

Ukázkové Python skripty: Depth Mesh

2

Zde bude popsáno několik příkladů skriptů, které upravují trojúhelníkovou síť vzniklou z hloubkového obrazu scény v aplikaci Depth Mesh.

Posunutí doprava

Tento skript posune všechny vrcholy trojúhelníkové sítě o jeden metr doprava - tedy o jeden metr v kladném směru x-ové osy.

```
1 for i in range(0, len(points), 3):
2     points[i] = points[i] + 1
3 return [points, uvs, faces]
```

2.0.1 Zvětšení

Tento skript zdvojnásobí vzdálenost mezi všemi vrcholy trojúhelníkové sítě. Neboli celou trojúhelníkovou síť zvětší na dvojnásobek jejího aktuálního rozměru.

```
1 for i in range(0, len(points)):
2     points[i] = points[i] * 2
3 return [points, uvs, faces]
```

2.0.1.1 Posunutí textury

Tento skript slouží k posunutí textury po modelu. Texturovací souřadnice jsou souřadnice do obrázku, který se má na model namapovat. Jedná se o souřadnici u ve vodorovném směru a v ve směru svislém. Hodnoty souřadnic jsou normalizované - tedy mohou mít hodnoty jen v intervalu $<0, 1>$. V tomto skriptu se ke každé texturovací souřadnici přičte 0.1 - výsledkem bude posun obrázku po vytvořeném modelu scény směrem doleva a nahoru.

```
1 for i in range(0, len(uvs)):
2     uvs[i] = (uvs[i] + 0.1)%1
3 return [points, uvs, faces]
```

2.0.2 Odstranění vrcholu

Tento skript ukazuje, jak je možné odstranit jeden vrchol z trojúhelníkové sítě. Index odstraňovaného vrcholu je daný proměnnou *tIndex*. Pro odstranění vrcholu je třeba odstranit jeho souřadnice z pole *points*. Dále je třeba odstranit jeho uv souřadnice z pole *uvs*. Důležité je korektně odstranit všechny trojúhelníky z pole *faces*, ve kterých se tento vrchol nacházel. Trojúhelník je udán třemi čísly reprezentujícími pořadová čísla vrcholů, které tento trojúhelník tvoří. Při odstranění vrcholu dojde ke zkrácení pole *points*. Všechny vrcholy, které se nacházejí v poli *points* za odstraněným vrcholem, nyní leží na nižším indexu než předtím, a je třeba v poli *faces* o jedna snížit jejich pořadová čísla. Proto je třeba upravit čísla vrcholů uložené v poli *faces*.

```
1 tIndex = 10
2 newpoints = []
3 for i in range(0, len(points), 3):
4     x = points[i+0]
5     y = points[i+1]
6     z = points[i+2]
7     if ((i / 3) != tIndex):
8         newpoints.append(x)
9         newpoints.append(y)
10        newpoints.append(z)
11
12 newuvs = []
13 for i in range(0, len(uvs), 2):
14     u = uvs[i+0]
15     v = uvs[i+1]
16     if ((i / 2) != tIndex):
17         newuvs.append(u)
18         newuvs.append(v)
19
20 newF = []
21 for i in range(0, len(faces), 3):
22     f1 = faces[i+0]
23     f2 = faces[i+1]
24     f3 = faces[i+2]
25
26     if (f1 != tIndex and f2 != tIndex
27         and f3 != tIndex):
28         if (f1 > tIndex):
29             f1 = f1 - 1
30         if (f2 > tIndex):
31             f2 = f2 - 1
32         if (f3 > tIndex):
33             f3 = f3 - 1
34
35     newF.append(f1)
```

```
36     newF.append(f2)
37     newF.append(f3)
38
39 return [newpoints, newuvs, newF]
```

