CSC 220 Object Oriented Multimedia Programming, Fall 2021

Dr. Dale E. Parson, Assignment 2, 3D automated avatars in 3D spaces in Processing.
Also mandatory custom vector graphics (PShapes) using vertex() with optional textures.
This assignment is due via **D2L Assignments <u>CSC220F20Assn2_3D</u>** by **11:59 PM on October 10th**.
There will be one in-class work session.

Follow the instructions in our Assignment 1[1] for setting up your sketchbook location or download the Processing environment if you need to do this. You probably don't need to do this.

You can copy and paste the text for my handout sketch into a new sketch, then save it, as a starting point. Create your **<u>CSC220F20Assn2_3D</u>** folder by running File -> Save As -> **<u>CSC220F20Assn2_3D</u>** after setting up your sketch folder.

The handout code is here:
<u>http://faculty.kutztown.edu/parson/fall2021/CSC220F21Assn2_3D.txt</u> (handout code)
HOWEVER, to earn 100% you must complete all redesigns specified below. It must not be a small change to mine; it must be your own visual design. Nevertheless, starting with mine is necessary because substantial parts of the assignment remain intact, including the interface Avatar, the abstract class Collision Detector, the setup() function, draw(), overlap(), and the keyPressed() function. <u>Search and read</u> **STUDENT** <u>comments in the code before starting</u>.

REQUIREMENTS (copied from top of my handout sketch):

```
/*   SUMMARY OF STUDENT REQUIREMENTS:
/*   Save a working copy of this sketch as CSC220F20Assn2_3D WITH YOUR NAME at the top.
/*   1. 50% Completely replace my Professor class with your class that uses 3D boxes & spheres;
/*      it may also use some 2D shapes. Surfaces must adjoin; you could have a free-floating
/*      body part like a halo, but generally the Avatar must be a contiguous 3D image.
/*      I will deduct at least 10 points for a random collection of shapes that don't
/*      look like anything. Make sure to document your class with comments.
/*      getBoundingBox() now returns 6 values per comments in interface Avatar.
/*      TO GET STARTED You could rename Professor to your class name globally, run it,
/*      and then edit your changes into its functions and possibly data fields.
```
ADDED 9/23/2021: **DO NOT** use the variables **myx, myy, or myz** again after this point in display():

```
 // The display() function simply draws the Avatar object.
 // The move() function updates the Avatar object's state.
 void display() {
  // Draw the avatar.
  push(); // STUDENT *MUST* use push() & translate first in display().
  translate(myx, myy, myz);
```

If you use them to place graphical objects then your code effectively adds them to the objects' coordinates TWICE, once in the above translate call, and again in the graphical objects. These variables should appear ONLY in that initial translate call near the top of display().

---

[1] <u>https://faculty.kutztown.edu/parson/fall2021/csc220fall2021assn1.pdf</u>

```
/*
/*   2. 25% Rewrite function makeCustomPShape() to make your own custom 3D PShape used by
/*      my existing class VectorAvatar. Build your PShape using the vertex() function
/*      on one or more planar faces of the PShape. You may use other PShapes such as ELLIPSE,
/*      BOX, etc. in a GROUP, but you must build at least one surface using vertex() calls.
/*
/*      I will award 10% bonus points if your code successfully textures a PShape planar face.
/*      You must use your own image file for this, and make sure to turn it in to D2L.
/*
/*      HINT: To texture, make an individual planar PShape 2D instead of 3D -- no Z coordinates --
/*         and use 3D translations & rotations to put it into place in a GROUP,
/*         similar to my sketch makeCustomPShape().
/*
/*   3. 15% Modify VectorAvatar.getBoundingBox() to enclose your displayed PShape
/*      correctly. Other enhancements to VectorAvatar are optional.
/*
/*   4. 10% Change any constructor calls in setup() to match your constructor.
/*
/*   Do not change classes Furniture or Paddle for this assignment.
/*   Test it and make sure it works. Do not break any of the keyboard commands. TEST THEM!
/*   If you change your key commands, document them at the top like the handout code.
```

We will use a class for working on this project. If you do not get it done in class, you will have to complete it as homework. I expect it to be to me by the due date via D2L. **I will deduct 10% for each day it is late**. Also, re-read the above requirements when you turn it in, to ensure that you don't miss anything. If you make changes after turning it in, just turn in another copy of your sketch via D2L. I will look at the last one that you turn in.

TURNING IT IN: When your work is completed, and you have re-read and satisfied the project requirements, you can use the Windows Explorer to find the file **CSC220F20Assn2_3D.pde** in your sketch folder. Drag **CSC220F20Assn2_3D.pde** into the **Assignment 2 dropbox** under our course's D2L account by the due date. If you find you have created an error, you can drop an updated CSC220F20Assn2_3D.pde into the dropbox. **Assignment 2** is under **Assessments -> Assignments** in our D2L account. If you are working on a laptop or your machine at home, turn it in via D2L in the same way. Finally, if your sketch loads any image (e.g., for texture) or .svg vector graphics files, and/or if you use Processor editor TABS to create multiple .pde files, turn in those individual files via D2L. Note that your custom PShape must use the vertex() function & not simply load a SVG file; you may use a SVG file as part of your avatar. See the spring 2018 csc120 course page for instructions on how to save Illustrator files to .svg files that will work with Processing, if you are an Illustrator user who wishes to do so (optional)[2]. I cannot run a sketch that depends on additional files without receiving those files.

---

[2] http://faculty.kutztown.edu/parson/spring2018/CSC120Spring2018.html