1. Assume you are creating the rocket object to the left. For just the shape, indicate the polygons you will create and their vertices in the correct order (to right).

2. Suppose I want to also be able to scale my polygon by a uniform scale factor s. Indicate the source to destination transformation include any scaling, rotation, and translation.

\[ x' = s (x \cos(a) - y \sin(a)) + p_x \]
\[ y' = s (x \sin(a) + y \cos(a)) + p_y \]

3. What variables would I need to add to my hexagon class to support moving with acceleration, wind drag, rotation and rotational acceleration? Indicate the type of each variable (Vector2 or float).

```java
private Vector2 p;
private Vector2 v;
private Vector2 a;
float kp;
float kr;
float r;
float rv;
float ra;
```
4. a) Provide an Advance routine to update these variables (to right).

```csharp
void Advance(float dt)
{
    Vector2 ap = a + v * -kp;
    v += ap * dt;
    p += v * dt;
    float rap = ra + rv * -kr;
    rv += rap * dt;
    r += rv * dt;
}
```

b) What modifications to the advance routine would be needed to support gravity and a
floor at y=1? (to right)

```csharp
void Advance(float dt)
{
    Vector2 ag = a;
    if(p.Y > 1)
    {
        ag += Vector2(0, -9.8);
    }

    Vector2 ap = ag + v * -kp;
    v += ap * dt;
    p += v * dt;
    float rap = ra + rv * -kr;
    rv += rap * dt;
    r += rv * dt;

    if(p.Y < 1)
    {
        p.Y = 1;
    }
}
```

5. Provide right arrow key up and key down code to move the hexagon.

Depends on what you want it to do.

```csharp
    case Keys.Right: // Key down
        hexagon.V = new Vector2(5, 0);
        break;
```
case Keys.Right: // Key up
    hexagon.V = new Vector2(0, 0);
    break;

or maybe:

case Keys.Right: // Key down
    hexagon.A = new Vector2(5, 0);
    break;

case Keys.Right: // Key up
    hexagon.A = new Vector2(0, 0);
    break;

6. Provide up arrow key up and key down code to jump.

Just need key down:

    case Keys.Up:
        hexagon.Jump();
        break;

public void Jump()
{
    if(p.Y > 1)
    {
        v = new Vector2(0, 5);
    }