

# Jet Model

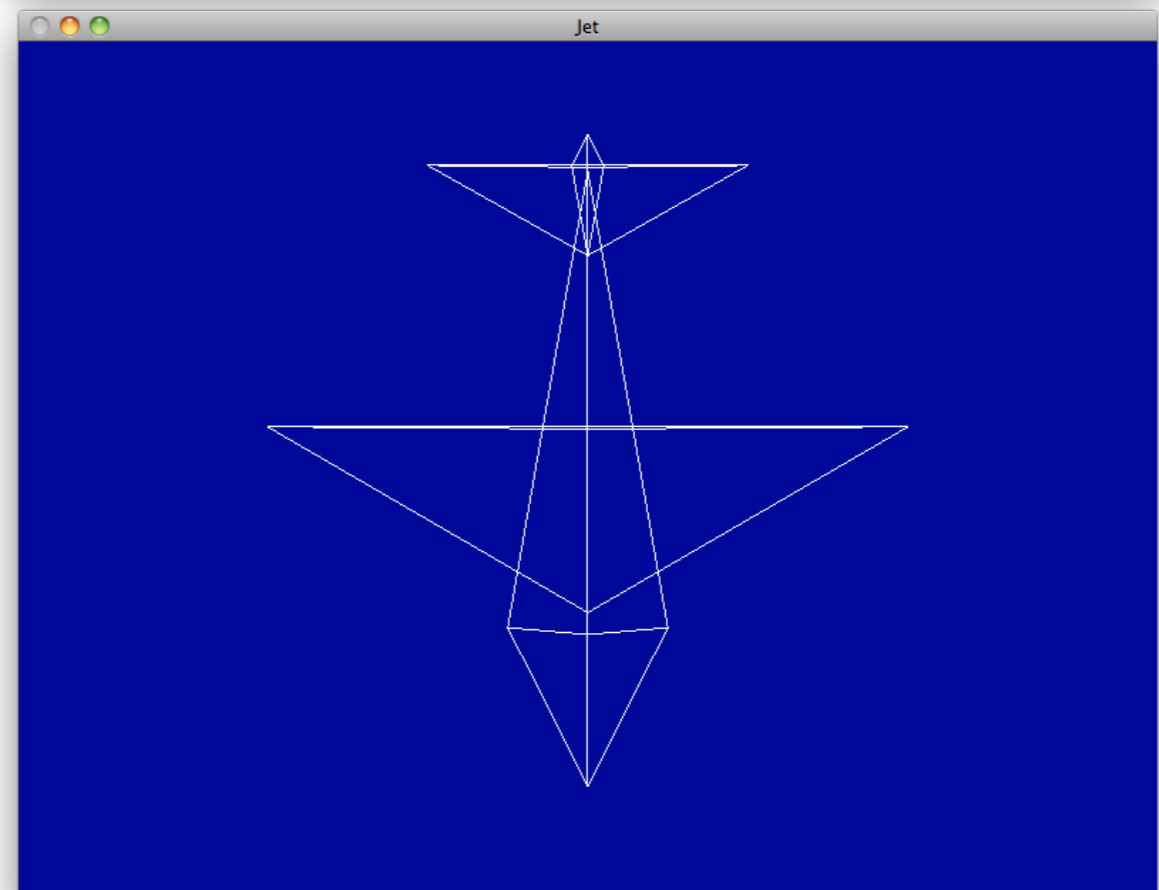
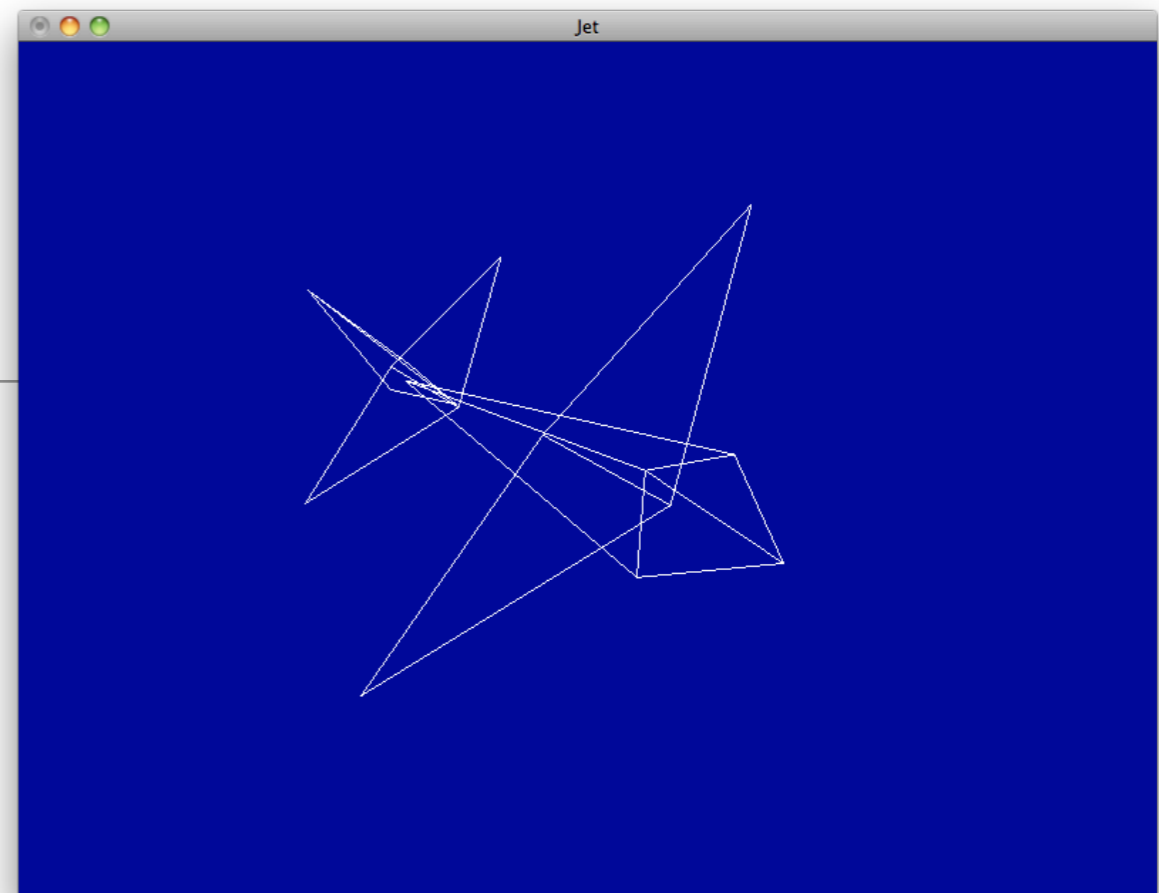
---

