15-100 Moose Sections Homework 7

Start: Fri 2.13.9
Due: in class Mon 2.23.9
Goal: Lost in Space Again

Course Web Site:
www.andrew.cmu.edu/course/15-100mooseNsquirrel

Assignment:
You are returning to 3-D space but this time with a lot more "tools" in your tool kit, and unlike the astronauts on the last flight, your tool kit will not float away. You are going to draw one of your initials (HW 3) in 3-D space and then "fly" around it using the translate and rotate methods in the extended Processing API. Here is what you must do. In the Homeworkt.pde file that is distributed with this homework:

- Add a method named drawInitial() that has NO parameters
- Copy your 3-d initial drawing code from homework 3 and paste this into the drawInitial() method.
- As you coded your initial for HW3, all of the elements of your initial are in the positive quadrant with the (0,0,0) point being the upper left corner of the window. We want the (0,0,0) point to be in the center of the window and at the center of your initial. The initial translation to move the coordinate center to the middle of the window is already in the code – do not erase this line of code.
  You need to edit the x and y coordinate values (the z values "should" be ok) of the elements that make up your initial so that the (0,0,0) point of the coordinate place falls in the geographic center of your initial. Magic numbers are ok but strongly discouraged. We would like to be able to increase and decrease the window size and have your initial grow/shrink with the window size. See the first two pictures below:
- Add global variables needed to support translations and rotations
- Add a keyPressed method that changes the global variables so your position moves laterally and rotationally. The motion is the viewer's, not the initial's; e.g. [r] moves you to the right causing the initial to appear to move left. Here are the key sequences that you must use:

Lateral movement or translation:	Rotational Motion
move in ---- [i]	**left (y axis) -- [left arrow]**
move out --- [o]	**right(y axis) -- [right arrow]**
move right - [r]	**up (x axis) -- [up arrow]**
move left -- [l]	**down (x axis)- [down arrow]**
move up ---- [u]	**CCW (z axis) - [,]**
move down -- [d]	**CW (z axis) --- [.]**
reset to zero -[space]	

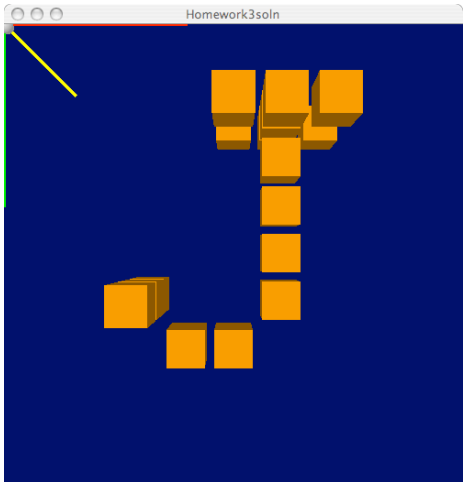
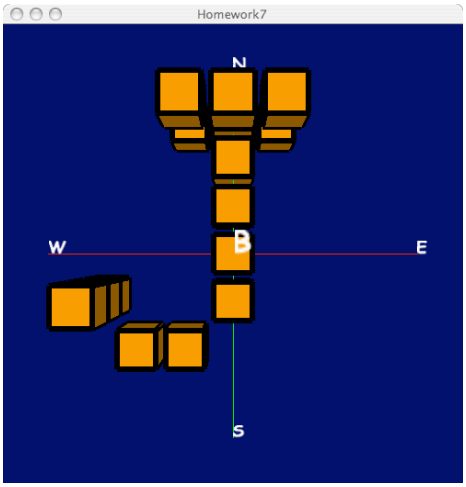
The code to "read" the arrows keys is somewhat different than reading the character keys. This is explained in the API and it will be covered in class.

Feel free to re-design your initial if you want to. Just remember that it must "exist" in more than one plane in each dimension. A flat (x, y) initial will not receive full credit.
Documentation

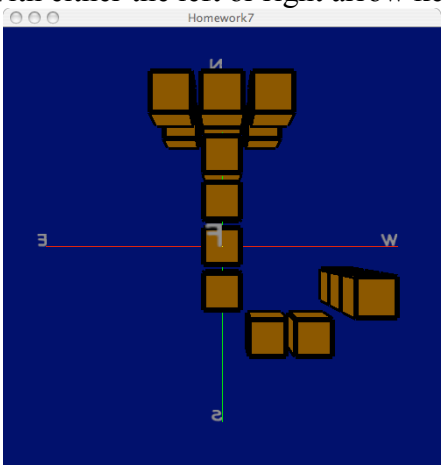
There is a comment at the top of the file that is called JavaDoc. This JavaDoc comment will display the movement controls in the applet window. Please do not edit or delete this comment.

Handin:
The usual – but (due to the test and a request from Lauren) shifted to Monday – there is no late handin. After Monday morning, the handin is closed and the Very Late opened.

