

דוח מס' 1: שיפורי איכות התמונה

מגישות: חיה יזרסקי 5876 ונחמה ברון 8830

פרטי התמונה:

זה המצלמה, והיא נמצאת מעט שמאלה, לא ישר על הצירים:

```
Camera camera = new Camera(new Point(65, 5, 20), new Vector(-7, -0.5, -2), new Vector(-1, 0, 1))
    .setVPSize(300, 300).setVPDistance(800)
```

ציפור: גוף וכנפיים

```
Geometries geometries = new Geometries(

    //wings
    //left
    new Triangle(new Point(14, 4, 4), new Point(14, 1, 1), new Point(9, 4, -10)).setEmission(Color.YELLOW),
    //right
    new Triangle(new Point(14, 4, 4), new Point(14, 7, 1), new Point(9, 4, -10)).setEmission(Color.YELLOW),
    //BODY
    new Sphere(new Point(14, 4, 1), radius: 3)
        .setEmission(Color.YELLOW)
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),

    //HEAD
    new Sphere(new Point(16, 4, 3.5), radius: 2)
        .setEmission(Color.YELLOW)
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
```

עיניים של הציפור:

```
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
    //PUPIL
    new Sphere(new Point(17, 5, 4.6), radius: 0.3)
        .setEmission(Color.BLACK)
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
    //PUPIL
    new Sphere(new Point(17, 3, 4.6), radius: 0.3)
        .setEmission(Color.BLACK)
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
    //white around eyes
    new Sphere(new Point(16.8, 4.9, 4.6), radius: 0.4)
        .setEmission(Color.WHITE)
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),

    new Sphere(new Point(16.8, 3.1, 4.6), radius: 0.4)
        .setEmission(Color.WHITE)
        .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
```

ביצים:

```
//eggs
```

```
new Sphere(new Point(18, 2,1), radius: 0.7)
    .setEmission(new Color(java.awt.Color.gray))
    .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
new Sphere(new Point(18, 3, 1), radius: 0.7)
    .setEmission(new Color(java.awt.Color.gray))
    .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
new Sphere(new Point(18, 4, 1), radius: 0.7)
    .setEmission(new Color(java.awt.Color.gray))
    .setMaterial(new Material().setKd(0.5).setKs(0.5).setShininess(30)),
// ...
```

מקור, וענף עליו הביצים:

```
//mouth beak
new Triangle(new Point(19.5, 4.3, 2),new Point(17.5, 5, 4),new Point(17.5, 3.3, 4)).setEmission(Color.ORANGE),
new Triangle(new Point(19.5, 4.3, 3),new Point(17.5,4.3, 4.5),new Point(17.5, 3.3, 4)).setEmission(Color.ORANGE),
new Triangle(new Point(19.5, 4.3, 3),new Point(17.5, 5, 4),new Point(17.5,4.3, 4.5)).setEmission(Color.ORANGE),
//BRANCH
new Quadrangle(new Point(17, 0, 0.3),new Point(17, 8, 0.3),new Point(19, 8, 0.3),new Point(19, 0, 0.3)).setColor(Color.BROWN),
```

כדי להראות את אפקט המראה, הוספנו גם בית שמשקף את הציפור:

```
//house
//front of house
new Quadrangle(new Point(10, 0, 0.3),new Point(10, 8, 0.3),new Point(10, 8, 10),new Point(10, 0, 10)).setColor(Color.BROWN).setReflectivity(new Double3( value: 0.2)),
//right of house
new Quadrangle(new Point(2, 8, 0.3),new Point(10, 8, 0.3),new Point(10, 8, 10),new Point(2,8,10)).setColor(Color.BROWN),
//left of house. it is crooked so I can view it
new Quadrangle(new Point(2, 0, 0.3),new Point(10, 0, 0.3),new Point(10, 0.1, 10),new Point(2,0.1,10)).setColor(Color.BROWN),
//back of house
new Quadrangle(new Point(2, 0, 0.3),new Point(2, 8, 0.3),new Point(2, 8, 10),new Point(2, 0, 10)).setColor(Color.BROWN),
```

אפשר לראות שהחלק הקדמי מאותחל כמשתקף.

