# Ray tracing - extra

Rui Ying u1234364

## Recursive ray tracing

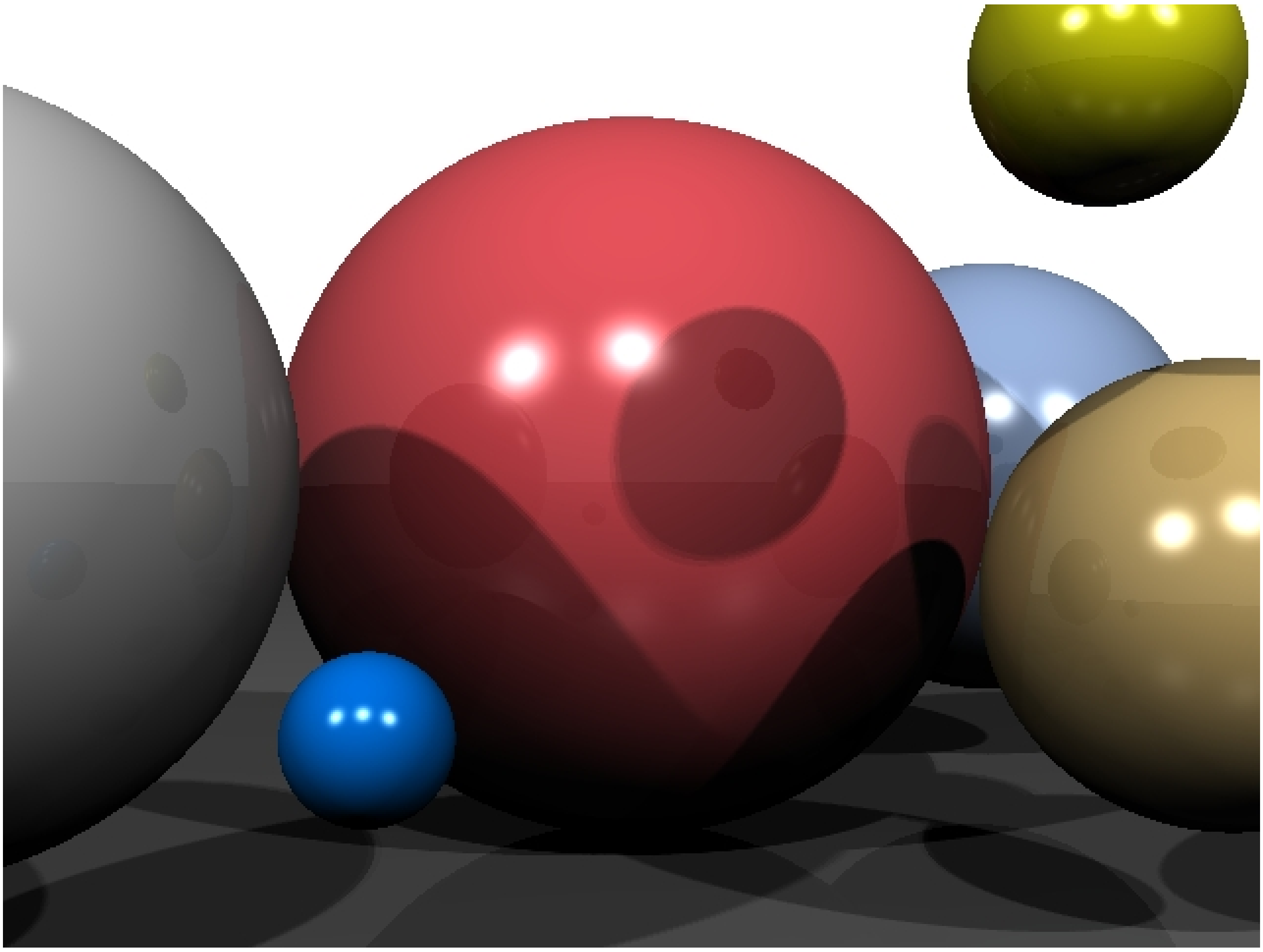
New `trace` function declaration

```
Vector3f trace(
    const Vector3f &rayOrigin,
    const Vector3f &rayDirection,
    const std::vector<Sphere> &spheres,
    int depth)
```

I use some parameter to ensure the primary ray has the most effect on pixel color:

```
pixelColor = 0.95* pixelColor + 0.05 * trace(hitPoint, R, spheres, depth);
```

I also add property `specular` to class `Sphere` so that we can see the difference. In the result, `the blue ball` is not specular and does not reflect other balls.



## Area light sources

I add 6 more close light sources to each original light souce so that they can be seen as three area lights.

```
std::vector<std::vector<Vector3f>> lightPositions = {
        {
                Vector3f(0.0, 60, 60),
                Vector3f(1.0, 60, 60),
                Vector3f(-1.0, 60, 60),
                Vector3f(0.0, 59, 60),
                Vector3f(0.0, 61, 60),
                Vector3f(0.0, 60, 59),
                Vector3f(0.0, 60, 61)
        },
        {
                Vector3f(-60.0, 60, 60),
                Vector3f(-59.0, 60, 60),
                Vector3f(-61.0, 60, 60),
                Vector3f(-60.0, 59, 60),
                Vector3f(-60.0, 61, 60),
                Vector3f(-60.0, 60, 59),
                Vector3f(-60.0, 60, 61)
        },
        {
                Vector3f(60.0, 60, 60),
                Vector3f(59.0, 60, 60),
                Vector3f(61.0, 60, 60),
                Vector3f(60.0, 59, 60),
                Vector3f(60.0, 61, 60),
                Vector3f(60.0, 60, 59),
                Vector3f(60.0, 60, 61)
        }
};
```

In `trace` function, I need to iterate through these new light sources as well. And be careful that colors are not divided by 3 now.

Shadows are softer because of multiple light sources.