OpenGL

# Jet Model

---

- Load model, and render
- Use the third person camera to view the geometry from different angles



# Model File

- Triangle based geometry
- Anti Clockwise Winding

```
Vector3 noseCone[][3] =
{ { Vector3 ( 0.0, 0.0, 6.0),
  Vector3 ( -1.5, 0.0, 3.0),
  Vector3 ( 1.5, 0.0, 3.0) },
  { Vector3 ( 1.5, 0.0, 3.0),
  Vector3 ( 0.0, 1.5, 3.0),
  Vector3 ( 0.0, 0.0, 6.0) },
  { Vector3 ( 0.0, 0.0, 6.0),
  Vector3 ( 0.0, 1.5, 3.0),
  Vector3 ( -1.5, 0.0, 3.0) }
};

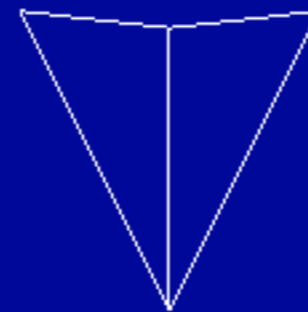
Vector3 body[][3] =
{ { Vector3 ( -1.5, 0.0, 3.0),
  Vector3 ( 0.0, 1.5, 3.0),
  Vector3 ( 0.0, 0.0, -5.6) },
  { Vector3 ( 0.0, 0.0, -5.6),
  Vector3 ( 0.0, 1.5, 3.0),
  Vector3 ( 1.5, 0.0, 3.0) },
  { Vector3 ( 1.5, 0.0, 3.0),
  Vector3 ( -1.5, 0.0, 3.0),
  Vector3 ( 0.0, 0.0, -5.6) }
};

Vector3 wings[][3] =
{ { Vector3 ( 0.0, .2, 2.7),
  Vector3 ( -6.0, .2, -.8),
  Vector3 ( 6.0, .2, -.8) },
  { Vector3 ( 6.0, .2, -.8),
  Vector3 ( 0.0, .7, -.8),
  Vector3 ( 0.0, .2, 2.7) },
  { Vector3 ( 6.0, .2, -.8),
  Vector3 ( -6.0, .2, -.8),
  Vector3 ( 0.0, .7, -.8) },
  { Vector3 ( 0.0, .2, 2.7),
  Vector3 ( 0.0, .7, -.8),
  Vector3 ( -6.0, .2, -.8) }
};
```

