# Mortal Shading

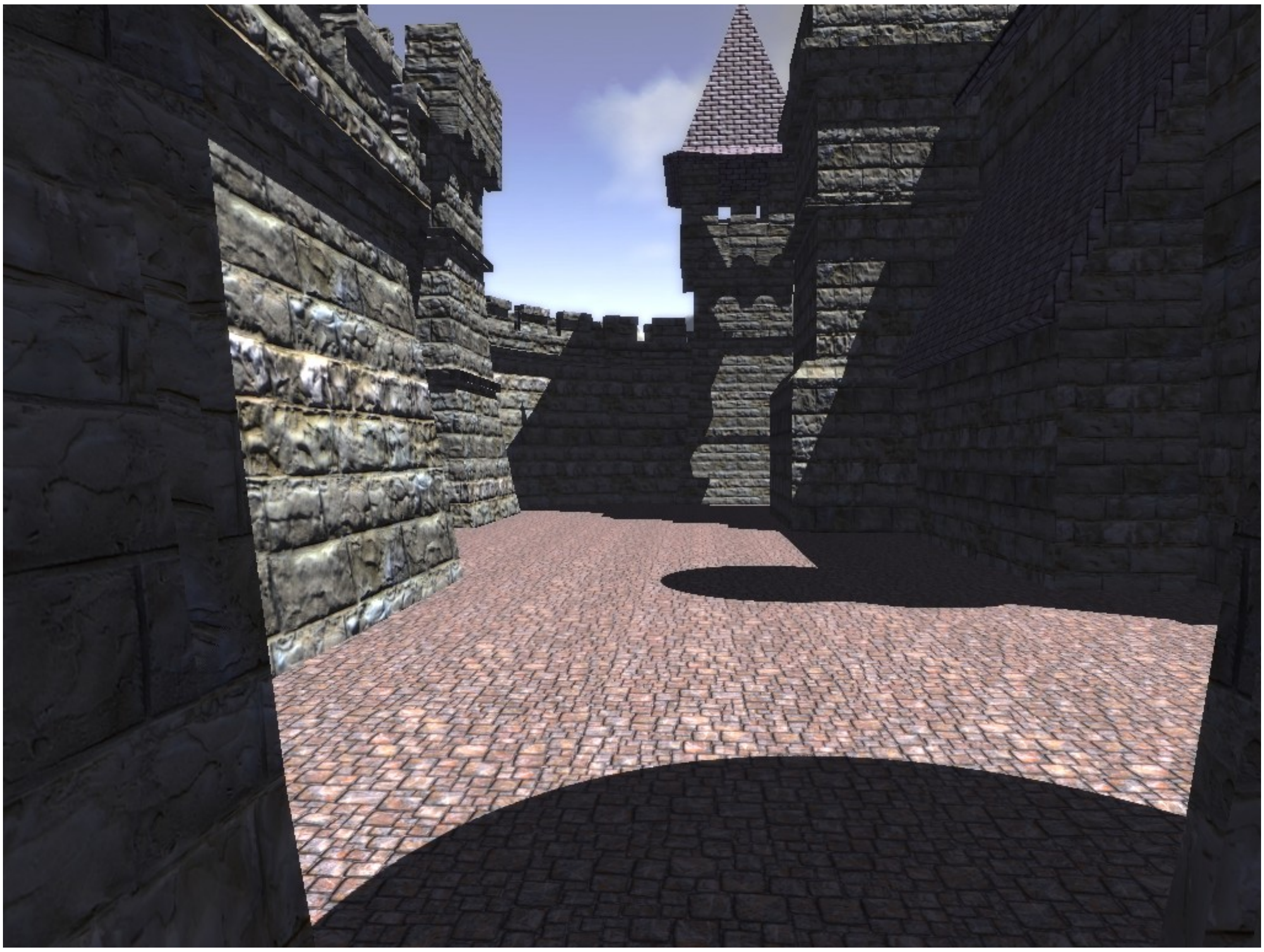
Lukas Pezenka, 0625948
coderonin@darkage.at
Nicolas Swoboda, 0425828
nicolas.swoboda@gmail.com

## 1 Scene

The viewer is taken through a medieval castle inhabited by a small population of dwarves.

