**דו''ח סופי**

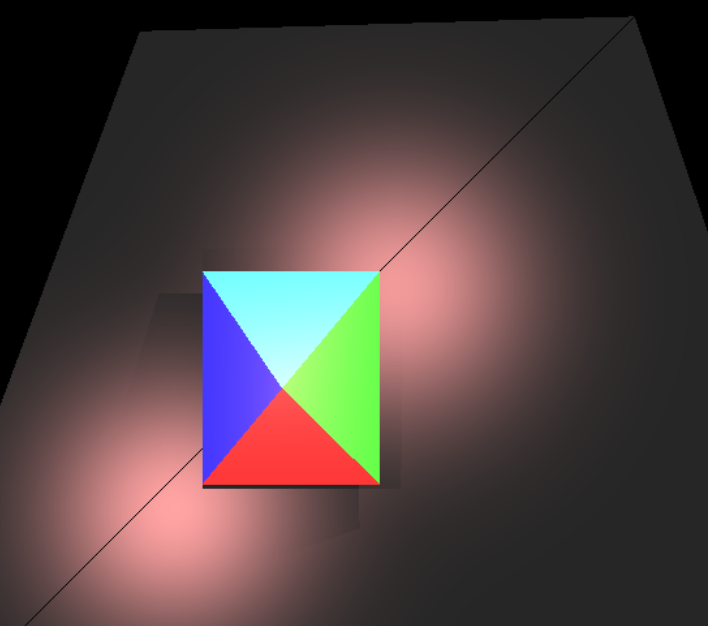
**מגישים:**

**יונתן טמיר**

**איתמר אוחנה**

* יצירת גיאומטריה חדשה-Pyramid:

1. כולל המתודות של findIntersections,getNormal (בדיקת חיתוך של פאה מסוימת).
2. קבלת Emission ו Material לכל פאה בנפרד או לכולם ביחד(עיצוב כיד המלך).
3. מורכב בפשטות על ידי בנאי של בסיס (4 נקודות) ונקודה של פסגת הפרמידה.



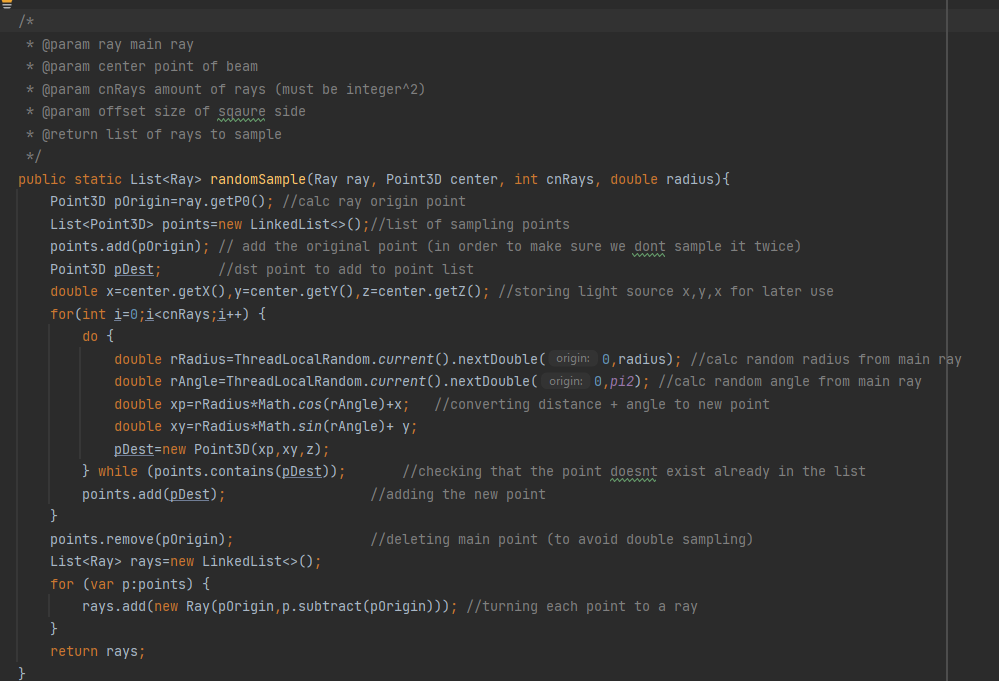
בתמונה רואים פירמידה שמורכבת מ4 משולשים ובסיס עם שני גופי תיאורה מצידיה,וemission שונה לכל פאה.

