Assignment on Virtual Reality and Physically-Based Simulation - Sheet 2

Due Date November 17, 2018

For this exercises, take a look at the Unreal Engine 4 Documentation and in case of problems also the UE4 AnswerHub.

Figure 1: Possible result of this assignment.

Exercise 1 (Unreal editor, 4 Credits)

In this exercise, the goal is to create an actor constructed from different parts which are arranged in a scene graph.

a) Create a new actor that looks similar to the character in Figure 2. You can model the individual parts with transformed Sphere Mesh Components.

b) Make sure that rotating the head also rotates the nose and the eyes by defining them in a hierarchy.

c) The arms and legs should rotate around the point, where they are attached to the body. Consider Scene components as virtual joints (do not use Unreal joints).

d) Place the actor into a level that is loaded when the game starts.

e) Rotate the legs with a looping run animation.

f) Make the animation speed configurable with a variable in the blueprint that is Instance Editable.

g) Place at least two instances of your actor in the scene. Both actors should have different walking speeds as you can see in the video.
Exercise 2 (Fishtank VR, 6 Credits)

Let us consider the view frustum of a moving spectator in front of a screen. The eye position is changing relative to the screen, so we have to compute the frustum everytime the eye position is changed. For an example see https://www.youtube.com/watch?v=hvrT7FqpFQE. As mentioned in the lecture, the view frustum in OpenGL is set with glfrustum(left, right, bottom, top, near, far). Your task is to find expressions for the above variables for the case of

a) an one eyed spectator (shown in figure 1).

b) for the left eye of a two eyed spectator similar to the lecture.

For the sake of simplicity assume that the observer is looking straight at the the screen (no head rotation). For your calculations assume that the values for the eye position \( p_e \), the screen corners \( p_1, p_2, p_3 \) and the distance to the near plane \( n \) are given.

Note: For a) it might be the easiest approach to calculate \( l', r', b', t' \) and transform them to the near plane.

![Figure 2: Setup one-eyed moving spectator](image-url)