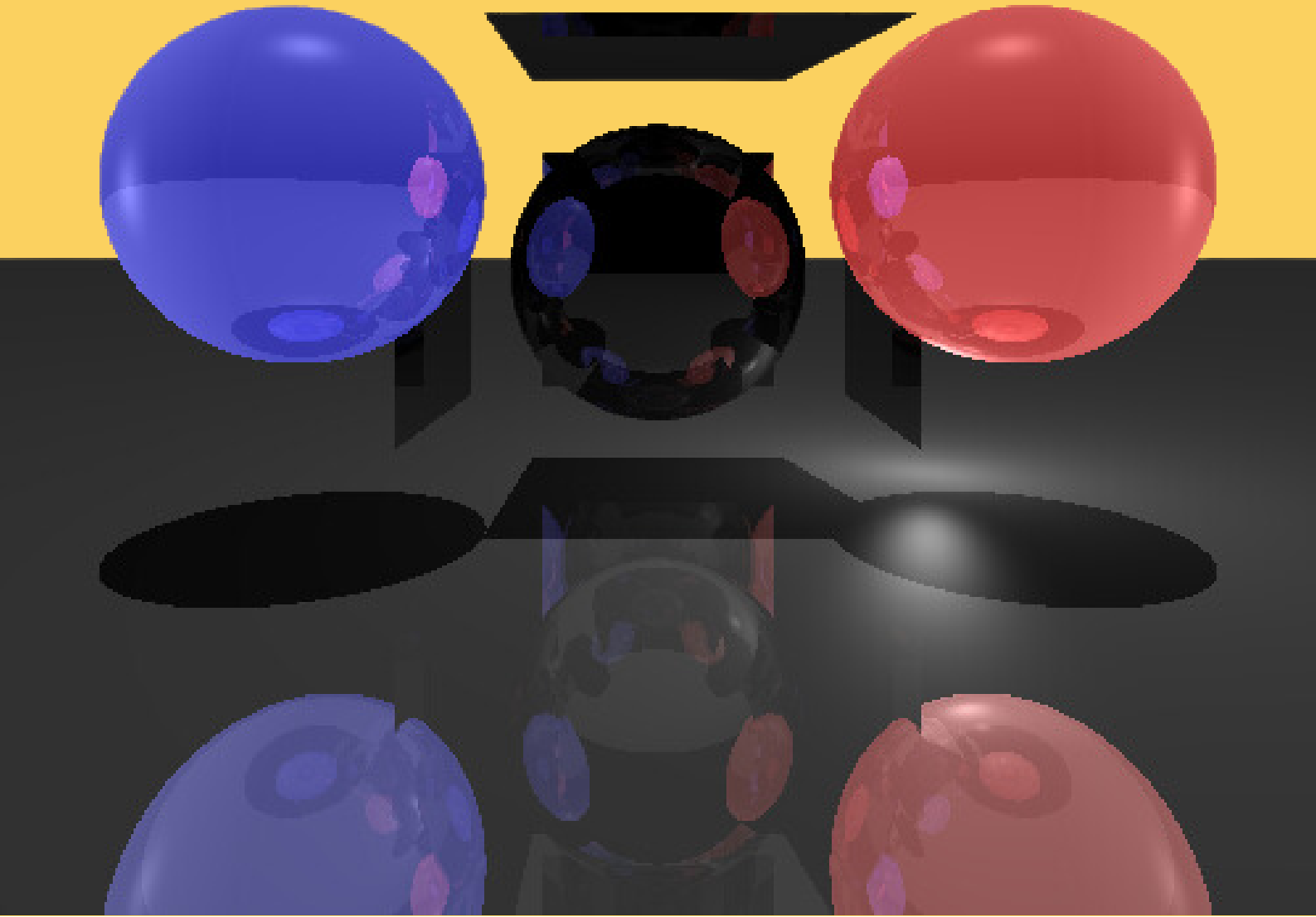
דו"ח שיפורים

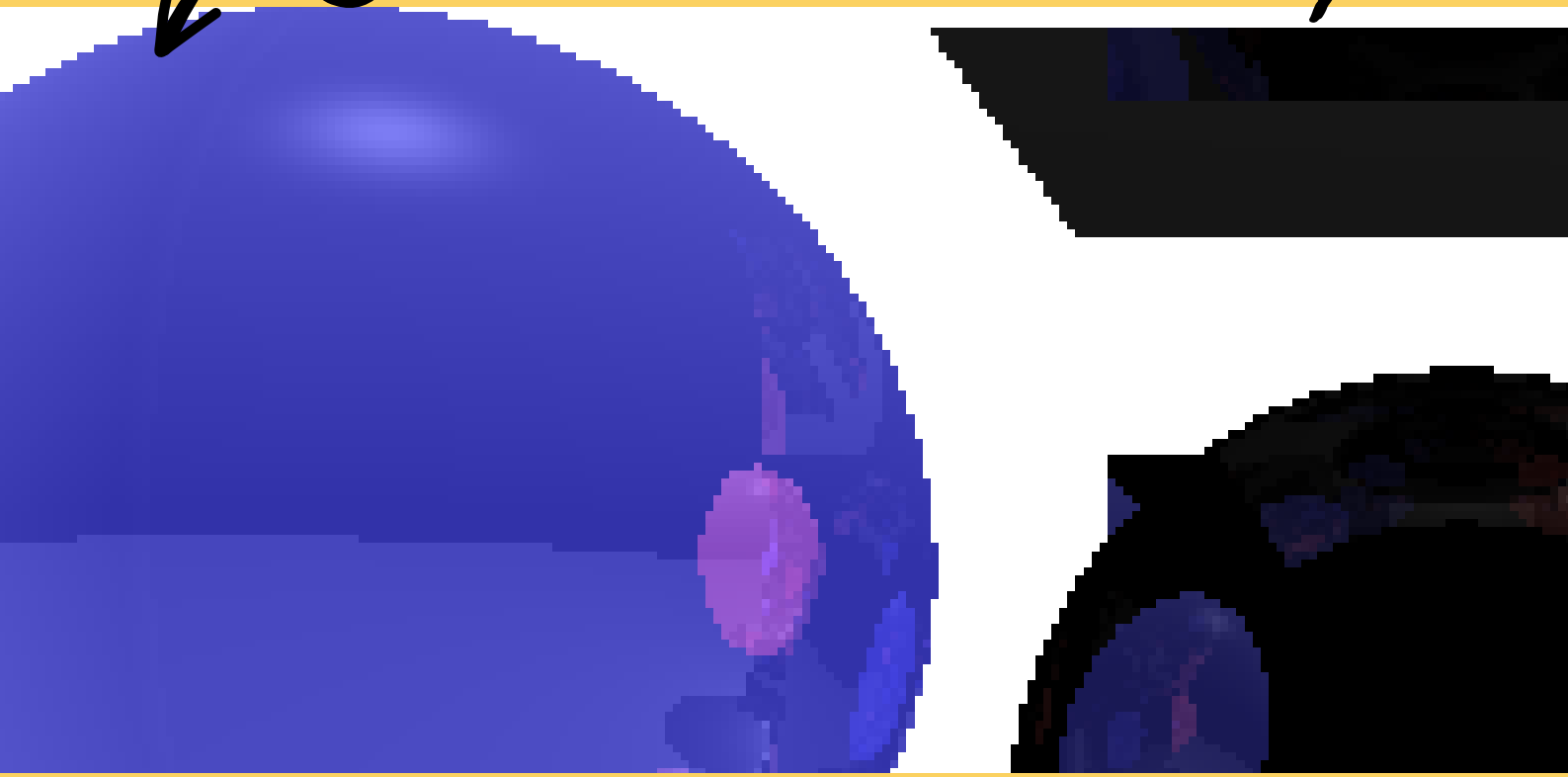
