בס"ד

דו"ח מיני פרויקט

מגישות: מעין זוהר – 209129832

רננה עמנואל – 318659893

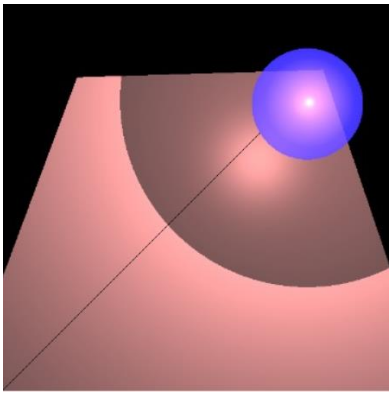
שלב ראשון

השיפור הראשון, הוא שיפור הצל – soft shadow.
לפני השיפור צבע הצל היה אחיד, במידה ונק' מסוימת הוסתרה ע"י גוף ממקור האור, נוצר צל וצבעו היה אחיד. אך תמונה כזו היא אינה אמיתית מאחר וקצוות הצל לרוב יהיו בגוון בהיר יותר מאחר וכן מגיע אליהם קצת אור.
על מנת לשפר נק' זו, מכל נק' צל, שלחנו 81 קרניים אל מקור האור וסביבתו וחישבו את עוצמת הצל מכל קרן (ktr שערכו נע בין 1-0), וכך עצמת הצל בנק' נלקחה כממוצע של 81 הקרניים.

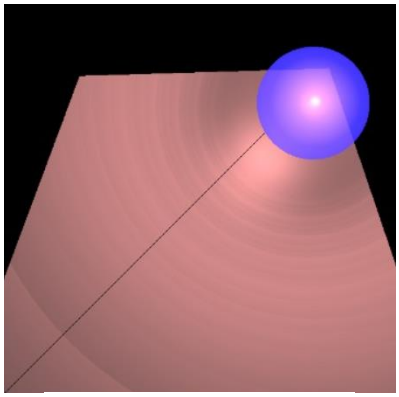
הפונקציה שמחשבת את צבע הנק' עם כל האפקטים calcLocalEffects, מזמנת את הפונקציה callTransparency, שבה מתבצעת הלולאה הבאה שמטילה 81 קרניים מנק' אל הסביבה של מקור האור:

```
//send ray to calculate and sum the ktr
for(int i=0; i<81;i++) { // find 81 point around the light

    double t = 2 * PI * Math.random(); //
    double r = radius * Math.random(); // radius random

    double alpha = r * Math.cos(t); // parameter from random point
    double beta = r * Math.sin(t); // parameter from random point

    Point3D randomPoint = lightP0.add(v.scale(alpha).add(w.scale(beta))); //random point around light
    Vector randomL = geopoint.point.subtract(randomPoint).normalized(); // random vector around light
    sumKtr += transparency(light, randomL, n, geopoint); // sum of all ktr from random point around the light
}
return sumKtr/81; // average of ktr
```

ניתן לראות הבדל מהותי בטסט הבא שהרצנו לפני ואחרי השיפור:



הרצה לפני השיפור



הרצה לאחר השיפור

שלב שני

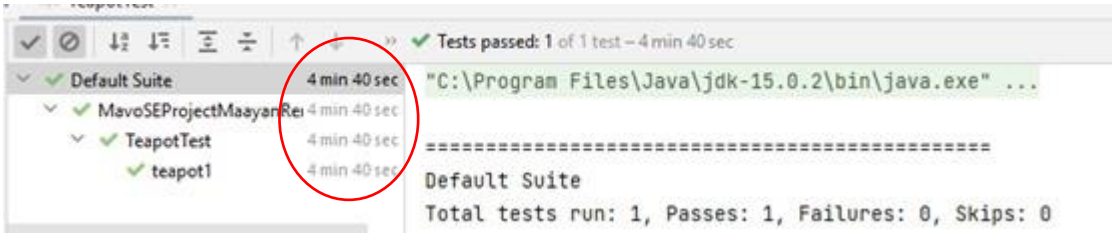
השיפור השני , הוא שיפור ביצועים - Adaptive supersampling.
לפני שיפור הביצועים, עבור כל נק' שנדרשנו לחשב לה צל, נשלחו 81 קרניים אל סביב התאורה, ועוצמת הצל נקבע ע"י ממוצע בין כל העוצמות שהתקבלו. כמובן ששליחת 81 היא מיותרת בהרבה מן המקרים והיא מעלה את העלות הכללית. השיפור מתבצע בכך שנשלח 4 קרניים אל קצוות התאורה ונראה האם עוצמת הצבע שחוזרת מהן, זהה. במידה והעוצמה אכן זהה, ניתן להסיק כי נק' זו כולה בצבע שהתקבל, אך במידה והעוצמות יהיו שונות, נצטרך לפעול בצורה רקורסיבית, כלומר, לחלק את מקור האור ל4 ולקחת את הנק' שחזרה עם צבע שונה ולזמן על האזור הזה את אותה הפעולה פעם נוספת. פעולה זו תתבצע עד שנקבל 4 פינות עם עוצמת צבע זהה או עד שנגיע לעומק הרקורסיה שהתקבל.
בדרך זו, נוריד משמעותית את כמות הקרניים שנשלחות ואת זמן הריצה של הפונקציה (מאחר ולכל קרן שנשלחת עוברת בדיקות רבות).

הפונקציה שמחשבת את עוצמת הצל ב4 קצוות מקור האור, בודקת האם הktr שחזר בין הנק שונה והאם לא הגענו למקסימום של עומק הרקורסיה. במידה ו2 התנאים אינם מתקיימים נכנס לתוך לולאת הif שבה מתבצע חישוב עבור קצוות האזור שעליהם נבצע את הפונקציה פעם נוספת:

```
if (!flag && depth > 0) // if not all ktr is same and depth > 0
{
    Point3D middleAC = new Point3D( x: 0.5 * (A.getX() + C.getX()),  y: 0.5 * (A.getY() + C.getY()),
            z: 0.5 * (A.getZ() + C.getZ()));  // middle point between A and C
    Point3D middleAB = new Point3D( x: 0.5 * (A.getX() + B.getX()),  y: 0.5 * (A.getY() + B.getY()),
            z: 0.5 * (A.getZ() + B.getZ()));  // middle point between A and B
    Point3D middleAD = new Point3D( x: 0.5 * (A.getX() + D.getX()),  y: 0.5 * (A.getY() + D.getY()),
            z: 0.5 * (A.getZ() + D.getZ()));  // middle point between A and D
    Point3D middleDC = new Point3D( x: 0.5 * (D.getX() + C.getX()),  y: 0.5 * (D.getY() + C.getY()),
            z: 0.5 * (D.getZ() + C.getZ()));  // middle point between D and C
    Point3D middleBC = new Point3D( x: 0.5 * (B.getX() + C.getX()),  y: 0.5 * (B.getY() + C.getY()),
            z: 0.5 * (B.getZ() + C.getZ()));  // middle point between B and C
```

ניתן לראות הבדל מהותי בטסט הבא שהרצנו לפני ואחרי השיפור:

הרצה לפני השיפור



הרצה לאחר השיפור