Coordinate Mapping Fix for 3D Visualization

Problem

The 3D visualization was displaying the room rotated incorrectly because matplotlib’s 3D axes have **Z as the vertical axis** by default, but our data stores **Y as height (vertical)**.

Solution

Implemented Y/Z coordinate swapping when plotting to matplotlib’s 3D axes.

Data Storage Format

Positions are stored as `[x, y, z]` where:
- **x (index 0)**: Width (0-30 ft)
- **y (index 1)**: Height (0-20 ft)
- **z (index 2)**: Length (0-75 ft)

Matplotlib Axis Mapping

Matplotlib 3D has Z as vertical, so we swap Y and Z when plotting:
- **matplotlib X-axis** ← data[0] (Width, 0-30 ft, horizontal)
- **matplotlib Y-axis** ← data[2] (Length, 0-75 ft, depth)
- **matplotlib Z-axis** ← data[1] (Height, 0-20 ft, **VERTICAL**)

Implementation Details

1. Axis Setup (`_setup_axes`)

```
# Set axis limits with Y/Z swapped
self.ax.set_xlim(0, ROOM_WIDTH)   # X: Width (0-30 ft)
self.ax.set_ylim(0, ROOM_LENGTH)  # Y: Length (0-75 ft, depth)
self.ax.set_zlim(0, ROOM_HEIGHT)  # Z: Height (0-20 ft, VERTICAL)

# Set labels
self.ax.set_xlabel('Width (ft)')
self.ax.set_ylabel('Length (ft)')   # Depth
self.ax.set_zlabel('Height (ft)')   # Vertical
```

2. Room Drawing (`_draw_room`)

```
def to_mpl(x, y, z):
    """Convert data [x, y, z] to matplotlib [X, Y, Z] = [x, z, y]"""
    return (x, z, y)

# Use to_mpl() to convert all corner coordinates
```

3. Fan Drawing (`_draw_fan`)

```
# Convert position from data to matplotlib coords
mpl_x = pos[0]  # Width → X
mpl_y = pos[2]  # Length → Y (depth)
mpl_z = pos[1]  # Height → Z (vertical)

# Draw fan circle in XZ plane (horizontal-vertical)
```

4. Sensor Drawing (`_draw_sensors`)

```
# Convert each sensor position
mpl_x = pos[0]  # Width → X
mpl_y = pos[2]  # Length → Y (depth)
mpl_z = pos[1]  # Height → Z (vertical)
```

5. Particle Drawing (`_draw_particles`)

```
# Swap Y/Z for all particles at once
mpl_xs = particles[:, 0]  # Width → X
mpl_ys = particles[:, 2]  # Length → Y (depth)
mpl_zs = particles[:, 1]  # Height → Z (VERTICAL)

self.ax.scatter(mpl_xs, mpl_ys, mpl_zs, ...)
```

Result

✅ Room displays upright with proper orientation
✅ Height axis is vertical (Z in matplotlib, 0-20 ft)
✅ Width axis is horizontal (X, 0-30 ft)
✅ Length axis is depth (Y in matplotlib, 0-75 ft)
✅ Fan appears on back wall at correct height
✅ Smoke rises vertically along the Z-axis
✅ Sensors display at correct heights

Files Modified

- `/home/ubuntu/smoke_simulation_tool/visualization/renderer_3d.py`
- Updated `_setup_axes()` method
- Updated `_draw_room()` method with `to_mpl()` helper
- Updated `_draw_fan()` method
- Updated `_draw_sensors()` method
- Updated `_draw_particles()` method

Testing

Run the application and verify:
1. Room appears upright (not rotated on side)
2. Axis labels show: Width (X), Length (Y), Height (Z)
3. Axis ranges show: 0-30, 0-75, 0-20 respectively

4. Smoke particles rise vertically
5. Fan is on back wall at correct height (15 ft)

Key Insight

Matplotlib’s 3D coordinate system naturally has Z as vertical, which differs from many physics/CAD systems that use Y as vertical. The fix maps our Y-up data to matplotlib’s Z-up system through coordinate swapping: `(x, y, z) → (x, z, y)`.