גג:

```
//top of house-roof-left
new Quadrangle(new Point(11, 4, 13),new Point(11, -1, 10),new Point(1, -1, 10),new Point(1, 4, 13)).setColor(Color.RED),
//top of house- roof - right
new Quadrangle(new Point(11, 9, 10),new Point(11, 4, 13),new Point(1, 4, 13),new Point(1, 9, 10)).setColor(Color.RED),
//bubbles in the sky
```

ולסיכום, הוספנו בועות, כדי להראות שאפשר ליצור גם אובייקטים שקופים מעט:

```
//bubbles in the sky
new Sphere(new Point(12, 0, 4), radius: 1).setMaterial(new Material().setKT(0.99).setShininess(30)).setEmission(Color.BLUE),
new Sphere(new Point(13, 0,7), radius: 0.7).setMaterial(new Material().setKT(0.99).setShininess(30)).setEmission(Color.BLUE),
new Sphere(new Point(12, 6, 4.5), radius: 0.5).setMaterial(new Material().setKT(0.99).setShininess(30)).setEmission(Color.BLUE),
new Sphere(new Point(13.5, 8, 4), radius: 0.7).setMaterial(new Material().setKT(0.99).setShininess(30)).setEmission(Color.BLUE)
```

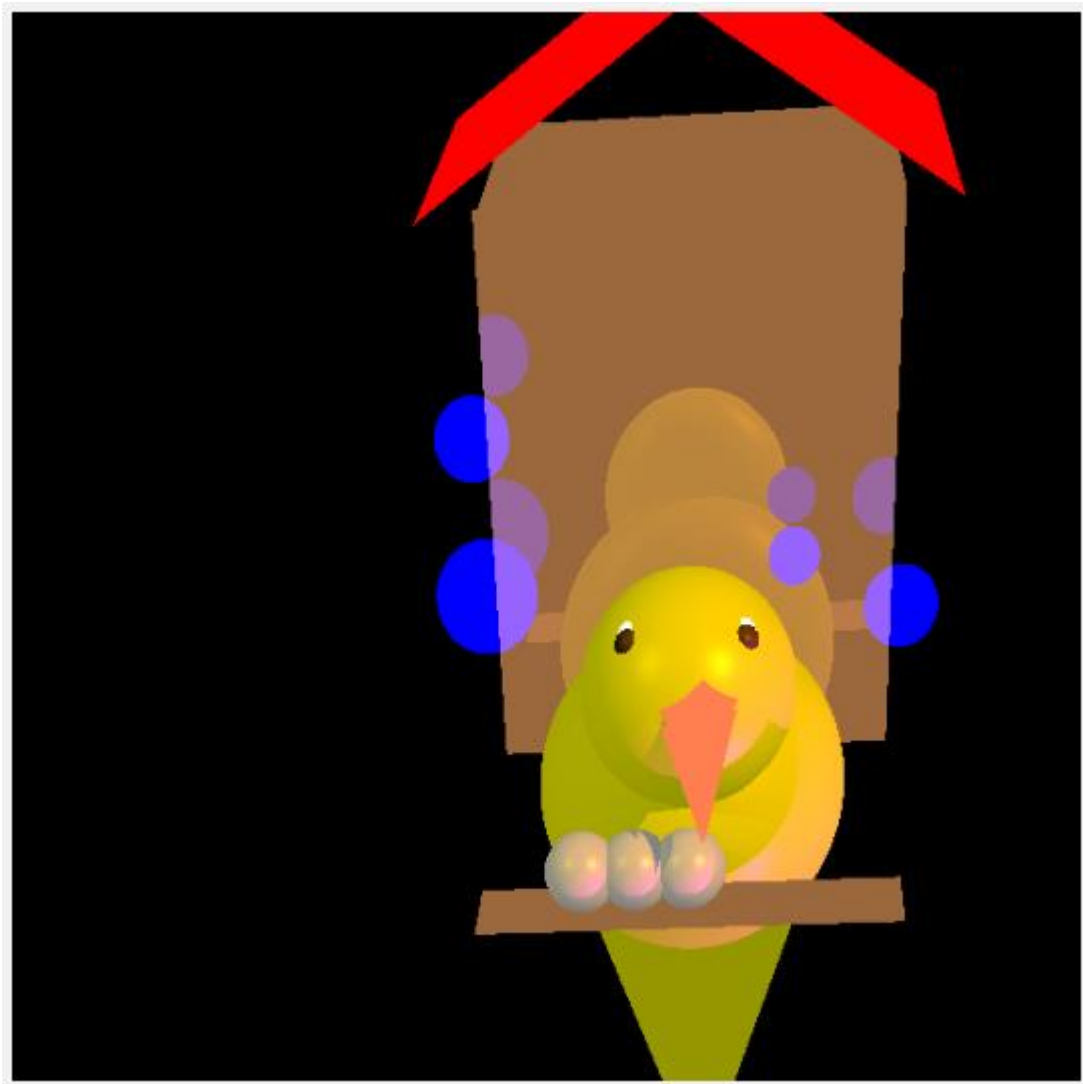
תאורה מכל סוג:

```
LinkedList<LightSource> lights = new LinkedList<>();
lights.add(new Spotlight(Color.MAGENTA,
    new Point(40, 40, -40),
    new Vector( x: -1, y: -1, z: 4)).setKl(0.0001).setKq(0.0001));
lights.add(new PointLight(Color.YELLOW, new Point(100, -100, 100)).setKl(0.001).setKq(0.0001));

lights.add(new DirectionalLight(Color.GOLD,new Vector( x: -2, y: -2, z: -2)));
```

