

# OpenGL - Lab Session 5

## Procedural animation

In this lab session, we are going to animate the model you have created.

Copy the code of `to_copy.cpp` in your on-going program. Call `updateAnimationParameters()` in `renderScene()`.

## 1 Animation loop and time step

### 1.1 Principle

As `updateAnimationParameters()` is called in `renderScene()`, it is called at every display time step. At each call of `updateAnimationParameters()`, the control parameters (also called animation parameters) are updated. Let's first analyse what this function does.

**Mesure du temps :** First of all, a parameter `t` simulates the elapsed time since the last call to that function. This time is normalized by a time step called `deltaT`, which controls the animation speed.

**Animation parameters :** The second part of `updateAnimationParameters()` updates the control parameters (here, a rotation angle : `animRotation`) with respect to `t`. Synchronizing the different control parameters on `t`, the different resulting animations are synchronized.

### 1.2 Application

If we compile and execute the program, nothing happens. Modify the program so that your model reacts to the control parameters. For instance, make it rotate.

**Rotation around a specific point :** When calling `glRotate()`, the rotation is done around the origin of the current reference frame. However, we may want to rotate around any point  $(x, y, z)$ . How can this be done? Implement it in your program.

## 2 Personalization of the movement

You know understand the principle of animating through control parameters :

1. measure elapsed time and decide on an animation time step,
2. synchronize and update the animation parameters,
3. parametrize the drawing functions.

Add other parameters (for rotation, simple translation, translation along a given trajectory, scale, color, ...) to animate your model.

**Tip :** You can create booleans allowing or not one animation parameter to be updated. For instance, in `updateAnimationParameters()`, update `animRotation` if and only if the global boolean `allowAnimRotation` is set to `true`. When the user presses `a`, invert `allowAnimRotation` (to implement in `commonKey()`).