בס"ד

מגישים: דניאל הוניג
הראל יששכר

דו"ח פרויקט

רקע כללי לפרויקט:

מטרת הפרויקט הינה יצירת תוכנית לדימוי סצנה גרפית וירטואלית תלת-ממדית. כלל הפרויקט נכתב בשפת JAVA תוך שמירה עבודה לפי עקרונות עיצוב והנדסת תכנה הנלמדו בקורס התאורטי.
הפרויקט נכתב בליווי בדיקות (TESTING) לפונקציות העיקריות כולל בדיקות מחלקות שקילות(Equivalence Partitions Tests) ובדיקת מקרי קצה(Boundary Values Tests).

הפרויקט מורכב ממספר חבילות (PACKAGES) באמצעותם נממש את המודל ליצירת תמונה תלת ממדית להלן פירוט החבילות והמחלקות:

-PRIMITIVE

מכילה את המרכיבים הבסיסיים להגדרת מרחב R3.

COORDINATE- הגדרת קאורדינטה, נקודה באחד משלושת המישורים שבמרחב התלת ממדי.

POINT3D- נקודה במחרב מורכבת משלוש קאורדיניטות

VECTOR- ייצוג ווקטור במרחב, מיוצג על ידי נקודה POINT3D שמגדירה את ראש הווקטור ועל ידי כך את הכיוון שלו.

RAY- קרן, קרן הינה ווקטור שמוגדר בנוסף על ידי מיקום התחלה כלומר הכיוון של הקרן מיוצג על ידי וקטור ובנוסף קיימת נקודה POINT3D שמגדירה את המיקום ההתחלתי שלו.

COLOR- הגדרת צבע לכל נקודה במרחב כלומר אם ישנו גוף במרחב אזי הגוף מסוגל לפלוט צבע מסוים ולכן נגדיר צבע לגופים, תאורות, ובאופן כללי נוכל להגדיר צבע רקע לסצנה עצמה או להגדיר העדר צבע כלומר צבע שחור.

MATERIAL- חומר, נגדיר סוג חומר לגופים כלומר מה רמת התגובה של הגוף בפגיעת קרן אור מסוימת הגדרה זו תתבצע על ידי מקדמי הנחתה לאפקטים של החזרת אור, מקדמי שקיפות והשתקפות ומידת הנצנוץ של החומר.

-GEOMETRIES

מכילה את הצורות הבסיסיות שנוכל לשתול בסצנה.

כלל הצורות יורשות מהמחלקה האבסטרקטית GEOMETRY אשר מממשת את הממשק- Intersectable .

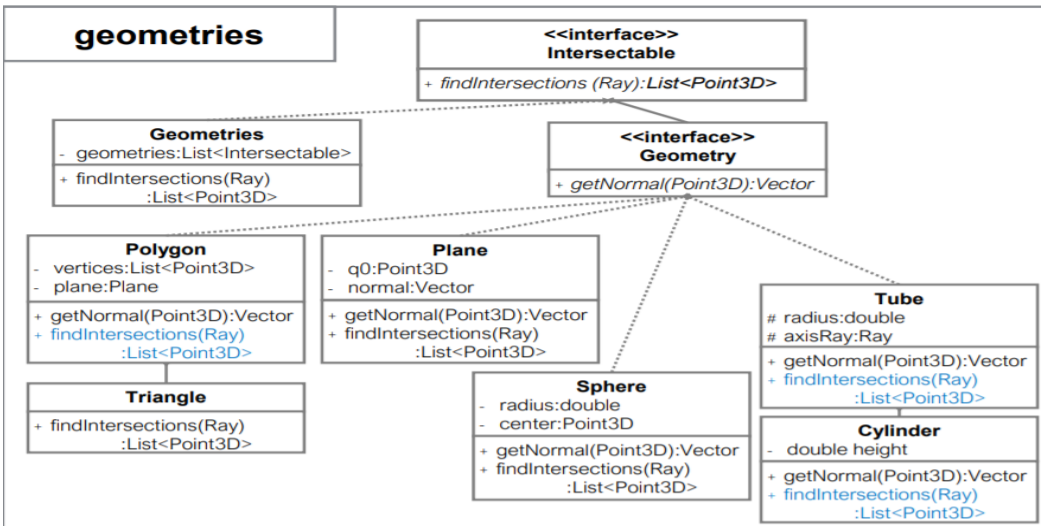
הפונקציה העיקרית שהמחלקות דורסות היא findGeoIntersections שמחזריה את נקודות החיתוך של הקרניים שנבנה מהמצלמה ושיעברו את המשטח (view plane) עם הגופים בסצנה.