שאר בניית התמונה היא כמו כל טסט אחר.

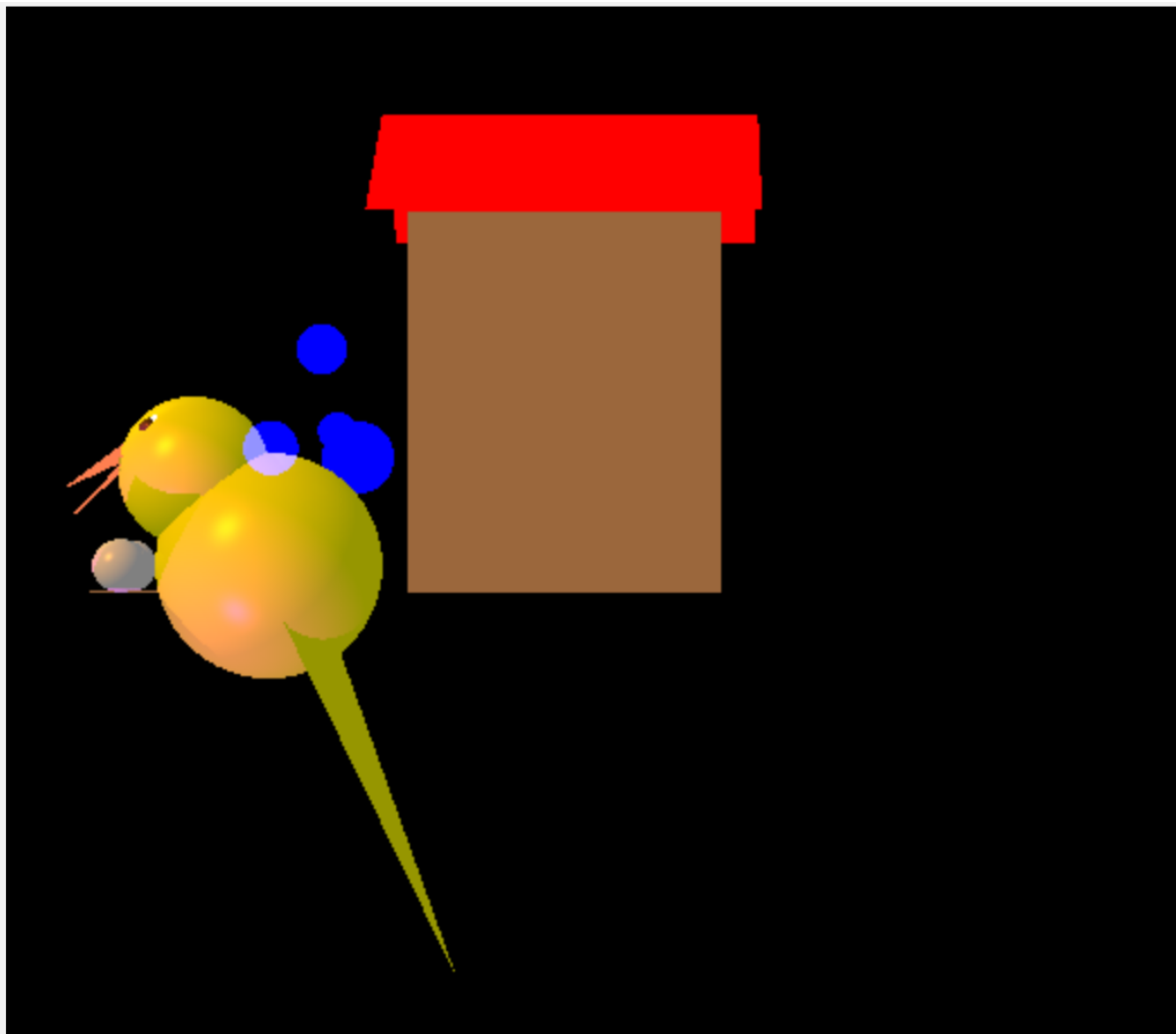
התמונה עם כלל האובייקטים:



בנינו גרסאות דומות מאוד של הטסט הזה, כדי להראות שהתמונה נראית טוב בתלת מימד לדוגמה, מהצד :

```
//side camera
Camera camera = new Camera(new Point(3,100,1), new Vector( x: 0, y: -1, z: 0), new Vector( x: 0, y: 0, z: 1))
.setVPSize( width: 300, height: 300).setVPDistance(800);
```

התמונה מהצד נראית כך :

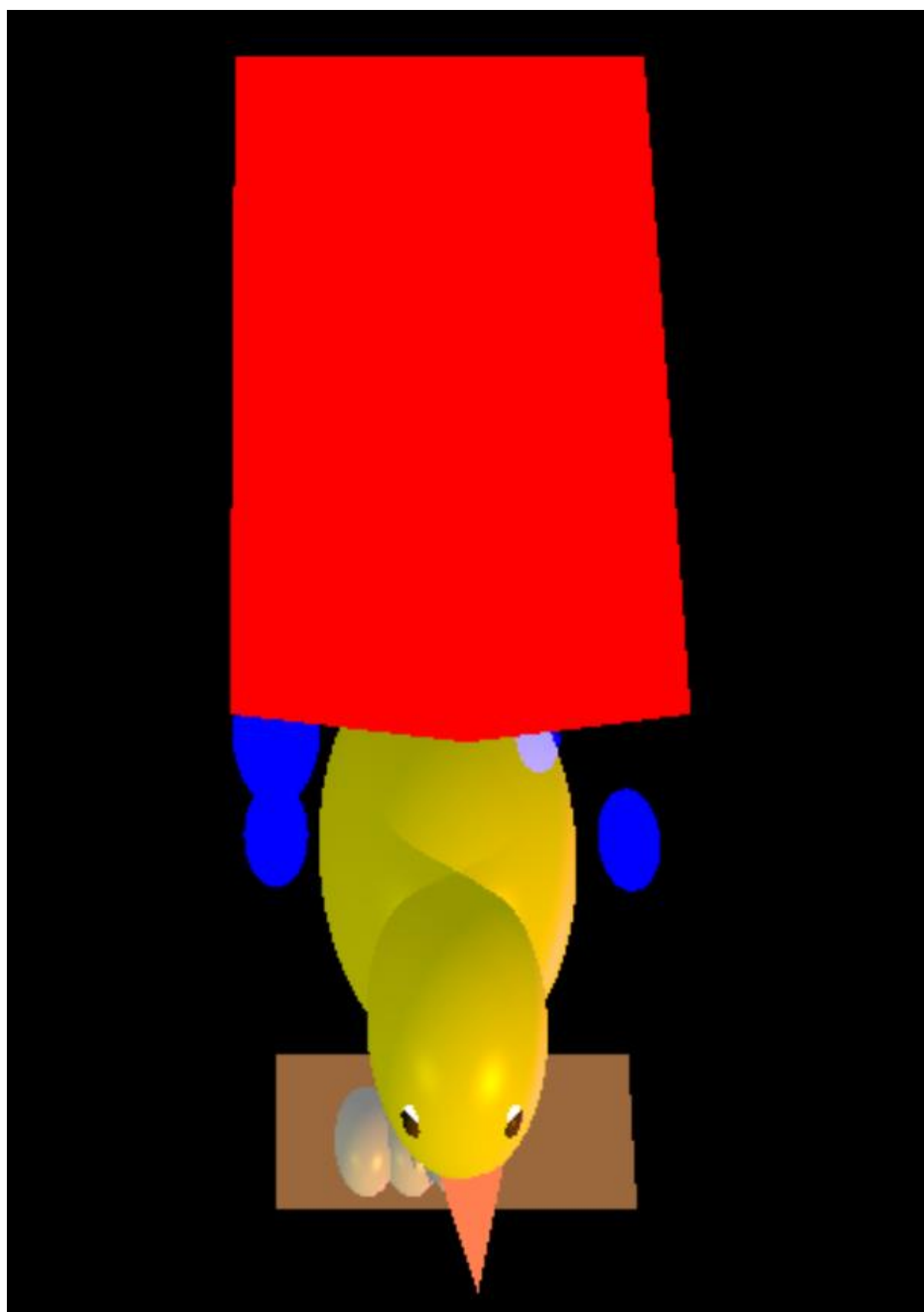


מלמעלה:

```
Camera camera = new Camera(new Point(0,0,100), new Vector( x: 0.1, y: 0, z: -1), new Vector( x: -1, y: 0, z: 1))  
.setVPSize( width: 300, height: 300).setVPDistance(800);
```



התמונה מלמעלה:



שיפורי התמונה:

: DEPTH OF FIELD

