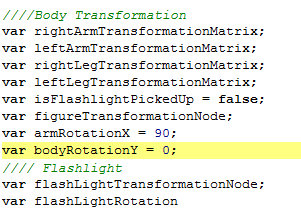
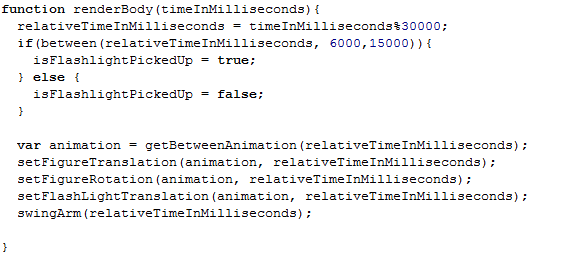
# Basic Movie Effects

## Composed Model

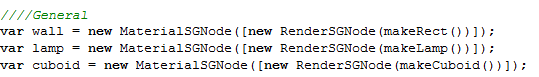
The Composed Model has several different Parts, each of those has an own transformationMatice, which allows individual movements.  
(During the animation, the model is picking up the flashlight, the arms are swinging and the composed Model itself is moving)

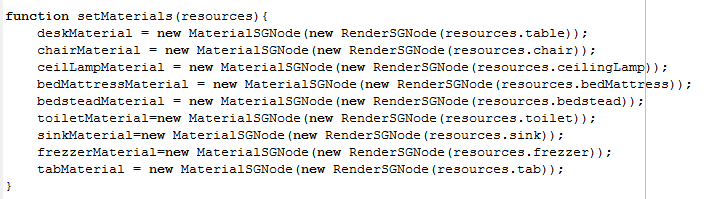




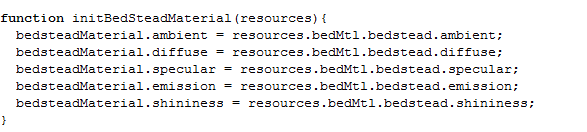
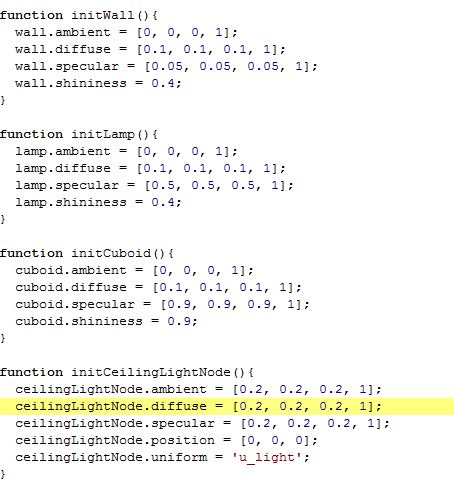
## Material

We set MaterialSGNodes as general Variables or load Models from the resources (for example):



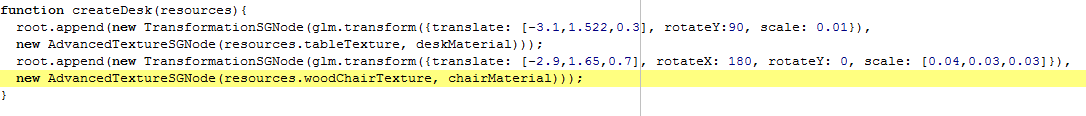


Afterwards within the init Funktion of the Scene, we initialize the nodes with the corresponding Material. The Material is either set manually or stored within the resources (example):

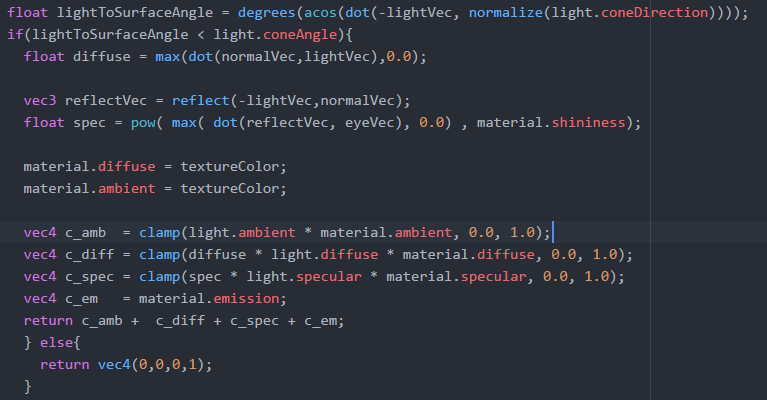
## Texturing

The Textures will be loaded in the load Resource function, afterwards we can apply the Textures to by creating an AdvancedTexturesSGNode.   
(all nodes except the water and the kitchen Table have a Texture applied with AdvancedTexturesSGNode)



## Illumination

For the spot light we check the angle between the fragment and the Light Source and the direction Vector of the Spot light. If the angle is smaller, that the parameter coneAngle, normal phong shading is applied, otherwise we do apply any shading.

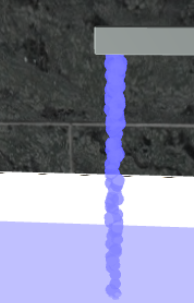


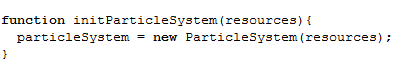
## Transparency

All water within the movie have a fixed color and transparency. Which is set within the watercolor shader. (The nodes, which support transparency are part of the bathtub and the particle System).