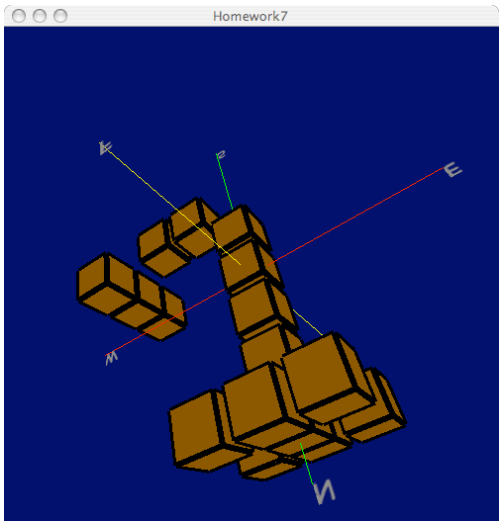
Sample:
NOTE: On the final read I just noticed that the 'B' and 'F' are wrong. They should be on the other ends of the coordinate line --- sigh…

Initial from HW3 with the (0,0,0) point marked and (x, y, z) axis lines drawn:	Initial modified for HW7 shifted to the center of the coordinate space with the axis lines drawn:
	

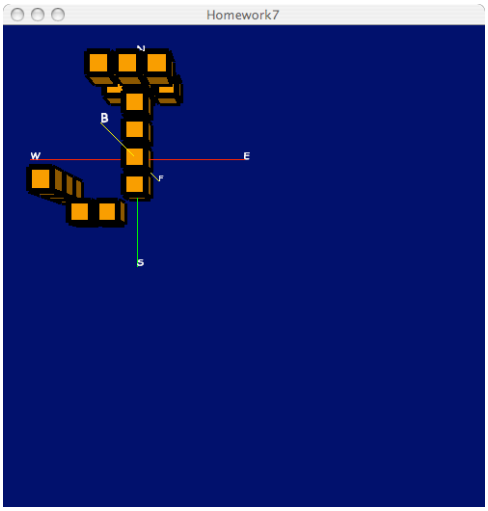
Here are some screen captures of Jim's flight (of fantasy…)
Rotating ournd the Y axis with either the left or right arrow keys:



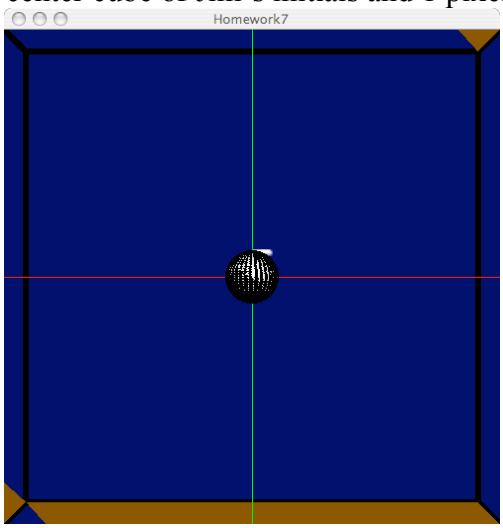
Multiple rotations:



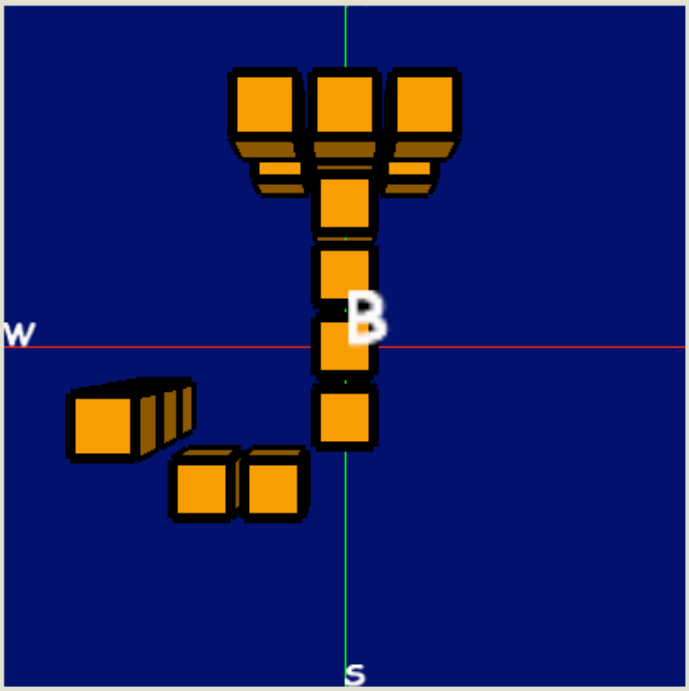
Lateral movement moving right [r], down [d], and out [o] so the initials appears to move up, left, and away



Lateral movement in with the [i] key to almost the geographic center of the window. The black sphere is inside the center cube of Jim's initials and 1 pixel in diameter.



This is the applet window. The JavaDoc comment is shown. The JavaDoc allows us to tell the user how to fly around.



Lateral Motion Commands

move in ---- [i]
move out --- [o]

move right - [r]
move left -- [l]

move up ---- [u]
move down -- [d]

Rotational Motion Commands

rotate left (y axis) --- [left arrow]
rotate right(y axis) --- [right arrow]

rotate up (x axis) --- [up arrow]
rotate down (x axis) - [down arrow]

rotate CCW (z axis) -- [,]
rotate CW (z axis) ---- [.]

reset to zero ---------- [space]

Source code: Homework7

Built with Processing