KTH Stockholm :: EECS :: CST
Introduction to Visualization and Computer Graphics
DH2320/DD2258, Spring 2020
Tino Weinkauf, Hazal Koptagel, Wiebke Köpp
Tom Axblad, Zhenghong Li, Shuyuan Zhang

Homework assignment No. 02

Due January 31, 2020

The files `cubeanimator.h/cpp` contain additional comments and instructions for Task 2.

Task 2.1: Transformations — 10 P + 5 EP

This is the first exercise of several in which you will be using Inviwo - an overview and the instructions for installing it can be found on the course Canvas page. When you run Inviwo and open *File* → *Example Workspaces* → *LabTransformations* → *Setup.inv* you should see a window like in Fig. 1. The XYZ global coordinate system is symbolized by the RGB axes of the ball in that order.

(a) The initial configuration is shown in Fig. 1. Transform all of the dice by modifying the parameters of the `World Transform Mesh` processors so that the configuration shown in Fig. 2 is achieved. Note that one `World Transform Mesh` processor can perform only a single transformation, either translation, rotation or scale. For a combination of multiple transformation, you will need multiple processors.
(Hint: Angles within the `World Transform Mesh` *processor are given in radians.)*

(b) (Extra Points: +5)
Transform the 6th die to obtain the configuration shown in Fig. 3.
(Hint: The lower corner of the 6th die is located at the center of the common upper edge of the dice below.)

Task 2.2: Animation — 10 P

Modify the code of the processor `Cube Animator` (in `inviwo-module-labtransformation`) to animate the 6th cube, so that it rotates around the other cubes and at the same time performs a swirly motion with similar distance and frequency as shown in the video `video_animation.avi`.

For this task you will need to add some properties to the `Cube Animator` processor. There are three places where we need to do something with a property in order to add it:

- Declare it in the header file, e.g. `FloatProperty radius_;`
- Initialize the property with an identifier, display name, initial value, min value and max value within the initialization list of the constructors, e.g. `radius_("radius", "Radius", 6, 1, 8)`
- Add the property to the processor in the body of the constructor, e.g. `addProperty(radius_);`

You can animate those properties with a `Property Animator` processor. Add the processor to your network and then:

1. Add a property of the correct type.
2. Connect the animator to your cube animator and click on that connection.
3. Connect the value of the created property with the property you want to animate. You can expand the dark grey property by clicking on it.
4. Change the delta to the value by which you would like to change the property for each animation step. Max and Min value can be adjusted here as well. (Rightclick → Property settings...).

Now you can animate the property by pressing Play. Also see Fig. 4 for these steps.