* **Super Sampling :**

1. **יצירת אלומות בצורה רנדומלית.**

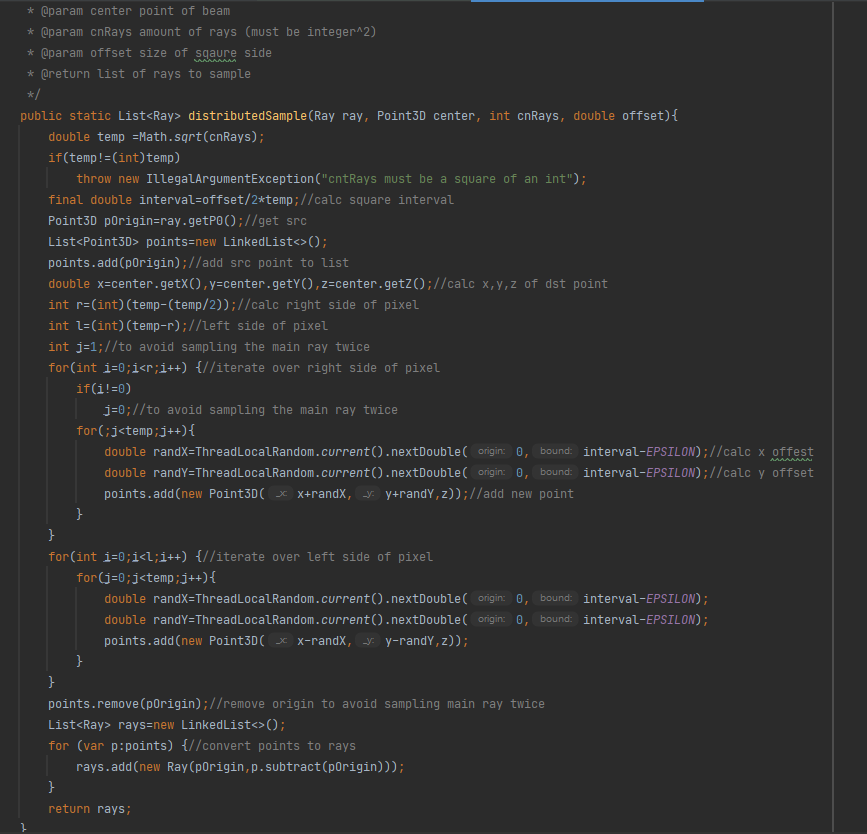
הקוד יוצר נקודות רנדומליות על ידי רדיוס ומעגל ויוצר מהם בעזרת הקרן המרכזית ונקודת מקור האור אלומת(רשימה) קרניים ומחזיר אותה.

תמונה של הקוד בעמוד הבא:



1. **יצירת אלומות על ידי התפלגות אחידה.**

הקוד בונה מרובע מסביב לנקודה ומחלק אותו לגריד על פי כמות הקרניים כאשר בתוך המשבצת המיקום של הקרן רנדומלי



* הוספת setter בשביל הטסטים לפרמטרים המוספים:

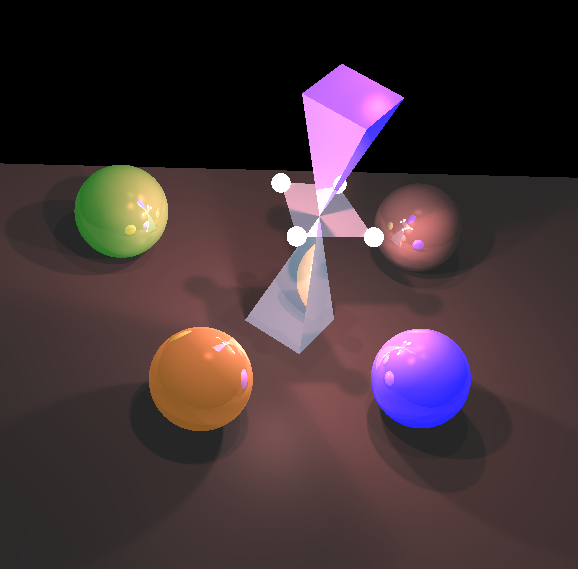
1. SoftShadows
2. AntiAliasing
3. כמות הקרניים,רדיוס הדגימה וכו'...