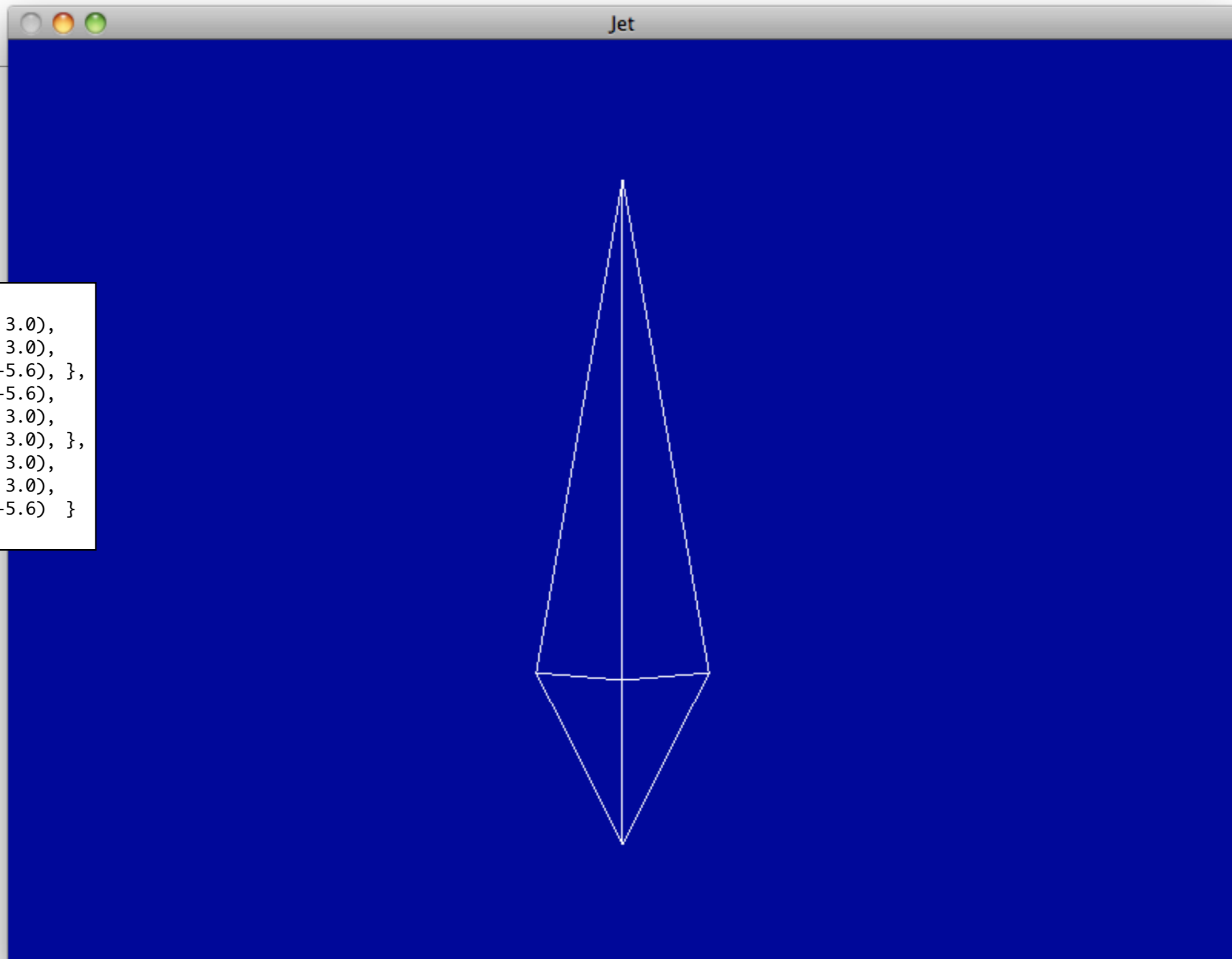
```
Vector3 tail[][3] =
{ { Vector3 ( -3.0, -.05, -5.7),
  Vector3 ( 3.0, -.05, -5.7),
  Vector3 ( 0.0, -.05, -4.0) },
  { Vector3 ( 0.0, -.05, -4.0),
  Vector3 ( 3.0, -.05, -5.7),
  Vector3 ( 0.0, .40, -5.7) },
  { Vector3 ( 0.0, .40, -5.7),
  Vector3 ( -3.0, -.05, -5.7),
  Vector3 ( 0.0, -.05, -4.0) },
  { Vector3 ( 3.0, -.05, -5.7),
  Vector3 ( -3.0, -.05, -5.7),
  Vector3 ( 0.0, .40, -5.7) },
  { Vector3 ( 0.0, .05, -4.0),
  Vector3 ( .3, .05, -5.7),
  Vector3 ( 0.0, 2.50, -6.5) },
  { Vector3 ( 0.0, 2.50, -6.5),
  Vector3 ( -.3, .05, -5.7),
  Vector3 ( 0.0, .05, -4.0) },
  { Vector3 ( .3, .05, -5.7),
  Vector3 ( -.3, .05, -5.7),
  Vector3 ( 0.0, 2.5, -6.5) }
};
```

