CPSC 314
Assignment 1

due: Wednesday, October 1, 2014, 11:59pm
Worth 9% of your final grade.

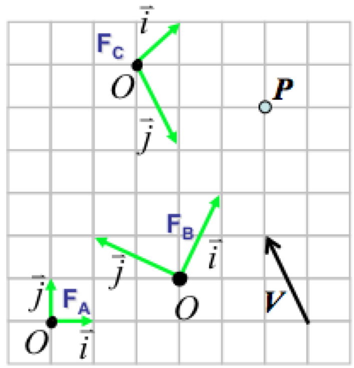
Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: ______________________________

Student Number: ______________________________

Question 1	/ 15
Question 2	/ 9
Question 3	/ 9
Question 4	/ 5
Question 5	/ 4
Question 6	/ 8
Question 7	/ 50
TOTAL	/ 100

1. Transformations as a change of coordinate frame



$$\begin{bmatrix} \\ \\ 1 \end{bmatrix}_A = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \\ \\ 1 \end{bmatrix}_C$$

$$\begin{bmatrix} \\ \\ 1 \end{bmatrix}_B = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \\ \\ 1 \end{bmatrix}_A$$

(a) (3 points) Express the coordinates of point P with respect to coordinate frames A, B, and C.

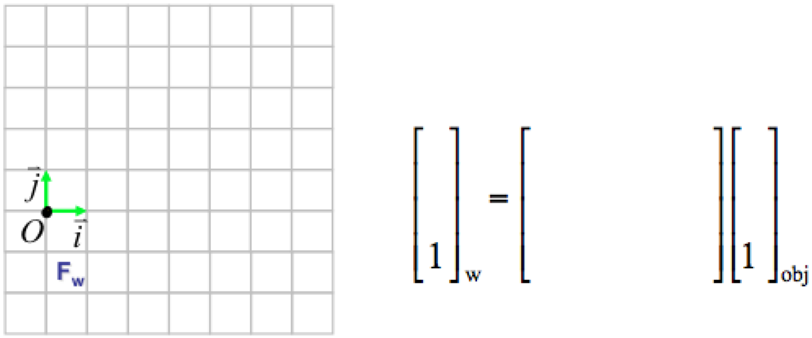
(b) (3 points) Express the coordinates of vector V with respect to coordinate frames A, B, and C.

(c) (3 points) Fill in the 2D transformation matrix that takes points from F_C to F_A, as given to the right of the above figure.

(d) (3 points) Fill in the 2D transformation matrix that takes points from F_A to F_B, as given to the right of the above figure.

(e) (3 points) Using the above two matrices, develop a 2D transformation matrix that takes points from F_C to F_B. Test your solution using point P.

2. Composing Transformations



(a) (3 points) Consider a house in the xy-plane, defined by the coordinates (0,0), (2,0), (2,2), (1,3), 0,2).
Sketch the house after applying the following transformations. Assume that the matrix `m` is initialised to the identity matrix.

```
m.translate(2,4,0);
m.rotate(-90, 0,0,1);
m.scale(2,1,1);
```

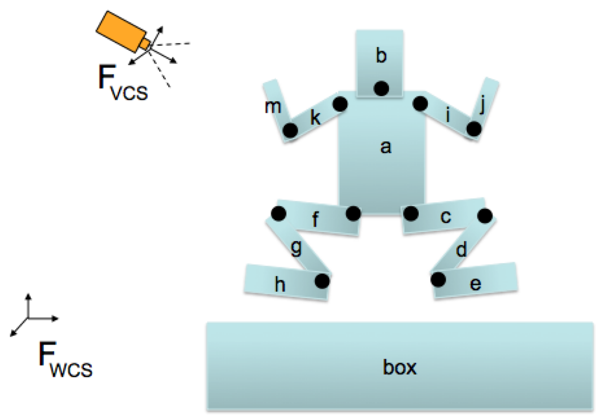
(b) (3 points) Give the resulting 4×4 transformation matrix. Assume that the transformation leaves z to be unaltered.

(c) (3 points) What values would need to be assigned to `theta, sf, x, y, z` in order for the following transformations to yield an identical final transformation?

```
m.rotate(theta, 0,0,1);
m.scale(a,b,c);
m.translate(x,y,z);
```

3. Scene Graphs

   (a) (3 points) Draw a scene graph for the scene with the frog, shown below. Label each of the edges in the scene graph with a unique name, M_{name}, to represent the transformation matrix that takes points from its given frame to its parent frame. Assume that the body of the frog and the box are positioned relative to the world frame, and that the frames of all the other shown parts are defined relative to their proximal parent links, i.e., the links closer to the body.

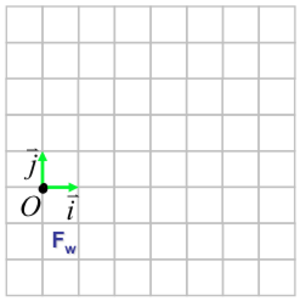


   (b) (2 points) Give an expression for the composite transformation that would be used when drawing link e i.e., to transform points in F_e to F_{VCS}. Your answer should be expressed as a product of the matrices used to label your scene graph.

