ETGG1803	Lab4: **Dot and Cross Product**	Points: **60**
Assigned: **2/6/2017**	Due: **2/17/2017**	

Tasks:

1. Get a working copy of our vector library (lab2 or lab3 solution)
2. Get a few spaceship sprites
    a. I used http://opengameart.org/content/2d-spaceship-sprites-with-engines
3. Modifications to vector.py
    a. (**4 points**) Make a **dot** *function* (should take two equally-sized VectorN's and return the dot-product). Raise an exception if the parameters are wrong.
    b. (**4 points**) Make a **cross** *function* (should only take 2 Vector3's and returns the cross-product). Raise an exception if the parameters are wrong.
    c. (**4 points**) In the Vector2 class, create a **perpendicular property** which returns a new Vector2 which is perpendicular to the original. Use your cross function to generate this (you'll probably need to temporarily create a 3d equivalent vector and cross it with (0,0,1)) .
4. Make a new utility.py module. We'll gradually add some "handy" functions here over the semester.
    a. (**6 points**) draw_arrow. Take a start position and end position (any dimension) and color paremeters as well as some optional parameters to control the size of the arrow head and line thickness.
5. Main program
    a. Create a ship class. I'll leave the design up to you, but include as much functionality as possible (as opposed to putting it directly in the main program).
    b. (**6 points**) Good design of the ship class
    c. (**5 points**) Create a list of ships, and draw all of them
    d. (**15 points**) Have a way to toggle which ship is active. For the active ship, draw its local axes, an arrow (use the new function) towards the closest ship *and* a grid aligned with the local axes.
    e. (**6 points**) Be able to rotate and move the active ship. Pressing WASD should trigger a fixed-size movement, even if the key is released. Make sure the grid and axes rotate appropriately
    f. (**10 points**) Be able to fire a rocket from the active ship. This rocket should move at a constant speed and turn towards the target (the closest ship when it was fired). Use the cross product technique for this.
6. Bonus features (**up to 20 points**) depending on complexity, difficulty, and mathey-ness.
    a. Make a game out of it (two teams, turn-based). More points for an AI
    b. (insert your idea here)
7. Here's a video of my solution (I've got a few bonus features included): **https://youtu.be/friuNYDyZzM**