# Nose Cone

```
Vector3 noseCone[][3] =  
{ { Vector3 ( 0.0, 0.0, 6.0),  
  Vector3 ( -1.5, 0.0, 3.0),  
  Vector3 ( 1.5, 0.0, 3.0) },  
  { Vector3 ( 1.5, 0.0, 3.0),  
  Vector3 ( 0.0, 1.5, 3.0),  
  Vector3 ( 0.0, 0.0, 6.0) },  
  { Vector3 ( 0.0, 0.0, 6.0),  
  Vector3 ( 0.0, 1.5, 3.0),  
  Vector3 ( -1.5, 0.0, 3.0) }  
};
```



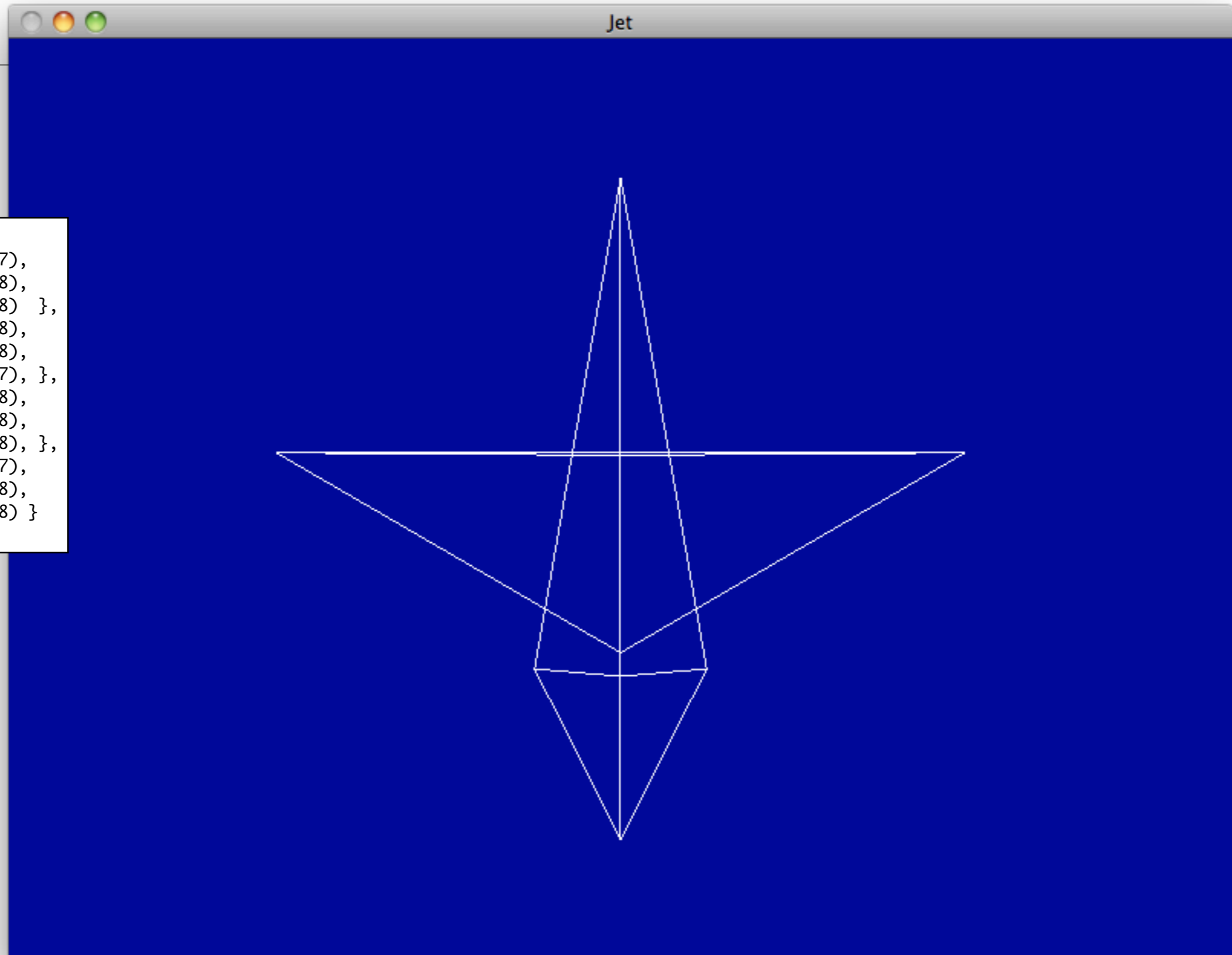
