IN204

Yu WANG

March 2021

Algorithme 1 Ray tracing

```
Input  $std :: vector < Object* >$， $std :: vector < Source* >$
Output: $pixels[Width * Height]$
for $each\ pixel\ of\ the\ image$ do
    $ray \leftarrow \textsc{generateRay}(Camera, xPostion, yPosition)$
    for $i = 0$ to $number\ of\ objects$ do
        $intersections[i] \leftarrow \textsc{findIntersections}(ray, objects)$
    end for
    $indexClosest \leftarrow \textsc{findIndexClosest}(intersections)$
    if $indexClosest == -1$ then
        $pixel \leftarrow BLACK$
    else
        $pixel \leftarrow \textsc{getColor}(indexClosest, ray, intersections, objects, sources)$
    end if
end for
```

Algorithme 2 getColor

```
Input  $indexClosest, ray, intersections, objects, sources$
Output: $finalColor$
$objectColor \leftarrow \textsc{getColor}(objects[indexClosest])$
$finalColor \leftarrow 0.2 * objectColor$
for $i = 0$ to $number\ of\ sourcea$ do
    if $shadow == false$ then
        $sourceColor \leftarrow \textsc{getColor}(sources[i])$
        $finalColor \leftarrow finalColor + \textsc{modulate}(objectColor, sourceColor)$
        $reflect \leftarrow \textsc{getReflect}(objects[indexClosest])$
        if $0 < reflect <= 1$ then
            $specular \leftarrow \textsc{dot}(reflectionDirection, lightDirection)$
            if $specular > 0$ then
                $specular \leftarrow \textsc{pow}(specular, 10)$
                $finalColor \leftarrow finalColor + \textsc{mul}(sourceColor, specular * reflect)$
            end if
        end if
    end if
end for
```