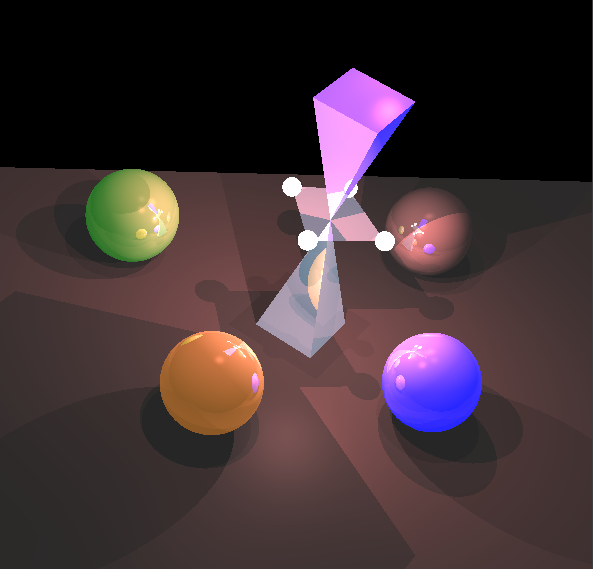
**שיפור תמונה**

**Soft Shadows :**

1. שיפור הצל של הגיאומטריות לריאליות חדשה (הצל מתמשך).
2. על ידי שימוש באלגוריתמים שונים המוצעים במחלקה SuperSampling.

יצירת סצנה מרובת גיאומטריות (2 פירמידות,8 ספירות,2 מצולעים) ותאורות.

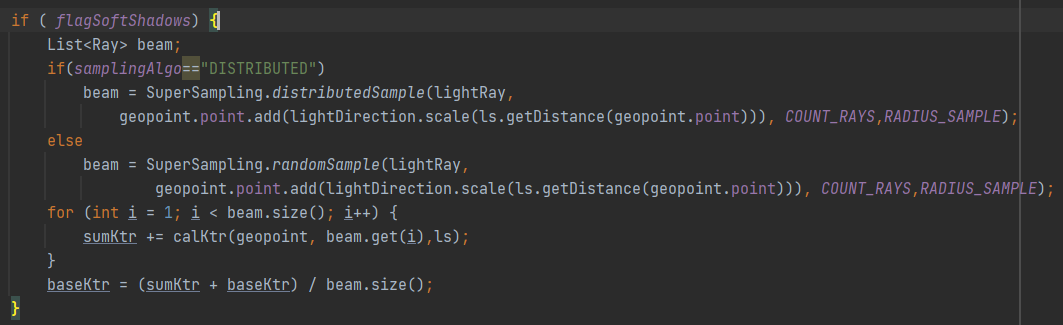
לפני: אחרי:



אחרי:

בתמונה מימין רואים איך הצל לא 'נמשך' אלא חד וחלק שלא כמו במציאות לעומת התמונה משמאל ששם הצל נמשך לאחר ה Soft Shadows.

בקוד רואים בדיקה אם הflag של ה softShadows דלוק ואם כן יש בדיקה על פי איזה אלגוריתם לממש אותו ואחרי שנוצרת האלומה חישוב ה ktr של כל קרן בה.

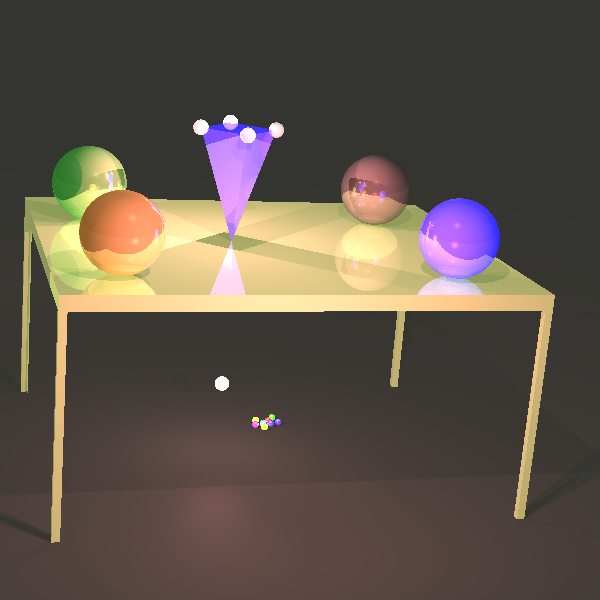


* הוספת תהליכונים:

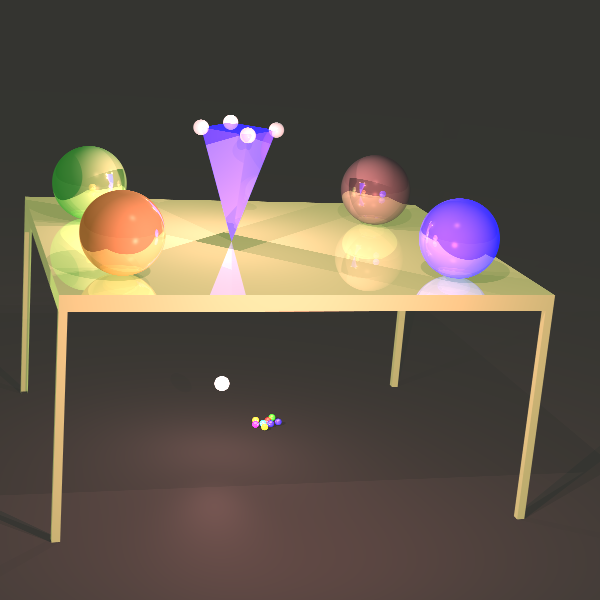
1. שינוי מחלקת הרינדור לתמיכה בתהליכונים.
2. הוספת setter לשינוי כמות ההליכונים בעת ההרצה.

* Adaptive Super Sampling :
* Anti aliasing - שיפור חדות התמונה על ידי טשטוש קצוות הגיאומטריות בהתאם לסביבתן כל זאת על מנת לספק לצופה חוויה אסתטית הרמטית תוך שזירת ההיגיון הסביבתי שבבסיס אסתטיקת היחסים בין הגופים ומקורות האור השונים.
* שיפור ביצועים על ידי חלוקת הפיקסל לתתי פיקסלים באופן רקורסיבי.
* Setter לבחירת שימוש בתכונה זו או לא.

לפני:

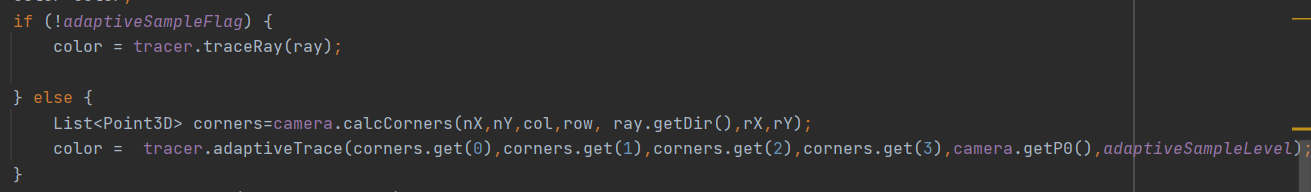


אחרי:

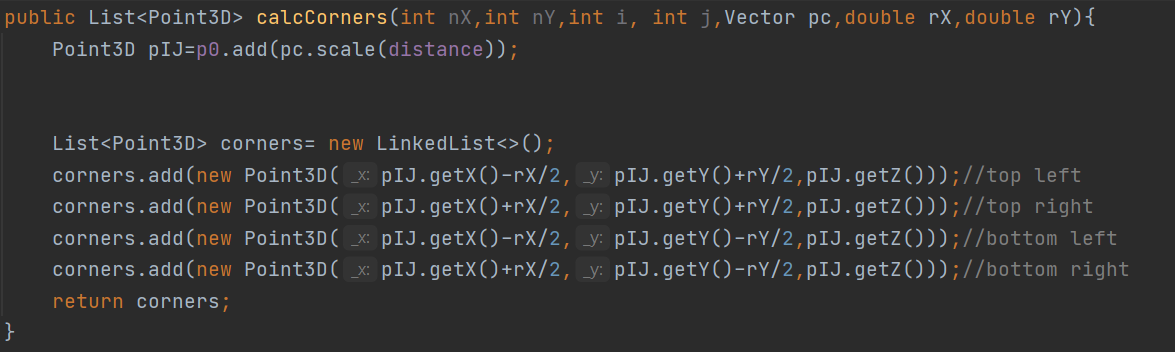


בתמונה הראשונה ניתן לראות את הגבול בבירור בין שני אובייקטים ואיך הפיקסלים משנים צבע בצורה ישירה, לעומת זאת בתמונה השנייה הפיקסלים מתאימים עצמם בהדרגתיות לצבעים והאובייקטים הסובבים אותם.