# Body



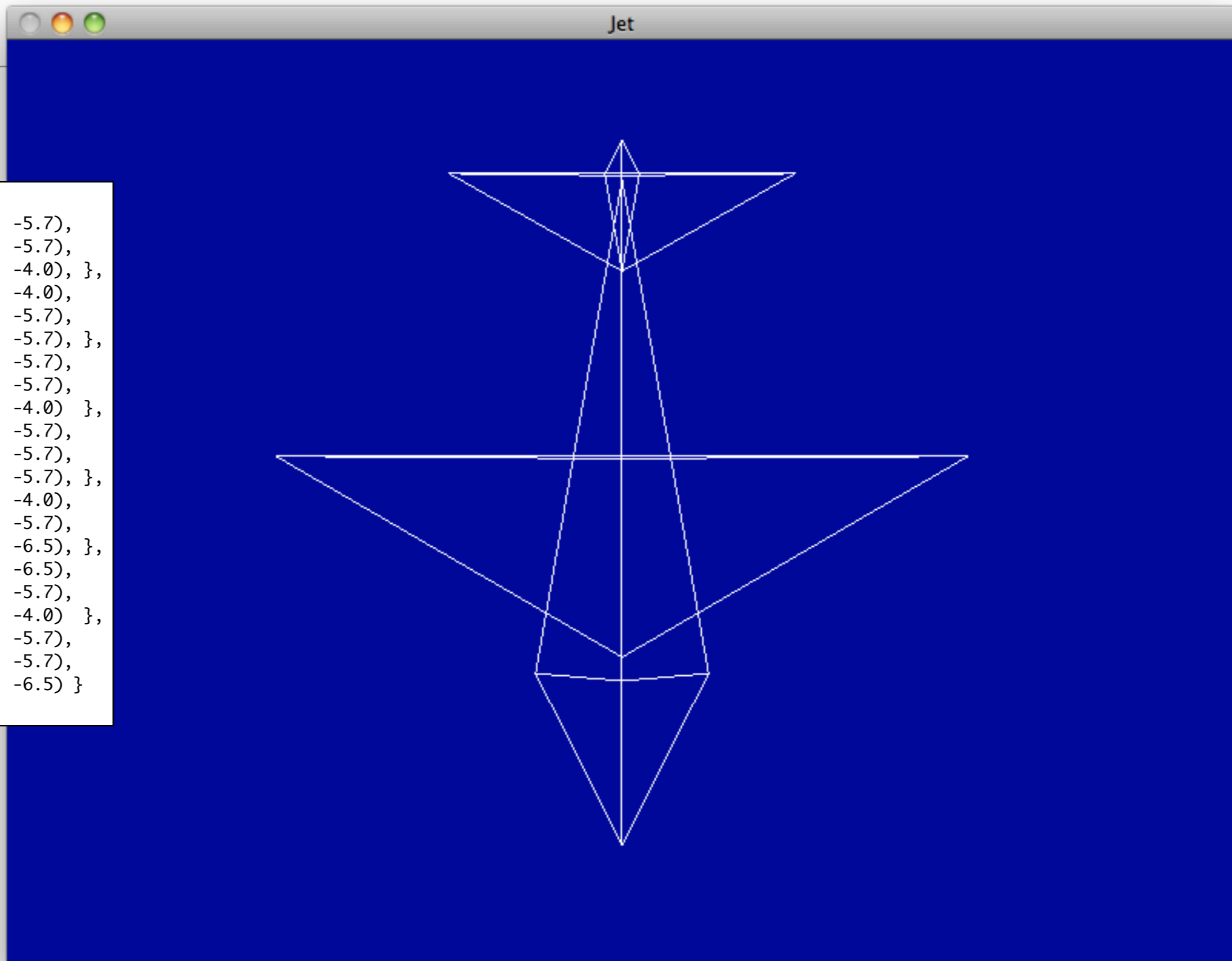
```
Vector3 body[][3] =  
{ { Vector3 ( -1.5, 0.0, 3.0),  
  Vector3 ( 0.0, 1.5, 3.0),  
  Vector3 ( 0.0, 0.0, -5.6), },  
  { Vector3 ( 0.0, 0.0, -5.6),  
  Vector3 ( 0.0, 1.5, 3.0),  
  Vector3 ( 1.5, 0.0, 3.0), },  
  { Vector3 ( 1.5, 0.0, 3.0),  
  Vector3 ( -1.5, 0.0, 3.0),  
  Vector3 ( 0.0, 0.0, -5.6) }  
};
```