תיאור בעיה: התמונה חדה מידי, אפשר לראות את כל האובייקטים כאילו הם בפקוס. היינו רוצים לקבל תחושה של מרחק, ז"א שהמצלמה רואה נקודה אחת ברור, וכל מה שלפניה או אחריה, יוצא מטושטש.

פתרונות אפשריים:

לפי המצגת, ניתן לחשב ריבוע ספציפי שהאמצע שלו זה הנקודה שלה אנחנו מחשבים צבע, ודרך ארבעת הקצוות מעבירים וקטורים, מחשבים את הצבע לפי הוקטורים, ועושים ממוצע בין הצבעים. ככל שמשוהו רחוק יותר, יהיה הפרש גדול יותר בין הצבעים, ויצא מטושטש.

פתרון שלנו: העברת קרן מהמצלמה דרך הנקודה של הפיקסל ומציאת נקודה שרחוקה במידה שנרצה. ז"א, מה שנמצא במרחק הזה יראה מדויק וברור. חישוב נקודה זו נעשית עבור כל פיקסל מחדש. לאחר מכן נעביר קרניים דרך אותה הנקודה מנקודות סביב המצלמה, ובעזרת אותם הקרניים נחשב את הצבעים ונעשה ממוצע ביניהם. הממוצע הזה הוא הצבע הסופי של הפיקסל.