## 2 Effects

**2.1 Shadow Maps**
with PCF as presented in the lecture (2011-11-23). Shadows are cast on everything and can frequently be seen in the scene during the camera fly-through.



### 2.2 Skeletal Animation

We use our GPU-based skeletal animation and skinning. This part was programmed in the CG2 Lab (SS 2010). Models and animations are loaded with the assimp library. Dwarves can be seen 5 times in the scene during the camera fly-through.. *http://assimp.sourceforge.net/*

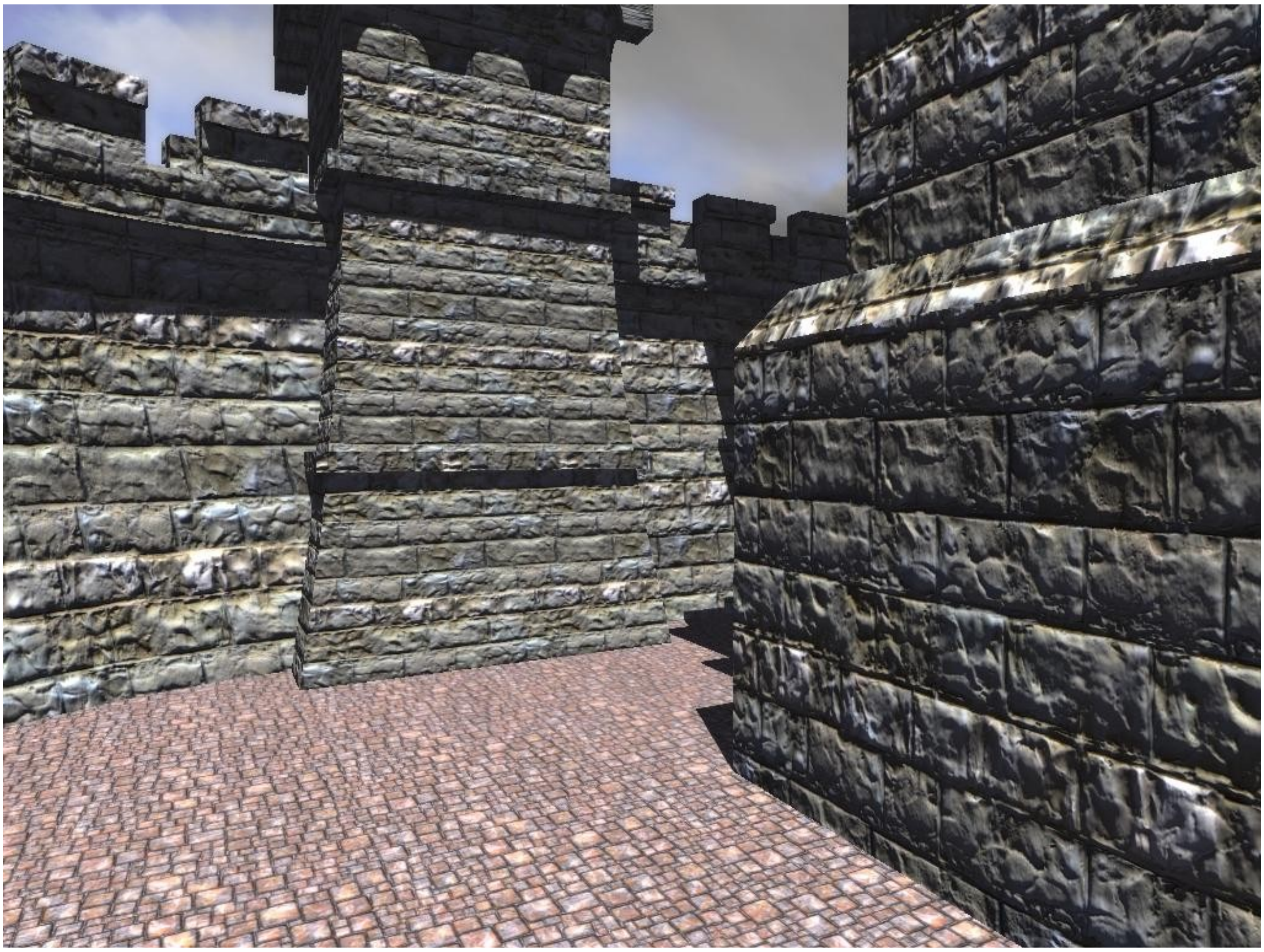
### 2.3 Pseudo Instancing

To speed up the skeletal animation, we'll make use of Pseudo Instancing, i.e. we'll have models of the same type share VBO's and only pass distinct worldspace- and bonetransformation matrices for each instance. Due to information gathered from the Nvidia Homepage ("The only drawback is that it requires special hardware capabilities and doesn't easily support skinning") and relevant discussion boards ("...I can tell you there is no point doing it unless you are CPU bound AND you are drawing 10,000+ instances..."), we decided that true hardware instancing may actually be detrimental to performance in our case.

http://developer.nvidia.com/node/20
http://www.opengl.org/discussion_boards/ubbthreads.php?ubb=showflat&Number=238181