# Wings

```
Vector3 wings[][3] =  
{ { Vector3 ( 0.0, .2, 2.7),  
  Vector3 ( -6.0, .2, -.8),  
  Vector3 ( 6.0, .2, -.8) },  
  { Vector3 ( 6.0, .2, -.8),  
  Vector3 ( 0.0, .7, -.8),  
  Vector3 ( 0.0, .2, 2.7), },  
  { Vector3 ( 6.0, .2, -.8),  
  Vector3 ( -6.0, .2, -.8),  
  Vector3 ( 0.0, .7, -.8), },  
  { Vector3 ( 0.0, .2, 2.7),  
  Vector3 ( 0.0, .7, -.8),  
  Vector3 ( -6.0, .2, -.8) }  
};
```



# Tail Parts



```
Vector3 tail[][3] =  
{ { Vector3 (-3.0, -.05, -5.7),  
  Vector3 ( 3.0, -.05, -5.7),  
  Vector3 ( 0.0, -.05, -4.0) },  
  { Vector3 ( 0.0, -.05, -4.0),  
    Vector3 ( 3.0, -.05, -5.7),  
    Vector3 ( 0.0, .40, -5.7) },  
  { Vector3 ( 0.0, .40, -5.7),  
    Vector3 (-3.0, -.05, -5.7),  
    Vector3 ( 0.0, -.05, -4.0) },  
  { Vector3 ( 3.0, -.05, -5.7),  
    Vector3 (-3.0, -.05, -5.7),  
    Vector3 ( 0.0, .40, -5.7) },  
  { Vector3 ( 0.0, .05, -4.0),  
    Vector3 ( .3, .05, -5.7),  
    Vector3 ( 0.0, 2.50, -6.5) },  
  { Vector3 ( 0.0, 2.50, -6.5),  
    Vector3 ( -.3, .05, -5.7),  
    Vector3 ( 0.0, .05, -4.0) },  
  { Vector3 ( .3, .05, -5.7),  
    Vector3 ( -.3, .05, -5.7),  
    Vector3 ( 0.0, 2.5, -6.5) }  
};
```