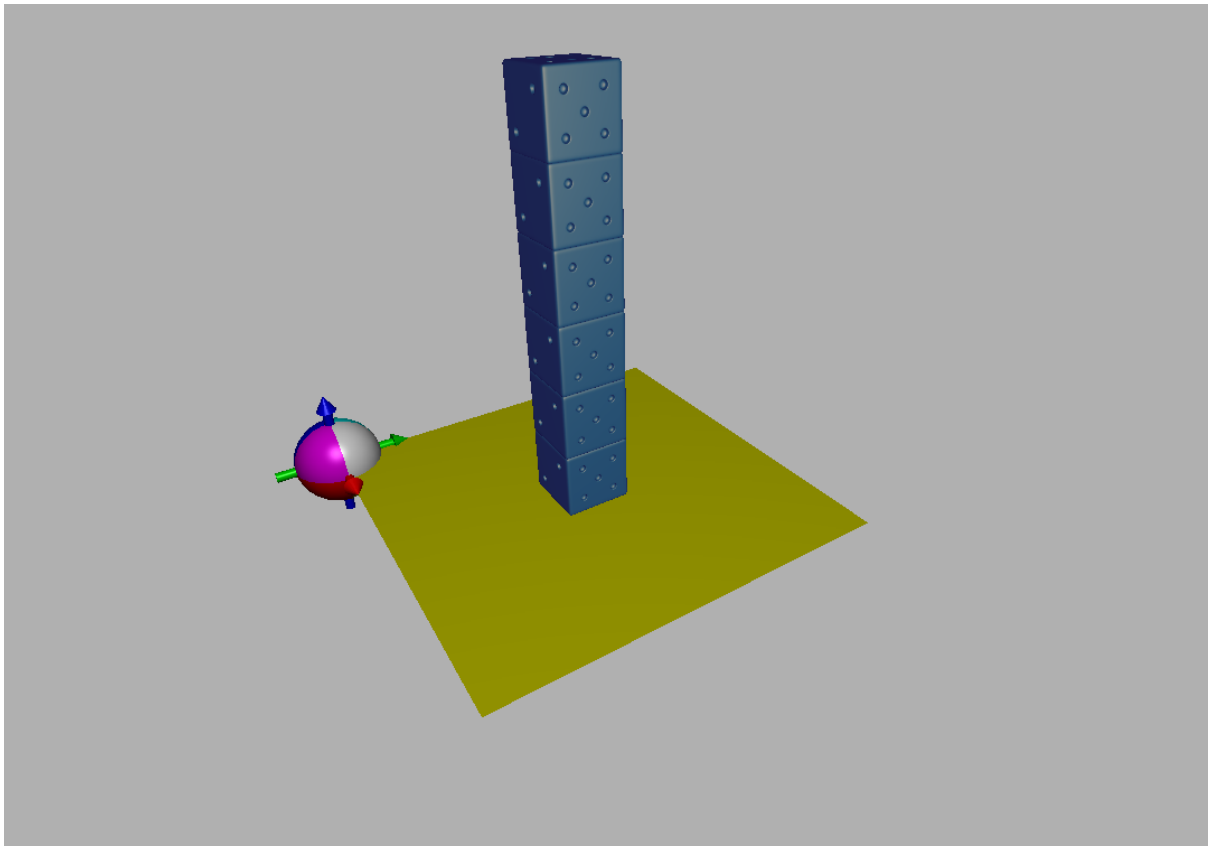

Figure 1: Initial Framework

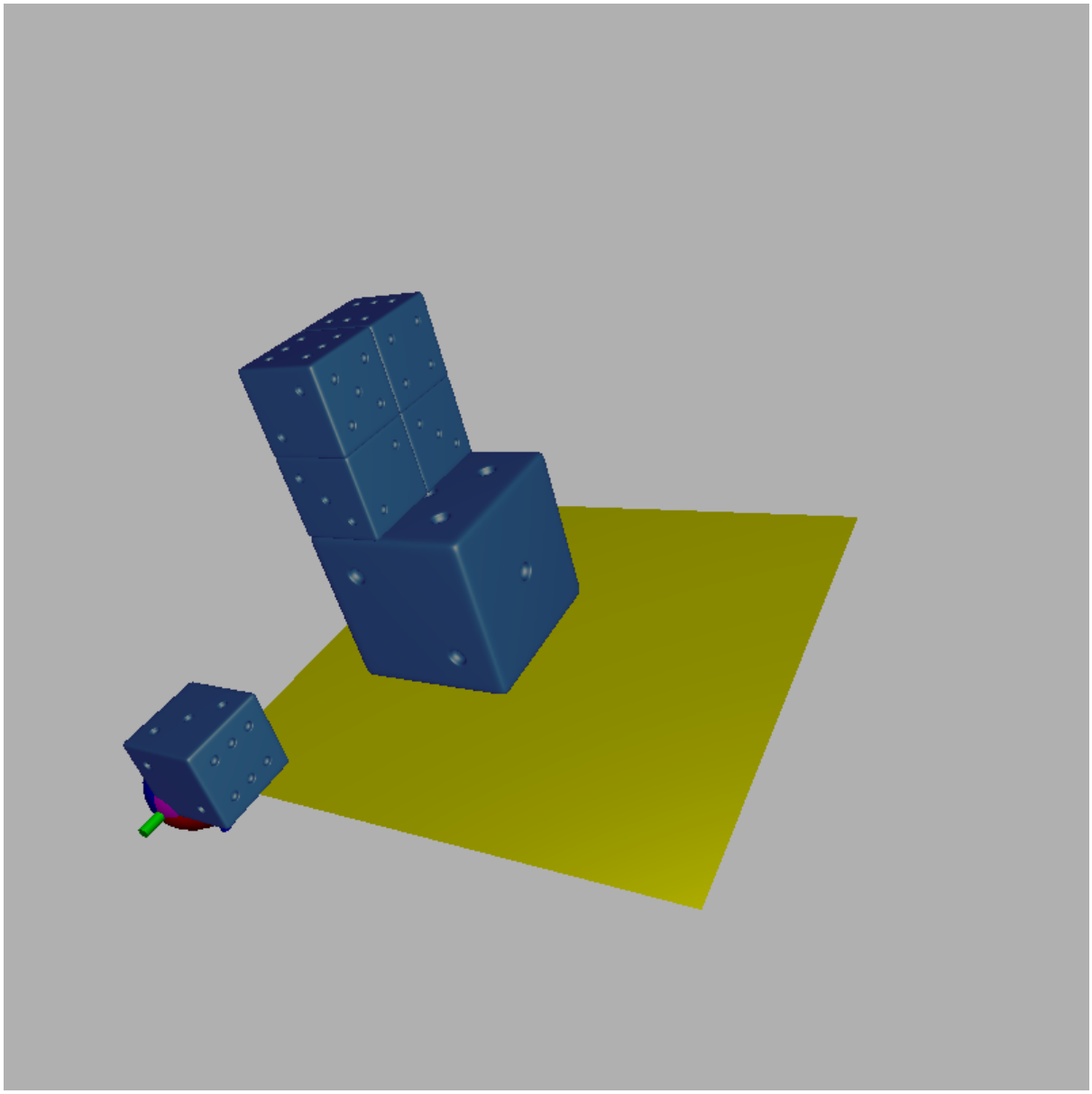

Figure 2: Expected Output for 2.1(a)