מיני פרויקט במבוא להנדסת תוכנה

אהובה ראשית אביתי ואביטל הרוביץ

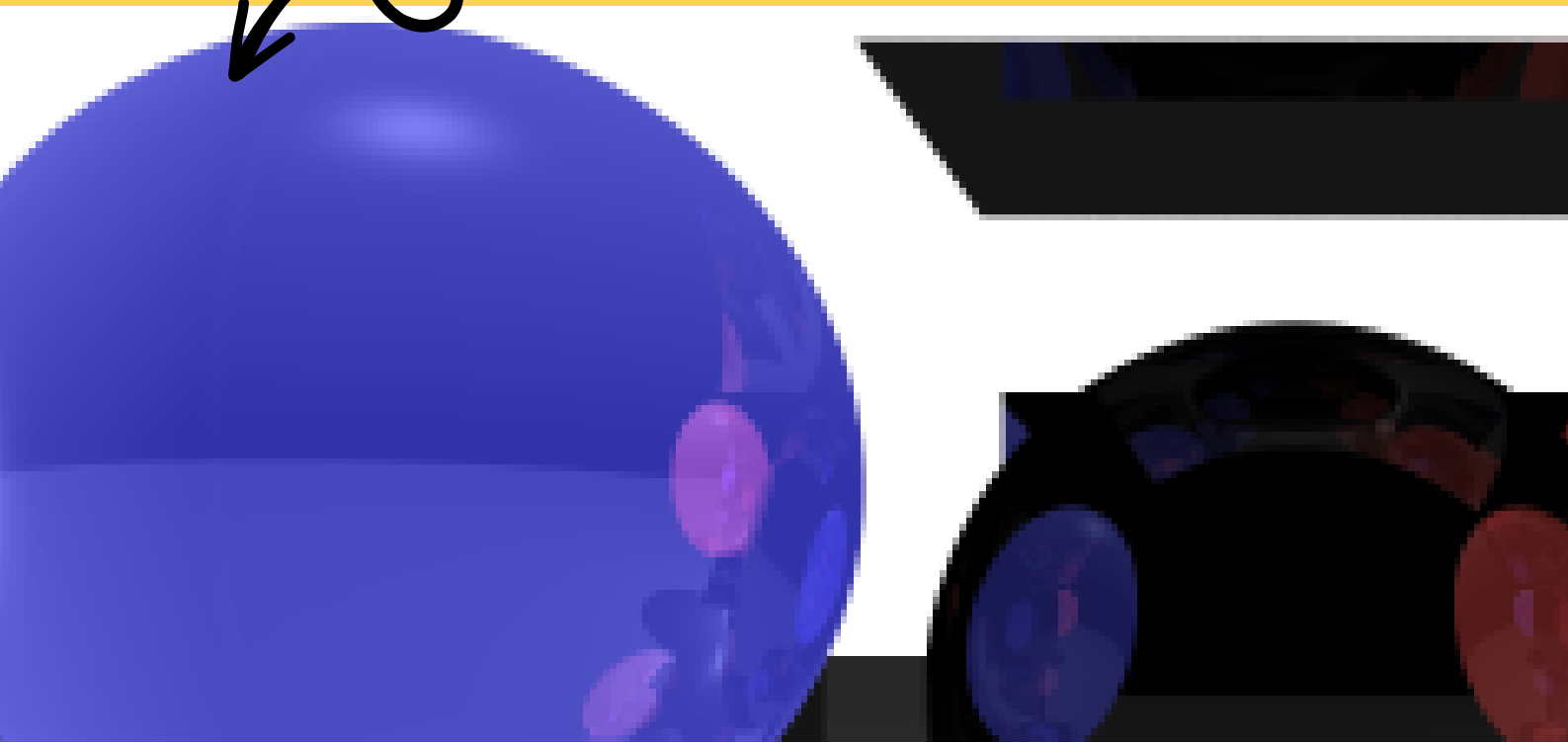
התמונה הבאה ממש מטריקסלית
וגם מבאס.
מנע ששיפרנו אותה!
בטבשו ותראו...



רק שתביני את רמת הפיקסלים...