ככה נראה הסטר

```
camera.setDepthButton( button: true, apertureSize: 3, focalLength: 50);
```

תמונה :



: ANTI ALIASING

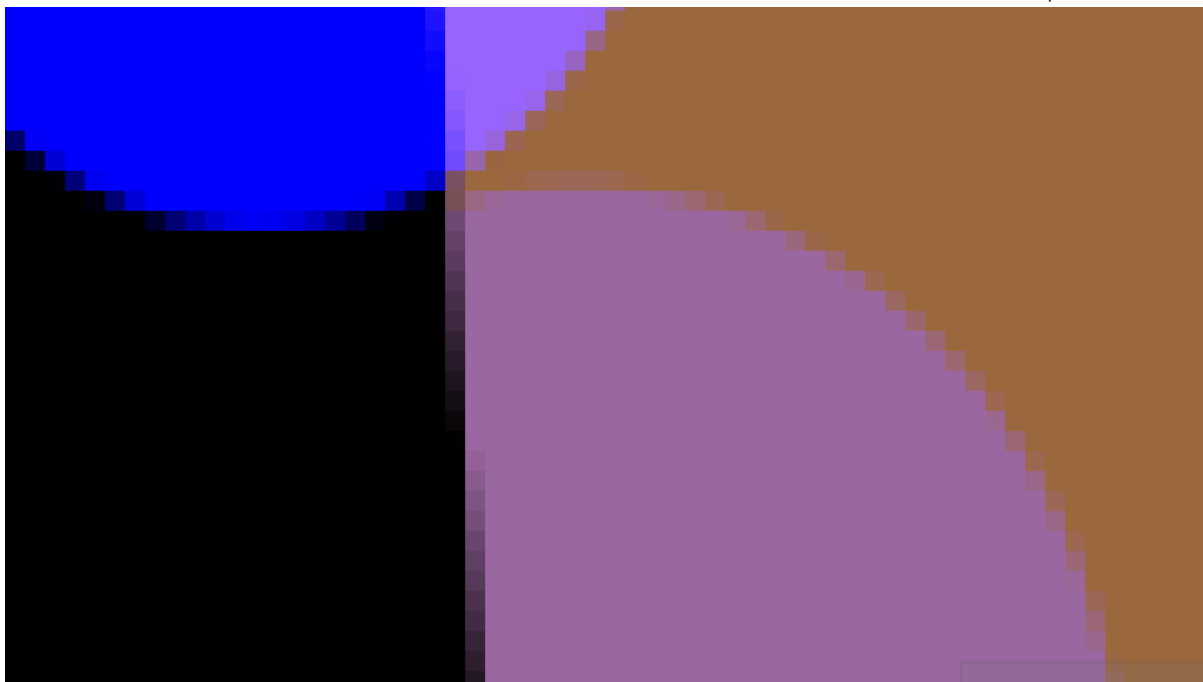
תאור בעיה : הקצוות של האובייקטים חדים מדי. רואים את זה במיוחד באובייקטים עגולים.
פתרונות אפשריים : במקום נקודה אחת בכל פיקסל, ניקח כמה נקודות בכל פיקסל ונעשה ממוצע בין הצבעים שלהם.
הפתרון שלנו : לקחת את השיטה הזו ולשכלל אותה- במקום נקודות רנדומליות, לחלק את הפיקסל לגריד, ולעשות ממוצע בין הצבעים שמוצאים דרכם.
מדליקים את השיפור בעזרת סטר-

```
camera.setJaggedEdgesButton( jaggedEdgesButton: true, numberOfMiniPixels: 20 );
```

תמונה :



נצלם יותר מקרוב: אחרי השיפור:



ללא שיפור:



SOFT SHADOWS:

בעיה: הצללים יותר מדי חדים, המעברים בין האור והצל.

פתרונות אפשריים:

במקום לעשות וקטור אחד בין הווקטור לנקודת האור (לכן אי אפשר לעשות את השיפור על DIRECTIONAL)

עושים שליחה של כמה קרניים, אל נקודות סביב התאורה. אפשר לשלוח סביב הנקודה בצורה של כדור, אפשר בצורה של משולש, אפשר בצורה של מרובע, ועוד.

