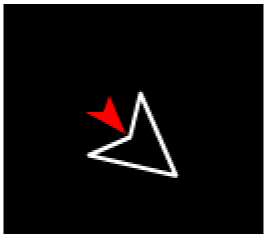
Game Project - Phase 2 (40 pts)

Let’s add some interaction to our game.

1. **(5 pts)** ship.rotateRight() and ship.rotateLeft() currently don’t do anything. Have it adjust the rotation angle and the direction the ship is pointing. The view must update with the current orientation (either via rotation of the ship’s geometry or through a modelview matrix) Have the ship look different if the thruster is on (when a user hits the spacebar). For example, give it some flames in the back



Now we add animation. Consider the animate() function inside main.cpp. This function will call the ship’s update function 30 times a second. If, for example, the position of the ship is moved incrementally, then it will appear to move smoothly as the framebuffer is drawn. Now, let’s recall some simple physics:

$F=ma$
$a = dv/dt$
$v = dp/dt$
(a = acceleration, v = velocity, t = time, F = force, p = position, m = mass = 1)

3. **(10 pts)** For every timestep update:

```
If the thruster is on:
    a = F
    v  = old_velocity + a*dt
p  = old_position + v*dt
```

   Have these updates affect the draw: If we consider p is the center of our ship and that we drew the ship centered at (0,0) in our world coordinates, then updating the draw position is adding a translation to the ship’s points either directly or through a modelview matrix.

4. **(5 pts)** You should clamp the acceleration and velocity to some maximum value and dampen the velocity slightly (0.98) at each update.
5. **(20 pts)** Finally add animation to your asteroid:
   5.1. Given it constant velocity at a random direction
   5.2. Have it rotate slowly over time