Questions and Exercises

These questions and exercises is an opportunity to see what you've learnt from the lecture as well as practice the new things we've been talking about. In other words, these questions and exercises are completely optional but it's recomended to do them. In the end of the document you will find the answers to the questions as well as possible solutions to the exercises, note that one can solve an exercise in different ways. There will also be some suggestions about what one could code if one want to continue with some more advanced things. These suggestions will not come with a possible solution and might include things that haven't been covered in the lecture.

Question 1
When drawing mouse hovering text how can we change the color of the text?

Question 2
What is the idea behind a tab system? How can we make sure that different things are being rendered depending on which tab is active?

Question 3
To render a model in an interface we can use RendeManager.instance.renderEntityWithPosYaw. Is this enough or do we have to do something else as well?

Exercise
Continue with the code from the exercise of the previous lecture. Add mouse hovering text to it. Add a scrollbar that defines the regeneration speed. Render a spinning cake in the interface.

If you want to you can get a texture to use from the link below.
https://dl.dropboxusercontent.com/u/46486053/CakeBoxTexturesLecture4.zip

Further explorations 1
When you're ready, there's an assignment waiting for you.

Further explorations 2
Continue with the code from the lecture. Allow the user to control the rotation of the rendered anvil with the mouse (by clicking and dragging). Alternatively, do this for the cake from the exercise instead.

Answers and solutions

Answer to Question 1
To specify the color of a text we can simply add a special tag in fron of the text. The tag is different for the 16 possible colors and ranges from \u00a70 (black) to \u00a7F (white)

Answer to Question 2
When drawing an interface we usually code it from within the interface, the interface controls what should be drawn. However, if we put the rendering code inside tab classes (different code depending on the sort of tab) we can tell just one tab to draw its things at any given moment. This gives the impression of different tabs being opened since we'll only draw the content of the tab which is currently active. Exactly how the rendering part of the tabs works depends on the implementation of the tab classes.

Answer to Question 3
The method in the question is the only thing required to actually render the model, but we need to prepare things to make it render properly. What we'll have to do is modify the drawing position, rotation and scale (among other things). It can look something like this:

```
GL11.glPushMatrix();
GL11.glTranslatef(gui.getLeft() + 90, gui.getTop() + 100, 100);

float scale = 30.0F;
GL11.glScalef(-scale, scale, scale);


RenderHelper.enableStandardItemLighting();

GL11.glRotatef(180, 0, 0, 1);
GL11.glRotatef(roll, 1, 0, 0);
GL11.glRotatef(yaw, 0, 1, 0);


RenderManager.instance.renderEntityWithPosYaw(anvil, 0, 0, 0, 0, 0);

GL11.glPopMatrix();
RenderHelper.disableStandardItemLighting();
```

Possible solution to Exercise
https://dl.dropboxusercontent.com/u/46486053/CakeBoxSolutionLecture4.zip