

מיני פרוייקט במבוא להנדסת תוכנה רנדור תמונה

מגישות:

נטע אבידר והילה לב-הר

מרצה מנחה:

אליעזר גנסבורגר

תיאור הפרויקט-

בפרויקט יצרנו כלים ליצירת תמונות דיגיטליות שנראות דרך view plane מהמצלמה עם מיקום וכיוונים , יצירת התמונה על ידי הגדרת סצנה המכילה צורות גאומטריות ומקורות אור.
הפרוייקט נכתב ב-JAVA ב-IntelliJ עם שימוש ב-Junit.
מבנה:

שיטת תכנות:

מיני פרוייקט 1 – שיפור תמונה – soft shadow

הבעיה:

צללים בתמונה יוצאים חדים ולא מציאותיים, אין מריחה/דעיכה. בעיה זו נוצרת בגלל שעבור כל נקודת חיתוך על קרן מהמצלמה מוצאים קרן מהנקודה למקור האור (קרן -L) ובעזרתה מחשבים האם יש גוף שמסתיר לנקודה את האור, אם אין נכניס את ההשפעה של האור לחישובים של צבע הנקודה, אם יש גוף שמסתיר אז לא נכניס ויוחזר 1 למקדם של ktr, שלומר אין שקיפות בכלל, מה שגורם לצל לצאת חד ולא מציאותי.

```
private Double3 transparency(GeoPoint geoPoint, LightSource light, Vector l, Vector n) {  
    Vector lightDirection = l.scale( scalar: -1d); // from point to light source  
    Ray lightRay = new Ray(geoPoint.point, n, lightDirection);  
    List<GeoPoint> intersections = scene.geometries.findGeoIntersections(lightRay);  
  
    if (intersections == null ) {  
        return Double3.ONE;  
    }  
}
```

הפתרון:

הפתרון לבעיה הוא במקום שחישוב הצל יקבע משליחת קרן אחת, נשלח אלומת קרניים, ועבור כל קרן באלומה נחשב את מקדם ktr ונחזיר את הממוצע. (ל ktr בחישוב הצבע בנקודה)

```
private Double3 softShadow(GeoPoint gp, LightSource lightSource, Vector n) { 1 usage  +1 hila  
    Double3 sumTransparency = Double3.ZERO;  
    List <Vector> lst = lightSource.getListL(gp.point, scene.getNumOfRaysAtBeam());  
    for (Vector lAroundLight: lst) {  
        sumTransparency = sumTransparency.add(transparency(gp, lightSource, lAroundLight, n));  
    }  
    return sumTransparency.reduce(lst.size());  
}
```

על מנת לבצע את השיפור נדרשנו להוסיף שדות ומתודות לחלק מהמחלקות. היה לנו חשוב שמיקום המשתנים החדשים שהוספנו יהיה לפי עיקרון RDD

שמאפשר גם למשתמש גישה נוחה לקביעת פרמטרים אלו.
הפרמטרים-

1. כמות הקרניים באלומה, הוספנו שדה NumOfRaysAtBeam ב Scene .

2. האם השיפור יופעל או לא . SoftShadow ב Scene .

```
public class Scene {  ⚡ Neta Avidar +1
    public String name; 2 usages
    public Color background = new Color( r: 0, g: 0, b: 0); 3 usages
    public AmbientLight ambientLight = AmbientLight.NONE; 2 usages
    public Geometries geometries = new Geometries() ;
    public List<LightSource> lights = new LinkedList<>(); 24 usages

    private boolean softShadow = false; 2 usages
    private int numOfRaysAtBeam = 1; 2 usages
```

3. הפתרון מתחשב בגודל הרדיוס של גוף התאורה point/spot light , כלומר
הצל יצא שונה עבור כל גוף תאורה ספציפי ולא רק עבור כל סוג שונה .
הוספנו שדה radius במחלקה PointLight .

```
public class PointLight extends Light implements LightSource{ 11 usag

    final Point position; 11 usages
    private double kQ = 0d; 2 usages
    private double kL = 0d; 2 usages
    private double kC = 1d; 2 usages
    private double radius =0; 9 usages

    //constructor
    public PointLight(Color intensity, Point position, double r ) {
        super(intensity);
        this.position = position;
        radius = r;
    }
```

הקפדנו על הערות jcd במחלקות ומתודות חשובות ועל יצירת ערכי ברירת מחדל (כגון soft shadow false , שדה הרדיוס ב point/spot light כך שאין אתחול השיפור לא יפעל)