בנוסף כל מחלקה שמייצגת גוף מסוים יודעת להחזיר את הוקטור הנורמלי(אנכי) לגוף זאת על מנת לחשב חיתוכים ואינטראקציות עם תאורה.

בס"ד

מגישים: דניאל הוניג
הראל יששכר

היררכית חבילת GEOMETRIES



-ELEMENTS

כפי שהוזכר לעיל אנו בונים סצנה גרפית על ידי משטח צפייה VIEW PLANE שמאחוריו ממוקמת מצלמה או לחילופין העין של המתבונן בסצנה.

נגדיר בחבילת זו את המחלקה CAMERA ובנוסף את כלל סוגי התאורה אשר נתייחס אליהם בפרויקט.

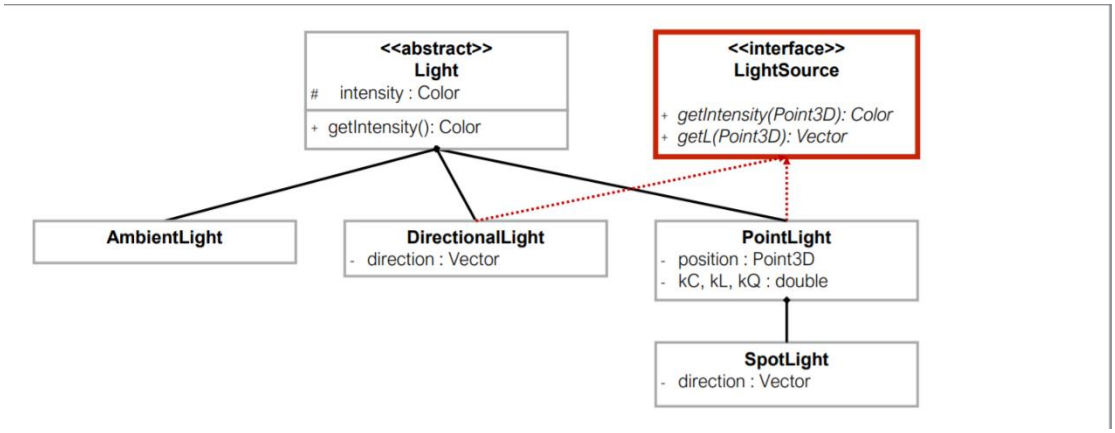
CAMERA- מורכבת מנקודה שמגדירה את המיקום של מרכז העדשה ובנוסף שלושה וקטורים – RIGHT ,TO,UP שמגדירים את כיוון העדשה ואת התזוזה של על הצירים.

הממשק LIGHTSOURCE –

כלל סוגי התאורה חוץ מתאורה סביבתית AMBIENT ממשמים ממשק זה הכולל את הפונקציות

getIntensity שמחשבת את עומת האור שמגיעה ממקור האור לנקודה וgetL שמחזירה את הוקטור ממקור האור לנקודה המוארת.

היררכית חבילת ELEMENTS



-RENDERER

חבילה זו מכילה את המחלקה העיקרית RENDER שמחזיקה את כל הנתונים הרלוונטיים לסצנה.

בנוסף קיימות המחלקות IMAGE WRITER שמייצרת קובץ JPG על ידי מטריצת צבעים שנבנית בתהליך הרנדור , RAY TRACER BASIC כלל החישובים להשגת צבע בכל נקודה בסצנה מתבצע במחלקה זו על ידי התחשבות באפקטים גלובליים , שקיפות והשתקפות (מראה) של חומרים ואפקטים לוקלים של תגובת החומר בפגיעת קרן אור והטלת הצל.