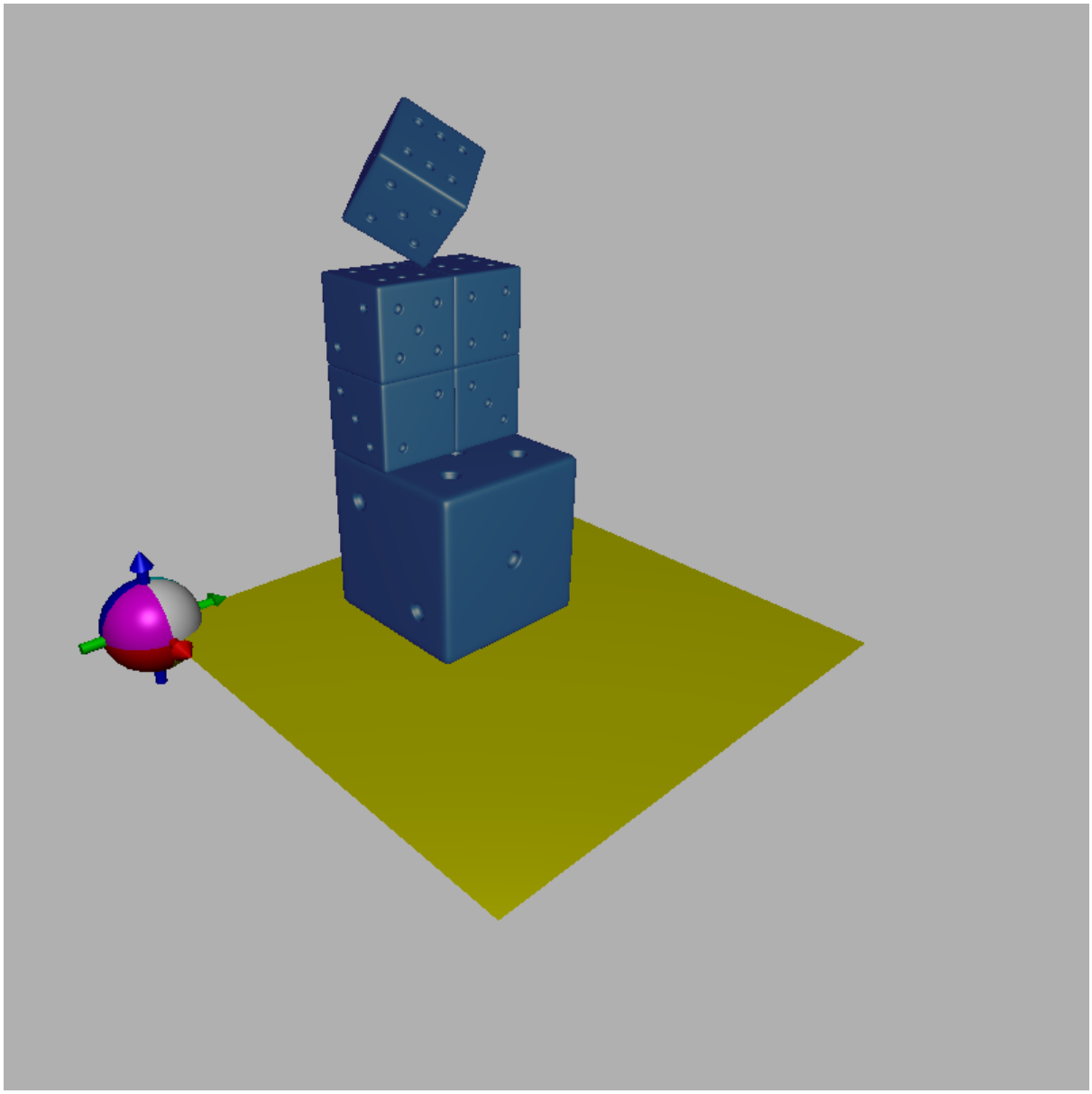

Figure 3: Expected Output for 2.1(b)

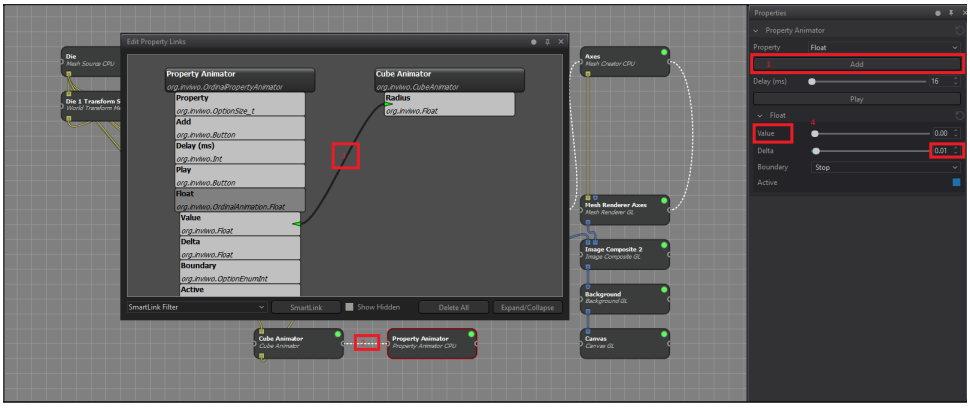

Figure 4: Steps to animate a property.