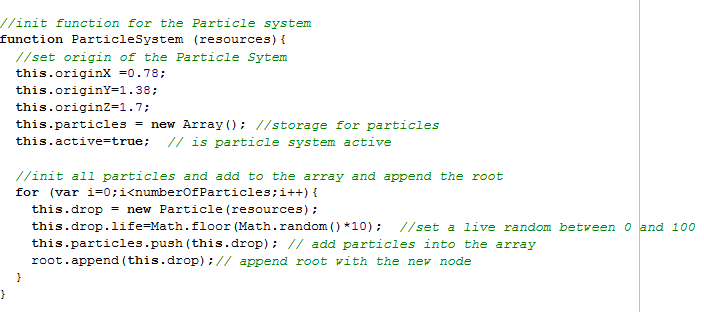
# Special Effects

## Particle System

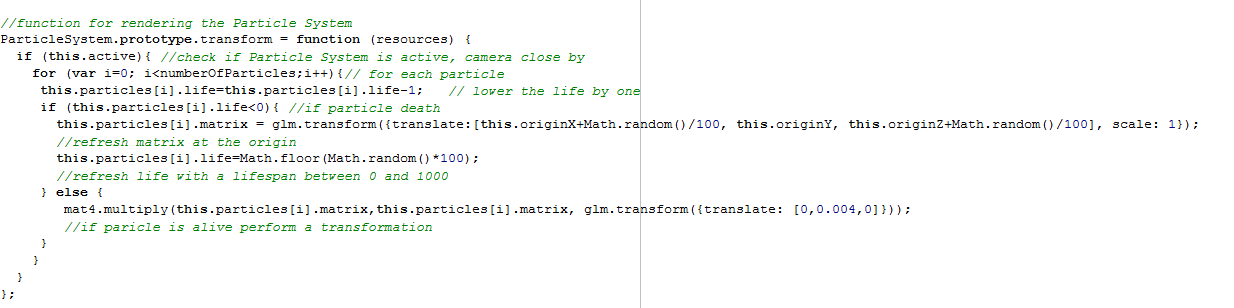
Initialize the Particle System:



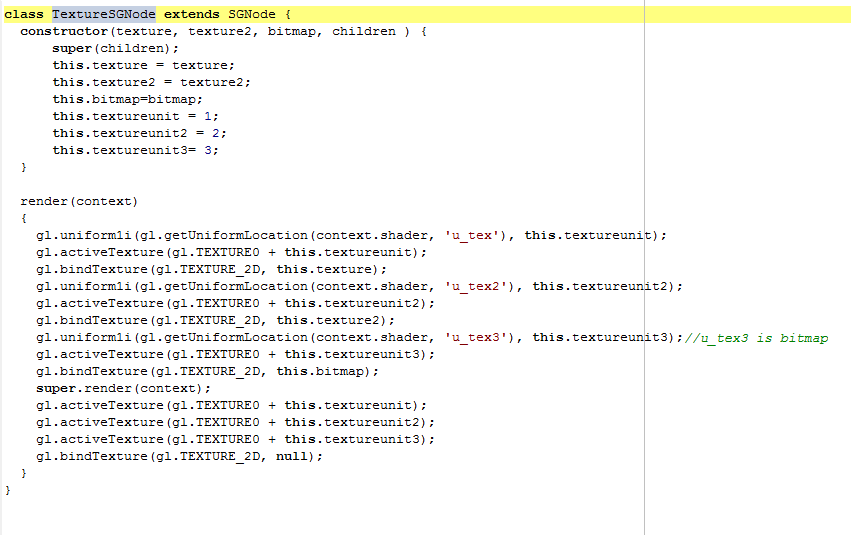
The Particle System contains a position, which describes the origin of all particles as well as an Array, which contain all generated particles. (We choose to render Spheres instead of triangles since a sphere is more fitting for describing water)



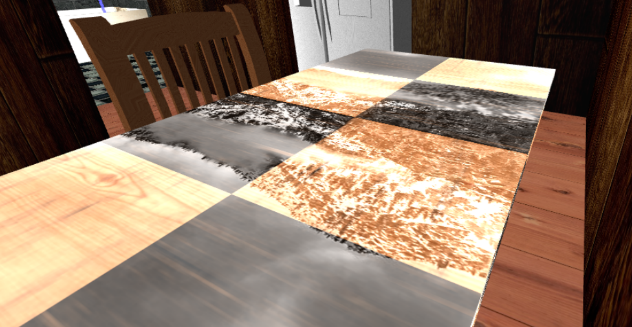
The particle itself contains a life and a transformationmatrice. Each time a particle moves we reduce the life by one. If the life reaches 0 we will refresh the particle with a new life and the tranformationmatrice will be reset at the origin.



## Multitexuring

First of all, the Textures will be initialized with the initMultitexturing(resources) Function.   
Afterwards during the creation of the Scene Graph the Textures will be used for a new TextureSGNode:

(The Multitexuring shader needs 3 textures u\_tex and u\_tex2 main textures and u\_tex3 as Alpha-Map. It generates the Alpha-Value with float c = texture2D(u\_tex3,v\_texCoord).r

Afterwards the covering Texture will be multiplied with the Alpha value and than added to the basic texture: gl\_FragColor=texture2D(u\_tex,v\_texCoord)\*c +texture2D(u\_tex2,v\_texCoord);

# Movement

Tab = change between free and animated Camera flight  
up/down = moving forward/backwards

Mouse = change camera pitch and yaw

# Additional Information

We adopted the given framework, so that Material-Files are now recognized and instantaneous loaded. This was approved by contacting the CG-Staff per E-Mail.