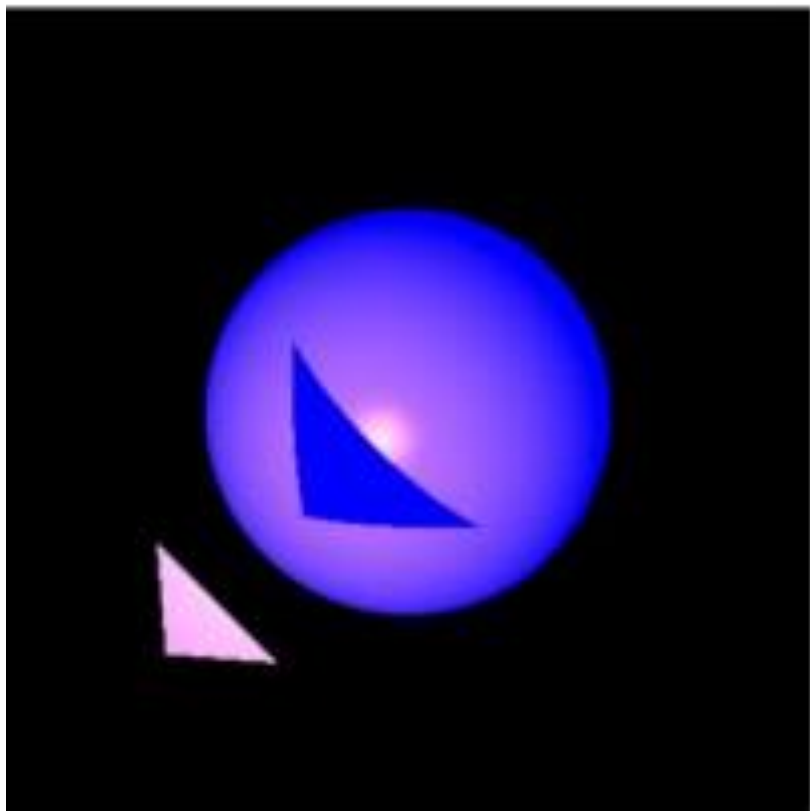
קביעת דרך החישוב של ktr במתודה calLocalEffect שמחשבת את האפקטים הלוקליים עבור נקודה ומחזירה צבע בהתאם , כאשר עושים את השיפור (שדה soft shadow true) ktr יחושב במתודה softShadow, וכאשר אין את השיפור ktr יחושב במתודה transparency .

```
if(scene.isSoftShadow()) {  
    ktr = softShadow(gp,lightSource,n);  
}  
else {  
    ktr = transparency(gp,lightSource,l,n);  
}
```

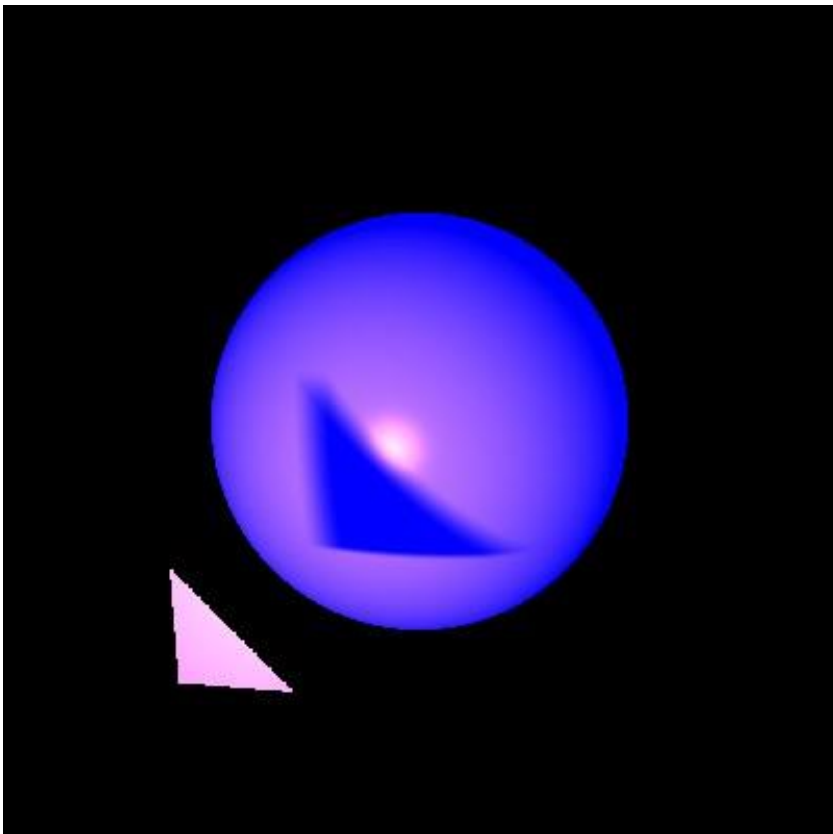
המשך ה- transparency

```
Double3 ktr = Double3.ONE;  
double pointLightD = light.getDistance(geoPoint.point);  
  
for (GeoPoint intresectGP : intersections) {  
    if (geoPoint.point.distance(intresectGP.point) < pointLightD){  
        ktr = ktr.product(intresectGP.geometry.getMaterial().kT);  
        if (ktr.lowerThan(MIN_CALC_COLOR_K)) {  
            return Double3.ZERO;  
        }  
    }  
}  
return ktr;  
}
```

לפני השיפור -



אחרי השיפור-



כאשר soft shadow עם ערך false (כלומר בלי השיפור)-



```
scene.setSoftShadow(false);
```

```
✓ Minip1 (renderer 3 min 25 sec  
✓ ourPicture() 3 min 25 sec
```

כאשר soft shadow עם ערך true (עם השיפור)-



```
scene.setSoftShadow(true);
```

```
✓ Minip1 (renderer) 10 min 3 sec  
✓ ourPicture() 10 min 3 sec
```