שם יש ביצוע מקבילי מאוד של
מה שנקרא החלקת לקומות

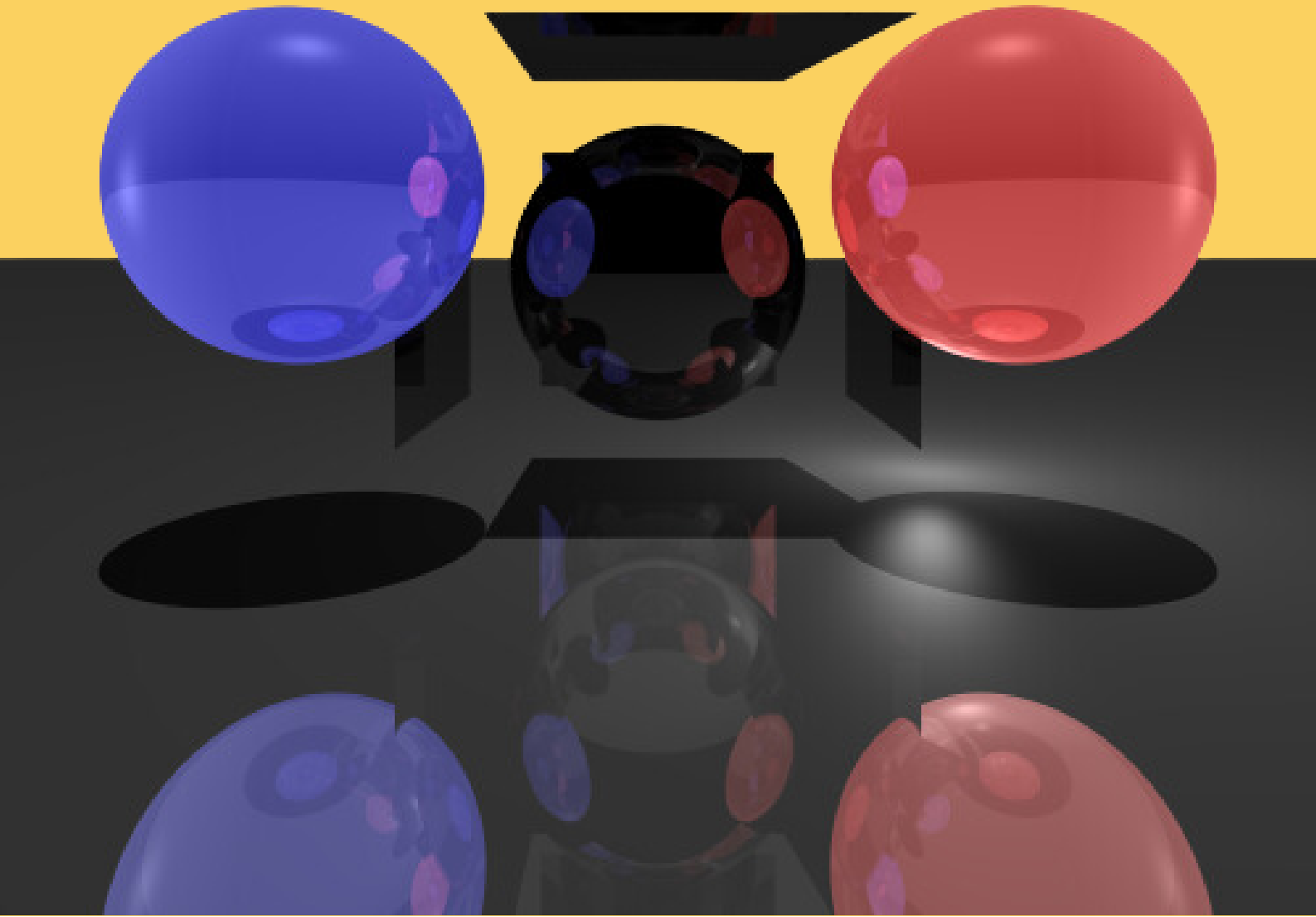


ביצענו anti alising = החלקת עקומות,
באמצעות חישוב ממוצע הצבע בכל פיקסל. וכך ישנה הדרגתיות במעברי צבע, כך שלעין זה נראה יותר אחיד ויפה.

את חישוב הממוצע עשינו בשתי דרכים שונות:

1.בדרך הראשונה החלטנו רנדומלית על המיקום של כל נקודה בתוך הפיקסל (בהתאם למספר הנקודות המבוקש) על סמך חישוב הצבע של נקודות רנדומליות אלו, חושב ממוצע הצבעים וניתן הצבע לפיקסל כולו.
דרך זו אמנם ביצעה את מה שניסינו להשיג - אך בצורה איטית מאוד.

לשם כך ביצענו את הדרך השניה:
2. בדרך זאת חושב קודם כל הצבע בארבעת פינות הפיקסל, ובמידה והוא לא היה תואם, הפיקסל חולק לארבע חלקים ובהם גם כן חושבו ארבעה פינות. עבור כל חלק הלולאה בוצעה מחדש: אם ארבעת הפינות לא תאמו, בוצעה עוד חלוקה. וכך חושב הממוצע. דרך זו מהירה בהרבה מהדרך הקודמת כי היא מצמצמת את כמות הקרניים שנצטרך לחשב את צבען.