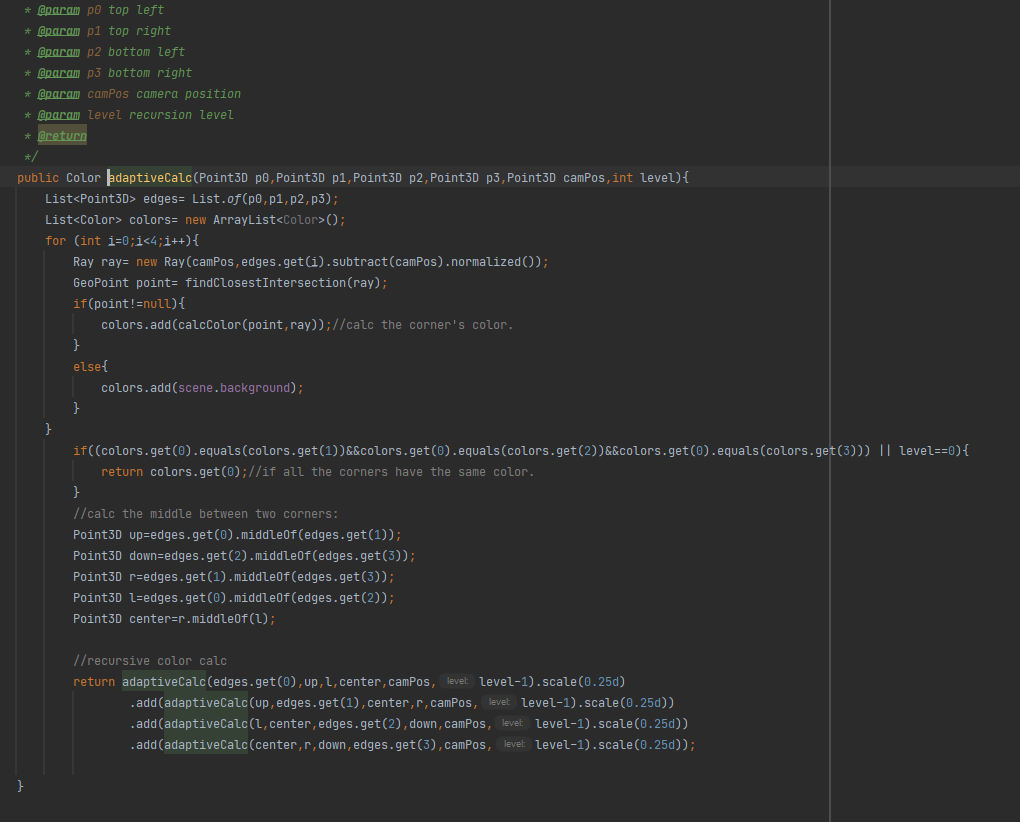
בקוד רואים בדיקה של הדגל של ה adaptive super sampling ואם הוא דולק אז מתבצע חישוב של הפינות של הפיקסל ושליחתם לפונקציה adaptiveTrace



חישוב פינות הפיקסל



הקוד מבצע חישוב רקורסיבי של אותם פינות עד הlevel שהוגדר או עד שכל הפינות באותו צבע



**היררכייה:**

* **האלגוריתמים-random sample ו distributed sample:**

algo->SuperSampling

שימוש מתבצע בפונקציה:

renderer->BasicRayTracer->transperency

* **בדיקת דגל ה adaptiveSuperSampling וקריאה לפונקציות החישוב:**

renderer->Render

* **הפונקציה calcCorners (חישוב הפינות):**

elements->Camera

שימוש מתבצע בפונקציה:

renderer->Render->castRay

* **חישוב הצבע על ידי הפונקציה adaptiveTrace (שקוראת לadaptiveCalc):**

renderer->BasicRayTracer

שימוש מתבצע בפונקציה:

renderer->Render->castRay