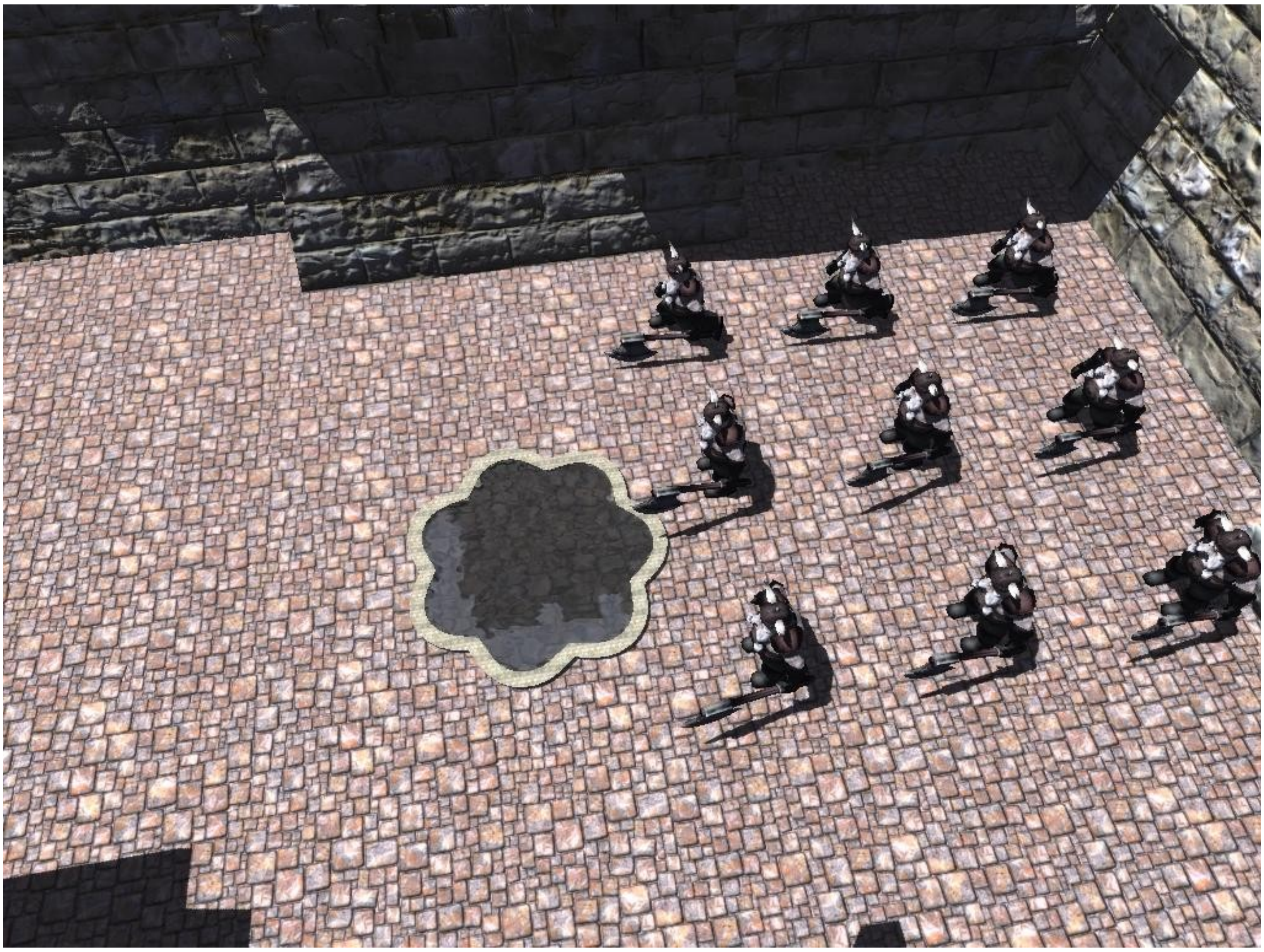
### 2.4 Bump Mapping

In the CG2 Lab we implemented a bump-mapping shader, which will be used for the current project as well. Bump mapping can be seen on every castle wall.



### 2.5 Water Shading

Water grids are placed in the scene around the castle as well as in a fountain on the inside. Reflections and Refractions are rendered in respect to the intersection between the viewing ray and the water grid. Those two terms are then blended together by taking into account the fresnel term. The idea was taken from http://habibs.wordpress.com/lake/.



