to create LoD-Meshes manually instead of using optimizer tools.

Lighting

Lighting is a costly operation so as few light sources as possible should be used. In terms of performance an ambient light is faster than a directional light which in turn is faster than a point light. The slowest light is a spotlight. In the Candera Graphics Engine, all light types can act as an implicit ambient light so in most cases there is no need to create an explicit one.
In any case, avoid scenes with partly covered lights (as might be done in real life). The same effect can usually be achieved with other but less costly techniques.

Textures

The general rule for textures is to make them as small as possible. Textures shouldn’t be bigger than the size they have when displayed on the target system. Though, depending on hardware limitations texture dimensions could be restricted to a power of 2.
Textures can often be reduced further in size through clever usage of texture coordinates and stretching. If the texture is regular, it could be stretched or repeated or mirrored.



The texture here only needs to be 1 pixel in width and is stretched over the whole mesh

Here only 1 quarter of the circle is used as texture and mirrored into the other parts

Figure 2 - Stretched or Mirrored Textures

Also, large empty areas inside of textures should be avoided. This can be done by chopping up the texture and packing the filled parts as close as possible. Then adapt the mesh texture coordinates accordingly. A mesh with a few more polygons and a small texture is often faster than one with a big texture.

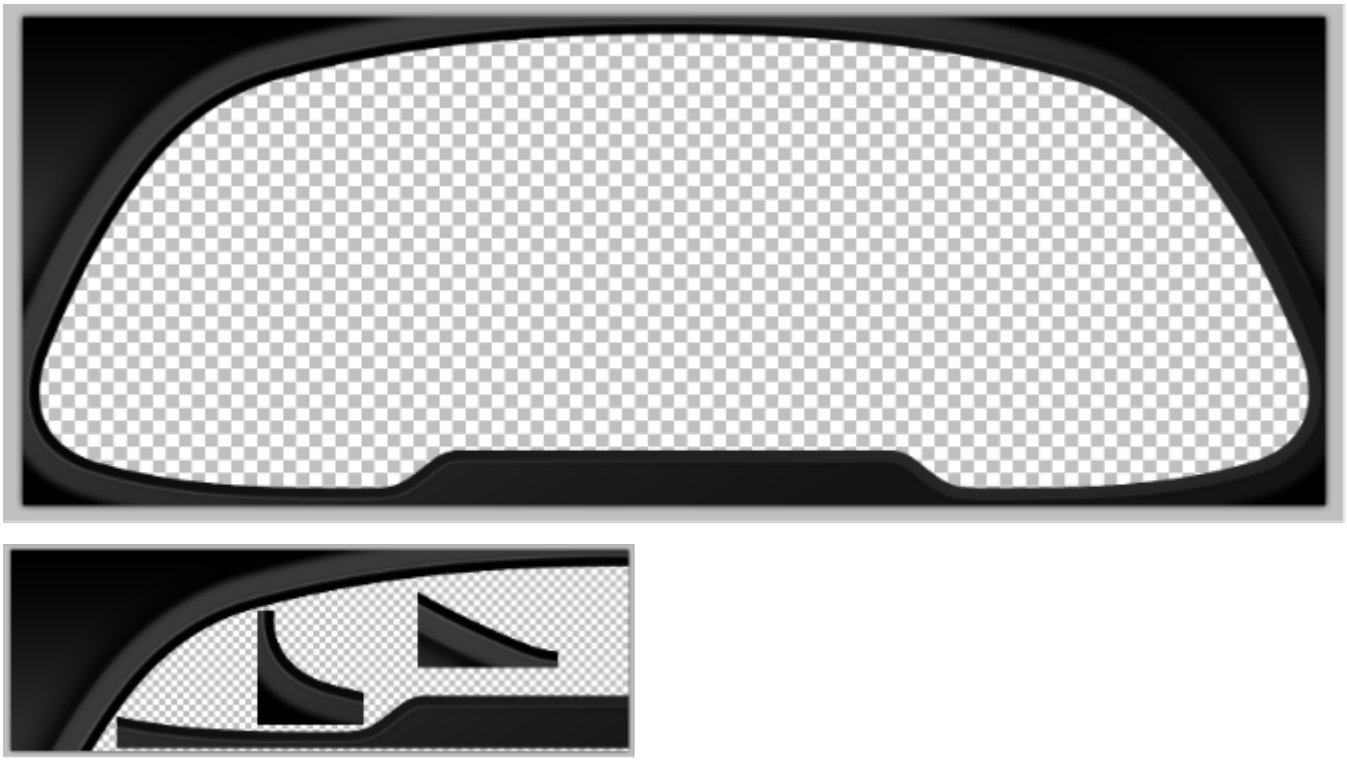


Figure 3 - Packed Texture

In above example, the texture is reduced to ¼ of the original size by cutting and packing parts closer together to avoid large empty areas. To display the object without errors again a special mesh needs to be created much like the one displayed in Figure 1.
Another possibility is to use 16-Bit Textures instead of 24/32-Bit to reduce the size of the content.

DCC Tool Specific Features and Capabilities

DCC Tool specific features and capabilities such as path extrusion, bone animations, shape morphing, particle effects, shadow casting, etc. are often either not transferable by simple export-import process or achieved in a different way on the target system. It is recommended to completely avoid these features or – if desired for early preview – not to rely on the availability of exactly these features on the target system. Nevertheless, most desired effects can be realized using specifically modeled objects and adding the regarding effect implementation with CGI Studio. How to implement a certain effect best needs to be clarified case by case with the Technical Artist implementing that effect for the target.

Revision #2
Created 2023-03-08 06:04:13 UTC by Tsuyoshi.Kato
Updated 2023-03-10 02:50:52 UTC by Tsuyoshi.Kato