# Loading the Colour Model

- Triangle has a colour attribute c.
- Read from the model file
- Rendered before vertices

```
struct JetPlane : public Actor
{
    JetPlane();
    void render();
};
```

```
void render (Vector3 vectors[][3], int size)
{
    for (int i=0; i<size; i++)
    {
        glBegin(GL_TRIANGLES);
        vectors[i][0].render();
        vectors[i][1].render();
        vectors[i][2].render();
        glEnd();
    }
}

JetPlane::JetPlane()
{}

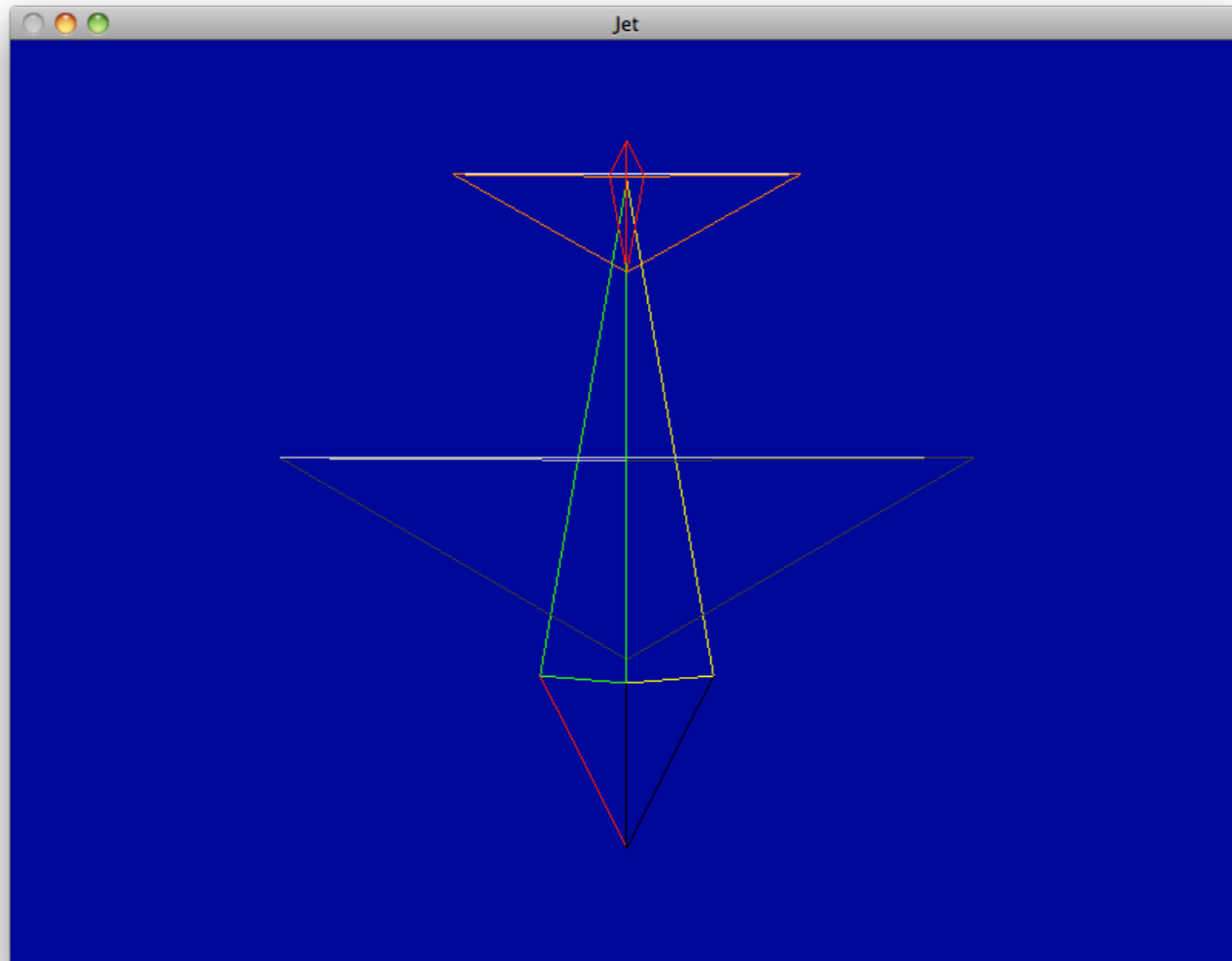
void JetPlane::render()
{
    glShadeModel(GL_SMOOTH);