שיפור נוסף שעשינו על מנת להאיץ את ייצור התמונה הוא חילוק המשימה למספר תהליכונים שונים הרצים במחשב במקביל.
וכך ייעלנו את ביצוע המשימה.

```
//over all the threads
while (threadsCount-- > 0) {
    new Thread(() -> {
        //over all the pixels ant print its
        for (Pixel pixel = new Pixel(); pixel.nextPixel(); Pixel.pixelDone())
            imageWriter.writePixel(pixel.col, pixel.row, SuperSampling(imageWriter.getNx(),
                    imageWriter.getNy(), pixel.col, pixel.row, antiAliasing, adaptive));
    }).start();
}
```

בונוס שלב 2 - נורמל לגליל סופי

```
@Override
public Vector getNormal(Point point) {

    //dotProduct between the axisRay to (point-p0)
    double tmp = axisRay.getDir().dotProduct(point.subtract(axisRay.getP0()));
    //on the base
    if (tmp == 0)
    {...}
    //on the upper base
    if (tmp==height)
    {
        //in the center of each base
        if (point.subtract(axisRay.getP0()).length() == height)
            return axisRay.getDir();
        Point point2 = point.add((axisRay.getDir()));//v
        return point2.subtract(point).normalize();
    }
    //on the round surface
    Point center = axisRay.getP0().add(axisRay.getDir().scale(tmp));
```

בונוס שלב 3 -

חיתוך עם מצולע,

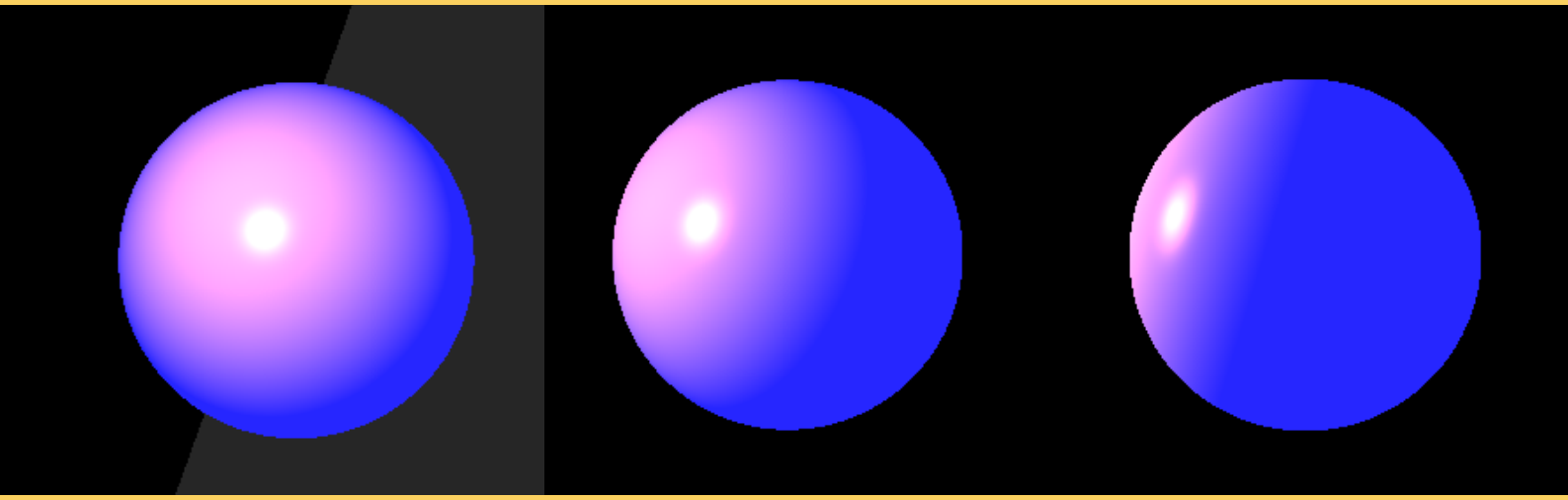
חיתוך עם גליל אין סופי

חיתוך עם גליל סופי.

```
/**
 * A method that receives a ray and checks the points of GeoIntersection of the ray with the tube
 * @param ray the ray received
 * @return null / list that includes all the GeoIntersection points (contains the geometry (shape) and the point
 */
3 usages   1 override   ahuva12 +1
@Override
protected List<GeoPoint> findGeoIntersectionsHelper(Ray ray, double maxDistance) {
    /*
```