הפונקציה העיקרית CALC COLOR הינה פונקציה שתפעל באופן רקורסיבי ותבנה קרני שקיפות והשתקפות ובכך תדמה לנו אפקטים של מראה או חומר שקוף שהצבע עובר דרכו.

-UNITESTS חבילת הבדיקות ויצירת תמונות מבחן במהלך הפרויקט.

לאחר שהסברנו בקצרה על הפרויקט נפרט על שיפורי התמונה שביצענו למיני פרויקט 1:

מטרת מיני פרויקט 1: שיפור התמונה על ידי מתן מראה מציאותי יותר.

<u>SOFT SHADOW</u>

הבעיה: הצל בתמונות נראה לא אמיתי מכיוון שהוא נראה כצל קשה ובעל מרקם אחיד בכל נקודה שמוצלת על ידי הגוף

פתרון: במקום להתייחס למקור אור כנקודה אחת במרחב נתייחס למקור האור ככדור בעל מספר נקודות במרחב. כעת נשלח מספר קרני צל אל מספר אזורים שונים במקור האור ואחוז התאורה בכל נקודה יהיה היחס בין מספר הפגיעות במקור האור לכמות קרני הצל שיצרנו.

מימוש:

למקור האור הוספנו רדיוס שייצג את מקור האור כעיגול, הרדיוס יהיה מאוחסן בכל מחלקה של מקור תאורה למעט תאורה סביבתית AMBIENT LIGHT .

לאחר מכן כדי שנוכל למצוא נקודות בתוך העיגול נצטרך למצוא את המישור שבו העיגול מוכל ועל כן נמצא שני וקטורים מוכלים במישור ממש כמו שהגדרנו ב VIEW PLANE .

על ידי שני הווקטורים במישור נוכל לזוז במישור ולקבל כל פעם נקודה אחרת בתוך העיגול שמייצג את אזור מקור האור.

כעת בשליחת קרני צל נותר לסכום את מספר הקרניים שיצאו מהגוף ופגעו בנקודה ששייכת למקור התאורה האיזורי ולחלק במספר הקרניים הכולל ששלחנו כך נקבל צל יחסי (SOFT SHADOW) ולא צל מוחלט בעל אותו מרקם (HARD SHADOW) .

להלן הקוד למציאת נקודות בעיגול האיזורי של מקור אור:

```
Point3D xPoint;     // movement in the x axis
Point3D yPoint;     // movement in the y axis
//number of moves from the center to each side in the direction of vector X(or it opposite direction)
//if the are was a square then the formula was :square (mn Points) / 2
//since the are is a circle (area is about 80 percent) and our coverage is about 80-90
// total we get about 2/3 of the original wanted amount
// there we multiple the previews formula by 1.25 times (square is 1.5 bigger. after 2/3 reduce we
// will get the wanted amount)
int PARTITION = (int) (Math.sqrt(minPoints) / 2 * 1.25);
double distance = _radius / PARTITION;
for (int i = -PARTITION; i <= PARTITION; i++) {
    if (Util.alignZero( number: i * distance) != 0) {
        xPoint = _position.add(x.scale(i * distance));
    } else {
        xPoint = _position;
    }
    double maxY = Math.sqrt((_radius * _radius) - (i * distance) * (i * distance));
    int moves = (int) (maxY / distance);
    for (int j = -moves; j <= moves; j++) {
        if (Util.alignZero( number: j * distance) != 0) {
            pointsInCircle.add(xPoint.add(y.scale(j * distance)));
        }
    }

}
return pointsInCircle;
```

סצנה למיני פרויקט 1:

יצירת סצנה של חדר המורכב מ6 משטחים (מישורים). בתוך החדר שולחן המורכב מ22 גאומטריות. ניתן לראות בתמונה את הצל של השולחן לפני ואחרי השיפור.

