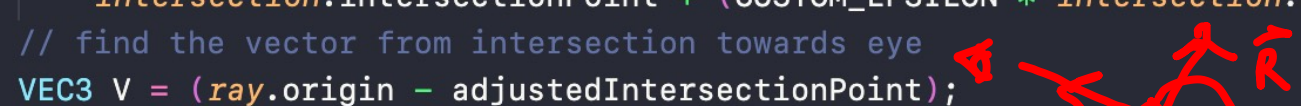
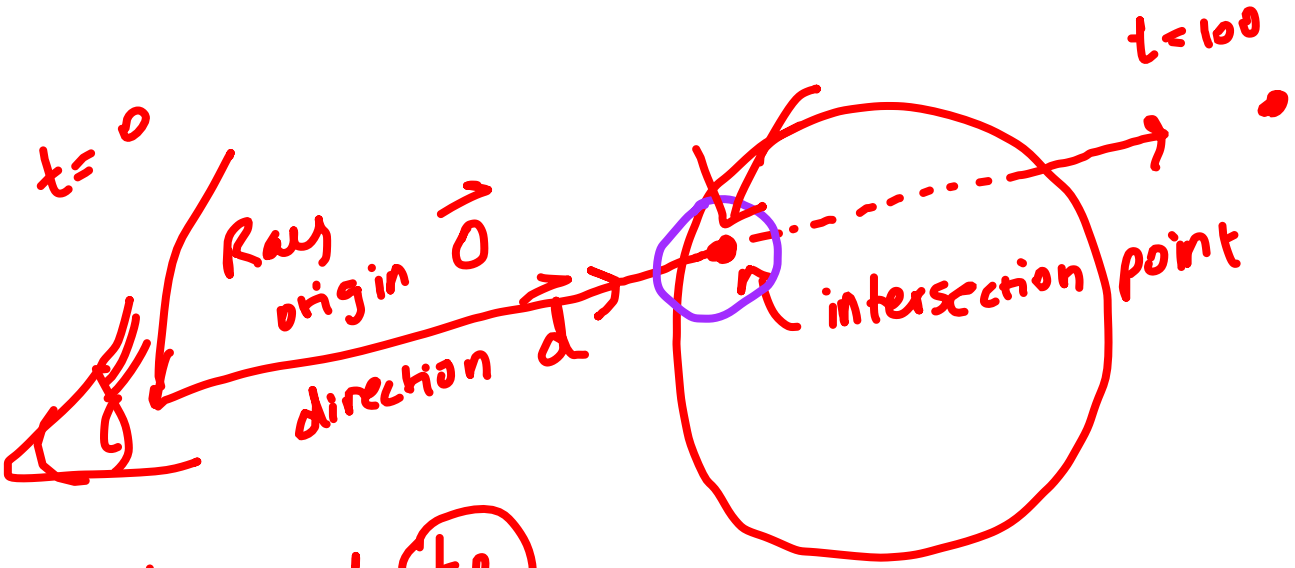
d

```
Ray createReflectionRay(IntersectResult intersection, Ray ray) {
    // adjust to avoid shadow acne problem
    VEC3 adjustedIntersectionPoint =
        intersection.intersectionPoint + (CUSTOM_EPSILON * intersection.normal);
    // find the vector from intersection towards eye
    VEC3 V = (ray.origin - adjustedIntersectionPoint);
    V /= V.norm();
    // get the reflection direction
    VEC3 R = -V + (2.0 * V.dot(intersection.normal) * intersection.normal);
    R /= R.norm();
    // generate ray
    Ray reflectionRay(adjustedIntersectionPoint, R);
    return reflectionRay;
}
```