## 2.6 Bloom

The scene is rendered into an FBO. On the shader, a highlight map is extracted, blurred and blended back into the original image. The basic idea was taken from the lecture (page 7 in the slides an screenspace effects). Our bloom shader is adaptive to average illumination (extracted from the last mipmap level of the scene texture), i.e. it blurs a bit more, when the scene is dark with only a few bright spots. Of help was the post-bloom shader found here: http://fps-creator.eu/index.php?page=Thread&postID=3465, although we passed on implementing this temporal adaptive function.

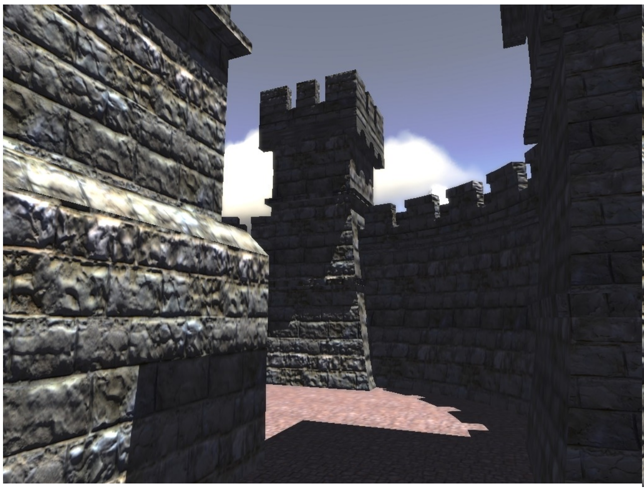


*Illustration 1: with bloom ...*

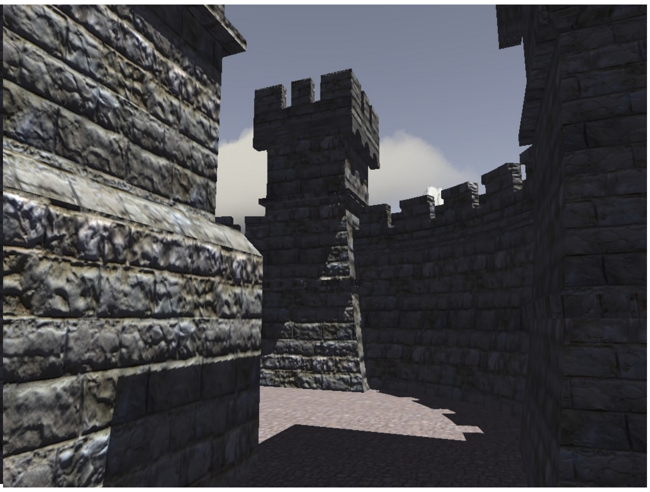


*Illustration 2: ... and without*