   (c) (2 points) Give an expression for the composite transformation that transforms points in F_e to F_m. Your answer should be expressed as a product of the matrices used to label your scene graph.

   (d) (2 points) Describe how you could use the previous transformation to compute the distance, d, between the tip of the right foot and the tip of the left arm. Assume that the coordinate frames for each link are centred at the joints that connect them with their parent link, and assume that P_m and P_e define the tips of links m and e, respectively, in their local coordinate frame.

4. Transformations as a change of basis



$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_w = \begin{bmatrix} -1 & -1 & 0 & 5 \\ 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{obj}$$

   (a) (2 points) Given the above transformation matrix, sketch the origin and basis vectors of the resulting object coordinate frame.

   (b) (1 point) Sketch a house that is defined by the following points in the object coordinate frame: $P_1(0,0), P_2(2,0), P_3(2,2), P_4(1,3), P_5(0,2)$.

   (c) (2 points) Give a sequence of `translate(x,y,z)`, `rotate(z,theta)`, and `scale(a,b,c)` transformations that would result in the transformation matrix given above.

5. (4 points) Viewing Transformations
   Determine the viewing transformation, M_{view}, that takes points from WCS (world coordinates) to VCS (viewing or camera coordinates) for the following camera parameters: $P_{eye} = (-20, 30, -10), P_{ref} = (-20, 0, -10), V_{up} = (1, 0, 0)$. Do not bother with numerically inverting any matrices.

6. Rotation Matrices

   (a) (2 points) The columns of a rotation matrix have unit magnitude and they should all be orthogonal to each other, i.e., have a zero dot product. Show that the inverse of a rotation matrix is given by its transpose.

   (b) (6 points) In order for a rotation matrix to represent a rigid body rotation, there is one more constraint that it should satisfy, in addition to those listed above. Which of the following 4×4 matrices represent a valide rotation matrix? Why or why not? What is the extra constraint that rotation matrices should satisfy?

$$A = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \quad B = \begin{bmatrix} -1 & & & \\ & -1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} & 1 & & \\ -1 & & & \\ & 1 & & \\ & & & 1 \end{bmatrix} \quad D = \begin{bmatrix} & 1 & & \\ 1 & & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

7. (50 points) Coding with Transformations

   Your goal will be to create a model of a dog. Be as creative and ambitious as you like. The best work will be posted on the course web site in the Hall of Fame and shown in class. Use the arm-ball example as your starting template code (see the link from the CPSC 314 lectures web page).

   Requirements:

   - Use faces, cubes, pyramids, spheres, and cylinders to model your dog and its environment. Apply non-uniform scaling to achieve elongated shapes. If desired, you can also create new kinds of primitives using quadrilaterals by making calls to createGeometry(). can be modeled using your own object models or anything provided by glut.
   - Your dog should be modeled using a hierarchical scene graph structure. The head should be placed relative to the body, the ears relative to the head, and so forth.
   - The dog should have at least the following parts: body, head, front and hind legs consisting of at least two parts, ears, eyes, and tail.
   - The dog should be able to assume various poses. Pressing the number keys between 0 and 5 should each show a different pose. The number '0' should respond to the default rest pose.
   - The dog should be animatable. Pressing 'w' should result in (possibly repeating) walking in place, Optional: use other keys to initiate other motions that are animated as a function of time.

   Suggested Strategy:

   - First develop your dog model in a 'rest' pose, where all the joint angles are considered to be set to zero.
   - Use a statically-defined array to develop the various poses. The array would contain the numbers that specify the position of all the degrees of freedom for your dog. You might use a 2D array, with each row giving the degrees of freedom for each pose. The degrees of freedom might include: body height, angle of body with respect to the world, angle of head with respect to the body, leg joint angles, etc.
   - Then develop 'keyframes', which are poses that help define the beginning or end of a motion segment. A 2D array will work nicely for storing a set of keyframes. You might find it useful to read the keyframes in from a text file. Test the display of your poses by binding the number key presses (0–5) to display the relevant pose.
   - Using global variables will be convenient for working with the keyframes and pose data.
   - Develop smooth animations by linearly interpolating the degrees of freedom between successive key poses. In order to control the speed of the interpolation, you could specify a display time for each keyframe and then using the global time, `g_time`, to drive the interpolation.

To hand in:

- Hand in a README.txt file. This should include your name, student number, and user name. List any external resources (web pages, books, consultation with others) that you used. This assignment is to be completed individually. Comment on your implemented features and keybindings in your README file.
- To hand in, create a `cs314` subdirectory under your home directory, and create an `a1` subdirectory underneath that. Copy all the files that you wish to hand in to this directory. The assignment can be handed in using `handin cs314 a1` or Web handin: `https://my.cs.ubc.ca/docs/hand-in`.
- Demonstrate your code to a TA in the lab, or at another to-be-arranged time.