

SRGGE - Project 2

Diego Campos Milian

June 11, 2020

1 Time-Critical Rendering

The application allows to visualize a certain number of different models in a tile base scenario. The different present models will be shown with different LODs depending on the outcome of the implemented time critical algorithm. This said algorithm works as follows:

Algorithm 1 Time Critical Rendering

```
1: procedure TCR( $maxTriangles, models$ )
2:    $contribution \leftarrow empty$ 
3:    $numTriangles = 0$ 
4:   for each  $model \in models$  do
5:      $c = diagonal(model) / (2^{LOD(model)} distance(model))$ 
6:      $c* = dot(normalized(distance(model)), camForward) > 0 ? 0 : 1$ 
7:      $contribution \leftarrow contribution \cup c$ 
8:      $numTriangles+ = triangles(model, LOD(model))$ 
9:   while  $numTriangles < maxTriangles$  do
10:     $maxId \leftarrow max(contribution)$ 
11:     $contribution(maxId) \leftarrow contribution(maxId) / 2$ 
12:     $LOD(maxId)+ = 1$ 
13:     $numTriangles+ = triangles(model, LOD(maxId))$ 
14:   while  $numTriangles > maxTriangles$  do
15:     $minId \leftarrow min(contribution)$ 
16:     $contribution(maxId) \leftarrow contribution(maxId) * 2$ 
17:     $LOD(minId)- = 1$ 
18:     $numTriangles- = triangles(model, LOD(minId))$ 
```

As one may notice, this algorithm differs of the one presented in class. In that, to decide which model contributed more to the scene and therefore will increase its LOD we used the difference between its current LOD and the next one. However, this is quite a redundant operation since this difference will always be half the current LOD contribution. This is caused by the fact that in the contribution equation we are dividing by 2^{LOD} , which means that the contribution of a given LOD will always be half of the previous level and the double of the next one, since the other factors diagonal and distance will remain constant for this frame. Therefore, if we precalculate its contribution for its initial LOD we can obtain all the other ones just by dividing or multiplying by 2. Other fact to note is that we only take into account those models that are in front of the camera, this can be performed just with a simple dot operation, as shown in line 6.

2 Visibility

The program allows to enable visibility, this means that meshes that are occluded by the environment will not be drawn. The visibility is recalculated as a random sampling.