    glPolygonMode(GL_FRONT, GL_FILL);

    Color::Yellow.render();
    ::render(noseCone, 3);
    Color::Red.render();
    ::render(body, 3);
    Color::Green.render();
    ::render(wings, 4);
    Color::Cyan.render();
    ::render(tail, 7);

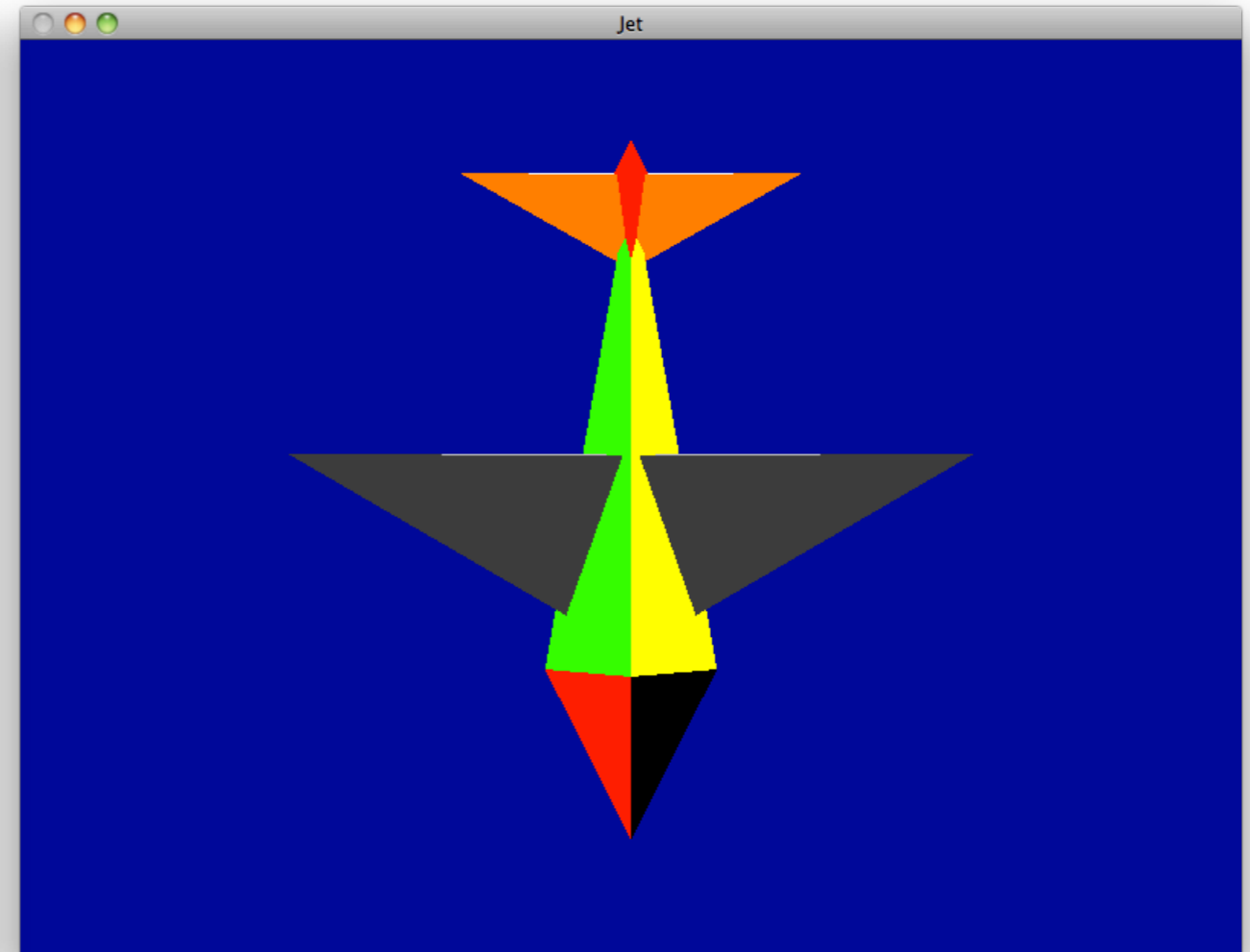
    Color::White.render();
    glPolygonMode(GL_FRONT, GL_LINE);
}
```



```
glEnable (GL_DEPTH_TEST);  
glEnable (GL_CULL_FACE);  
glFrontFace (GL_CCW);  
glClearColor (0.0f, 0.0f, 0.6f, 1.0f);
```



```
glPolygonMode (GL_FRONT, GL_LINE);
```



```
glPolygonMode (GL_FRONT, GL_FILL);
```