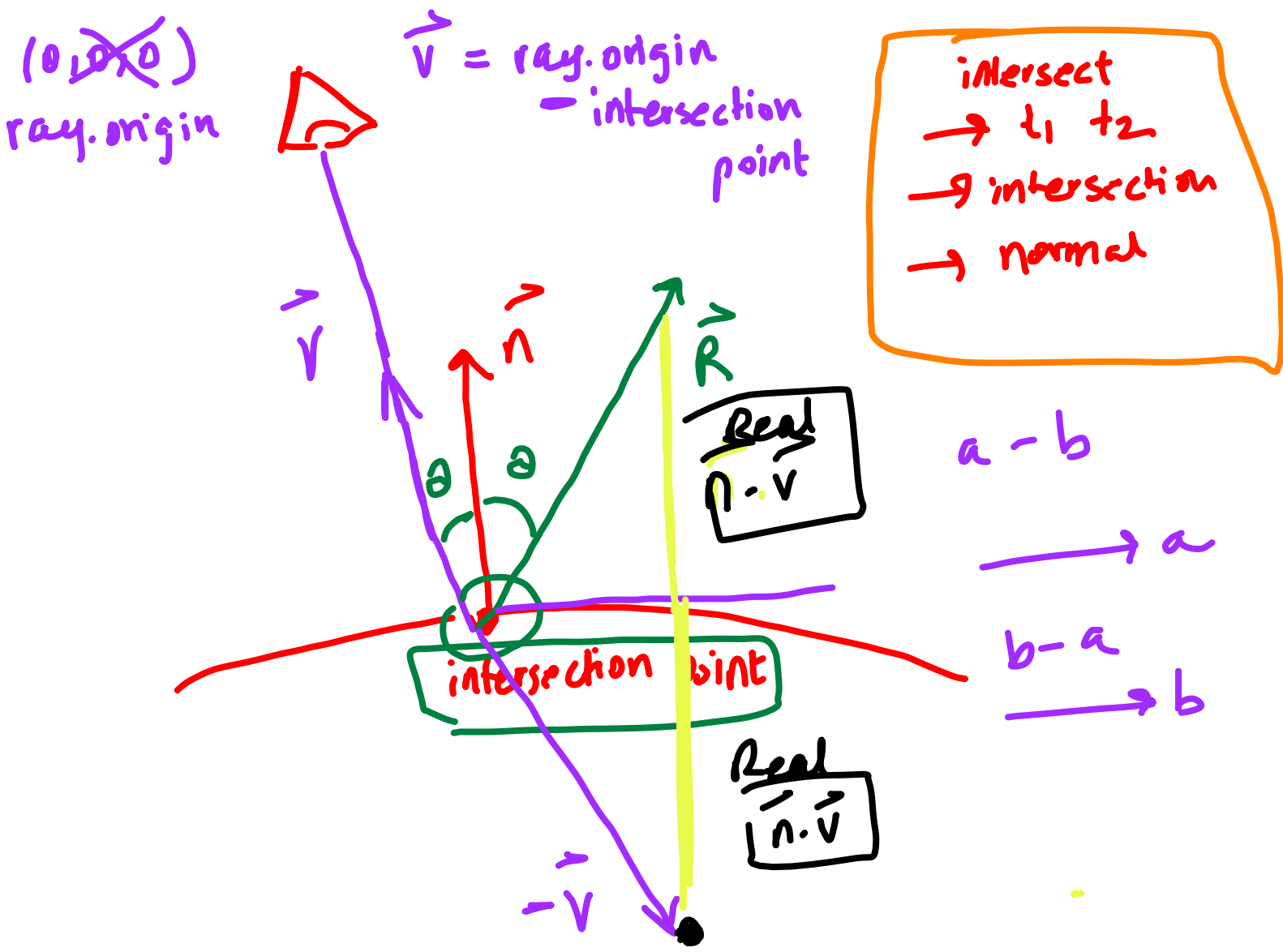
t1 and t2
- positive
- least of the two

VEC3 intersection point = ?

= origin + (t2 x direction)

ray = origin + t x direction

ray = origin + ~~t x direction~~



$$\vec{R} = -\vec{V} + 2(\vec{n} \cdot \vec{V})\vec{n}$$

VEC3 vector = vector + Real