בונוס שלב 4 -

טרנספורמציית מיקום וסיבוב של מצלמה

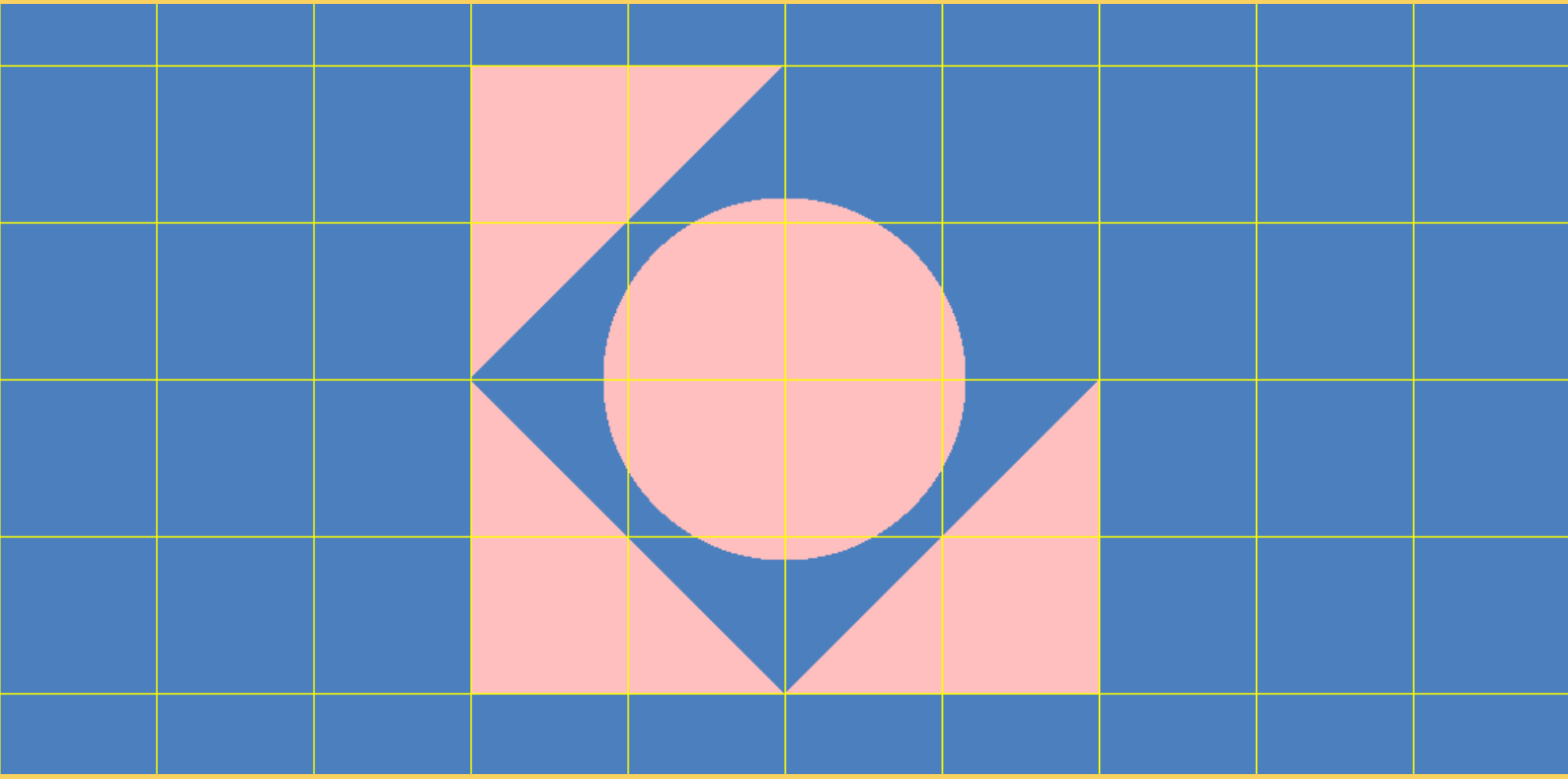


```
    public Camera rotate(double x, double y, double z) {
        //vector vTo after the rotation
        vTo = vTo.rotateX(x).rotateY(y).rotateZ(z);
        //vector vUp after the rotation
        vUp = vUp.rotateX(x).rotateY(y).rotateZ(z);
        //vector vRight after the rotation
        vRight = vTo.crossProduct(vUp);

        //return the camera after the rotation
        return this;
    }
}
```

בונוס שלב 5 -

שימוש בקבצי xml



בונוס שלב 6-

סיום מלא תוך שבוע

ספוט ממוקד

```
public PointLight setNarrowBeam(double narrow) {
    narrowBeam = narrow;
    return this;
}
```

בונוס שלב 7-

תמונה עם 10+ עצמים

צילום בונוס ראשון מזויות שונות

פתרון בעיית מרחק בהצללה בדרך 2