Function Documentation:`generate_points`

1 Description

The `generate_points` function generates a list of `N` points evenly spaced between two given points, `last_point` and `point`. The generated points are rounded to the nearest integer coordinates.

2 Function Definition

```python
def generate_points(last_point, point, N):
    """Generate N points between last_point and point."""
    x1, y1 = last_point
    x2, y2 = point

    points = []
    for i in range(1, N + 1):
        t = i / (N + 1)
        x = x1 + t * (x2 - x1)
        y = y1 + t * (y2 - y1)
        points.append((int(round(x)), int(round(y))))

    return points
```

3 Function Explanation

3.1 Step-by-Step Breakdown

Function 1: Calculate Interpolation Factor

Calculate the interpolation factor `t` for each point.

```python
t = i / (N + 1)
```

Explanation: The interpolation factor `t` is computed for each point to be generated. It ensures that points are evenly spaced between `last_point` and `point`.

Function 2: Calculate Coordinates

Calculate the coordinates of each generated point.

```
x = x1 + t * (x2 - x1)
y = y1 + t * (y2 - y1)
```

Explanation: The coordinates of each point are calculated by interpolating between the `last_point` and `point` using the interpolation factor `t`. The results are rounded to the nearest integer.

Function 3: Round and Append

Round the coordinates to the nearest integer and append to the list of points.

```
points.append((int(round(x)), int(round(y))))
```

Explanation: The calculated coordinates are rounded to the nearest integers to ensure they represent pixel locations or discrete points. These coordinates are then added to the list of generated points.

4 Conclusion

The `generate_points` function efficiently generates a specified number of evenly spaced points between two given coordinates. This function is useful for tasks requiring interpolation between points in various applications, including graphics and data analysis.