How to make JSON scene

First you must make objects array, then lights array and cameras array. Ex.:

```
{
      “objects”:[ {object1}, {object2}, … , {objectN} ],
      “lights”:[ {light1}, {light2}, … , {lightN} ],
      “cameras”:[ {camera1}, {camera2}, … , {cameraN} ]
}
```

How to make a simple object:

```
{
      “type”: “sphere”,
      “position”: [0, 1, -10],
      “color”: [0.1, 0.7, 0.2],
      “specular”: 1000
}
```

Instructions for all types will be lower.

How to make a simple light:

```
{
      “type”: “lamp”,
      “position”: [2, 2, 0],
      “color”: [0.1, 0.1, 0.3],
      “intensity”: 0.5
}
```

Instructions for all types will be lower.

How to make a simple camera:

```
{
      “type”: “common”,
      “rotation”: [0, 0, 0],
      “position”: [0, 0, 0],
      “field_of_view”: 50
}
```

Instructions for all types will be lower.

Objects

Type: Sphere

Main properties:

"type": "sphere"

"position" or "pos1": vector (x, y, z) in float any

"radius": 0 < val in float

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: Plane

Main properties:

"type": "plane"

"position" or "pos1": vector (x, y, z) in float any

"direction": vector (x, y, z) - planes normal

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: Cylinder

Main properties:

"type": "cylinder"

"position" or "pos1": vector (x, y, z) in float any

"radius": 0 < val in float

"direction": vector (x, y, z) - cylinder direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

"max": 0 < val upper limit.

Type: Conus

Main properties:

"type": "conus"

"position" or "pos1": vector (x, y, z) in float any

"direction": vector (x, y, z) - conus direction

"angle": angle for lower and upper part of conus

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

"min": val < 0 lower limit.

"max": 0 < val upper limit.

Type: **Disk**

Main properties:

"type": "disk"

"position" or "pos1": vector (x, y, z) in float any

"radius": 0 < val in float

"direction": vector (x, y, z) - disk normal

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: **Triangle**

Main properties:

"type": "triangle"

"pos1": vector (x, y, z) in float any

"pos2": vector (x, y, z) in float any

"pos3": vector (x, y, z) in float any

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: Paraboloid

Main properties:

"type": "paraboloid"

"position" or "pos1": vector (x, y, z) in float any

"radius": 0 < val in float

"direction": vector (x, y, z) - paraboloid direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

"max": 0 < val upper limit.

Type: Square

Main properties:

"type": "square"

"pos1": vector (x, y, z) in float any

"pos2": vector (x, y, z) in float any

"direction": vector (x, y, z) - conus direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: Capsula

Main properties:

"type": "capsula"

"position" or "pos1": vector (x, y, z) in float any

"radius": 0 < val in float

"direction": vector (x, y, z) - cylinder direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

"max": 0 < val upper limit.

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: Barbell

Main properties:

"type": "barbell"

"position" or "pos1": vector (x, y, z) in float any

"direction": vector (x, y, z) - conus direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"specular": int value of specularity from 1 to max

"max": 0 < val upper limit.

"radius": 0 < val in float

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

Type: **Elipsoid**

Main properties:

"type": "elipsoid"

"position" or "pos1": vector (x, y, z) in float any

"radius": 0 < val in float

"direction": vector (x, y, z) - cylinder direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"size": val < "radius"

"specular": int value of specularity from 1 to max

Additional properties:

"reflect": 0 <= val <= 1 in float

"refract": 0 <= val <= 1 in float

"ior": 0.5 <= val <= 2.0 in float

"pattern": "chessboard" / "gradient" / "perlin" / "circle" / "brick" / "custom-1" / "custom-2" / "custom-3" / "custom-4"

"pattern_scale": 1 <= val <= max short

"tex_angle": 0 < val < 360 in degrees. Angle for textures

"min": val < 0 lower limit.

"max": 0 < val upper limit.

Lights

Type: Area

Main properties:

"type": "area"

"position": vector (x, y, z) in float any

"color": vector (r, g, b) in float each 0 <= color <= 1.

"direction": vector (x, y, z) - lights direction

"angle": angle of area light

"intensity": 0 <= val

Type: Lamp

Main properties:

"type": "lamp"

"position": vector (x, y, z) in float any

"color": vector (r, g, b) in float each 0 <= color <= 1.

"intensity": 0 <= val

Type: Ambient

Main properties:

"type": "ambient"

"color": vector (r, g, b) in float each 0 <= color <= 1.

"intensity": 0 <= val

Type: Parallel

Main properties:

"type": "parallel"

"position": vector (x, y, z) in float any

"direction": vector (x, y, z) - ligts direction

"color": vector (r, g, b) in float each 0 <= color <= 1.

"intensity": 0 <= val

Camera

Type: Common

Main properties:

"type": "common"

"position": vector (x, y, z) in float any

"direction": vector (x, y, z) - lights direction

"field_of_view": angle of view area