פתרון שלנו: בחרנו בצורה של מלבן, ככה אפשר להשתמש בפונקציית בחירת הנקודות שבה השתמשנו גם בשיפורים האחרים.

ככה נראה הסטר-

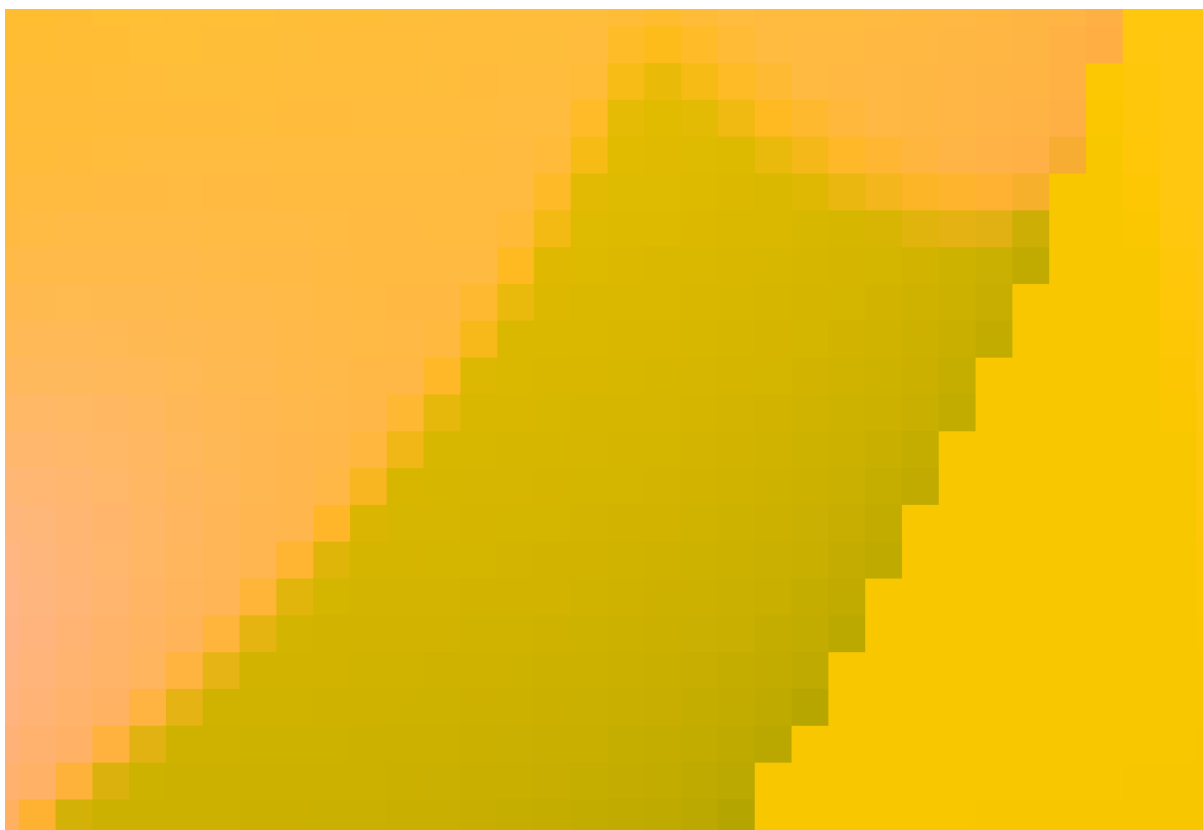
```
ImageWriter imageWriter = new ImageWriter( imageName: "bird from up top", nX: 600, nY: 600);
camera.setImageWriter(imageWriter)
    .setRayTracer(new RayTracerBasic(scene2).setSoftShadowsButton( softShadowsButton: true, number_of_points: 30))
    .renderImage()
    .writeToImage();
```

עשינו את הטסט עם 30*30 נקודות סביב התאורה.

תמונה:



וביותר מקרוב:
עם השיפור:



ללא שינוי:



כל השיפורים האלה מאיטים מאוד את זמן הריצה, ועל כן נסביר בדוח הבא איך לתקן זאת.