על השולחן מונחת פירמידה ובתוכה עוד פירמידה המורכבות כל אחת מ4 משולשים. הפירמידה החיצונית שקופה ולכן ניתן לראות דרכה את הפירמידה הפנימית.

ארון המורכב מ14 גאומטריות שני הדלתות השמאליות הינם בעלי מקדם השתקפות 1 ולכן יוצרות מראה דרכה אפשר לראות את השולחן והפירמידות.

שלושה כדורים על הקיר הימני בהם ניתן לראות את תגובת האור הספקולרי והדפיוסי.

כמו כן בסצנה שלושה מקורות אור:

- ספוט לייט שנמצא מעל השולחן וכיוונו אל השולחן.
- ספוט לייט מכוון אל הכדור האמצעי
- פוינט לייט מול הארון שניתן לראות את ההשתקפות שלו בדלת הימנית.

בס"ד

מגישים: דניאל הוניג
הראל יששכר

ניתן לראות את ההבדלים לאחר השיפור SOFT SHADOW בעיקר בצל המופק על ידי השולחן והארון.

תמונה לפני השיפור:

תמונה לאחר השיפור:

בס"ד

מגישים: דניאל הוניג
הראל יששכר

<u>שיפור שני – super sampling</u>

הבעיה: באזורים בהם יש שינוי דרמטי בצבע של הפיקסלים התמונה נראית מגורענת.

פתרון: במקום לבדוק כל פיקסל ע"י קרן אחת נבנה מספר רב של קרניים ונעשה ממוצע של הקרניים לבניית הצבע של הפיקסל. כך נקודות המעבר בין גופים בצבעים שונים יהיו בצבעים 'בין לבין' ומבחינה ויזואלית הגרעיניות תפחת בצורה משמעותית מאד.

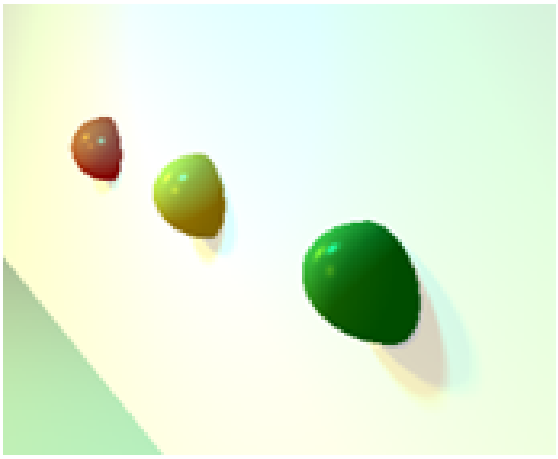
מימוש:

בבנית הקרניים בCamera , במקום ליצור קרן אחת למרכז הפיקסל כעת ניצור מספר רב של קרניים מפוזרים בצורה אחידה בפיקסל. לצורך כך כל שיש לעשות הוא לבנות רשת (grid) בפיקסל וליצור קרניים מהמצלמה לכל אחד מהנקודות הנ"ל. יצירת הרשת פשוטה ונעשית באופן דומה ליצירת כל הקרניים בcamera.

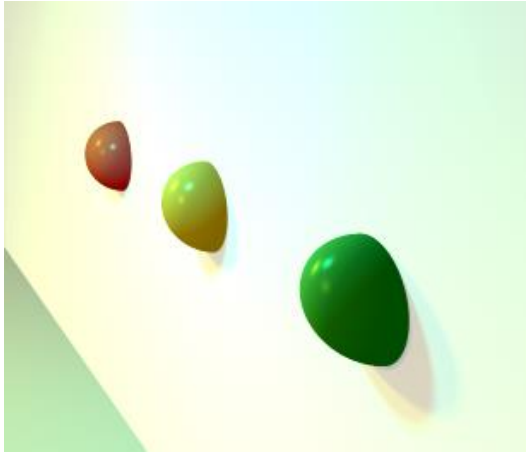
התמונה הסופית כולל את השיפור הנ"ל בנוסף לשיפור של soft shadow:

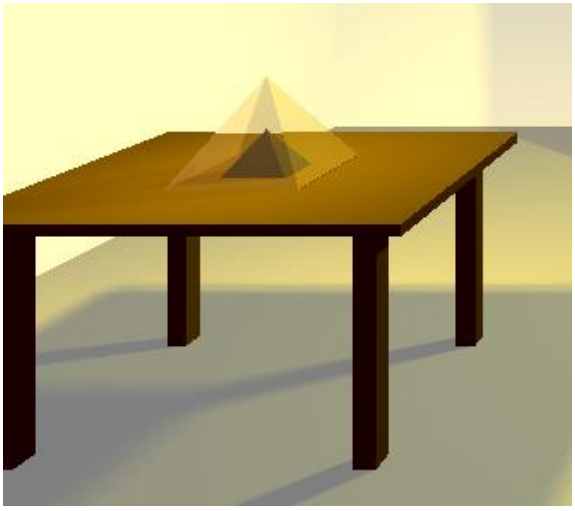
(ניתן לראות שהגרעיניות בכדורים כמעט נעלמה)

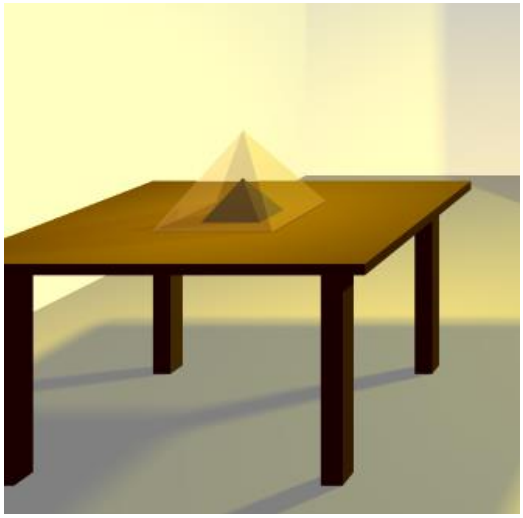
לפני adaptive multi sampling:



אחרי adaptive multi sampling:







<u>שיפור ביצועים – adaptive super sampling:</u>

מכיוון שברוב הפיקסלים בתמונה אין תנודות חדות בצבע ולכן אין בהם גרעיניות, מיותר לייצר גם בהם כמות גדולה של קרניים ולבדוק אותה לחינם.

לכן, במקום לייצר לדוגמה 81 קרניים לכל פיקסל נייצר בהתחלה 4 ונבדוק אם הצבע בהם דומה – כלומר אין מעבר בין גופים וצבעים שונים בפיקסל, אז אין צורך לייצר קרניים נוספות. ואם יש שוני נחלק את הפיקסל ל4 בצורה רקורסיבית וננסה לבדוק אם אפשר לצבוע כל רבע וכן באופן רקורסיבי עד למקסימום כלשהוא(חלוקה ל17*17 או 9*9 פיקסלים).

מהרצה של התמונה שלנו:

בלי super sampling: כ15 דקות

הערכה ל super sampling של 81 קרניים (9*9): כ20 שעות. (81 כפול רבע שעה)

זמן ריצה עם adaptive super sampling: כשעה ו5 דקות.

שיפור ביצועים של פי 18 לערך.

יש לציין שאופן הפעולה הרקורסיבי של עלול לגרום לכך שנחשב ray מספר פעמים. כמו כן בחישוב קרני המצלמה קרניים שבקצה של פיקסל מסויים הם גם קרניים של קצה של פיקסל אחר.

לכן יצרנו מחלקה חדשה ColorRay שמחזיקה Ray ו Color ביחד (מעין map, בדומה לרעיון של geoPoint). ולכן בכל פעם שנשתמש בקרן שכבר השתמשנו בה לא נצטרך לבדוק את הצבע שלה כי ברגע שחשבנו אותו בפעם הראשונה הוא כבר נשמר.

שיפור זה לבד חסך כ-50% זמן ריצה.