דו"ח פרויקט

* כל הפונקציות מחזירות פרמטרים חדשים תמיד – כולל הפונקציות של חיבור/חיסור, או הכפלה וקטורית (למעט נירמול שמתבצע על האיבר עצמו).
* יצרנו getters רק כשהיה חייב.
* יצרנו setters בתבנית עיצוב builder .
* הסברנו בכל מחלקה, שדה ופונקציה מה תפקידם בפורמט java doc.
* שיפורים:

- glossy surface -שיפור מראה משטח ההברקה.

- diffuse glass- שיפור מראה משטח שקוף

-בונוס- soft shadow (5 נקודות)-צללים רכים ולא חדים.

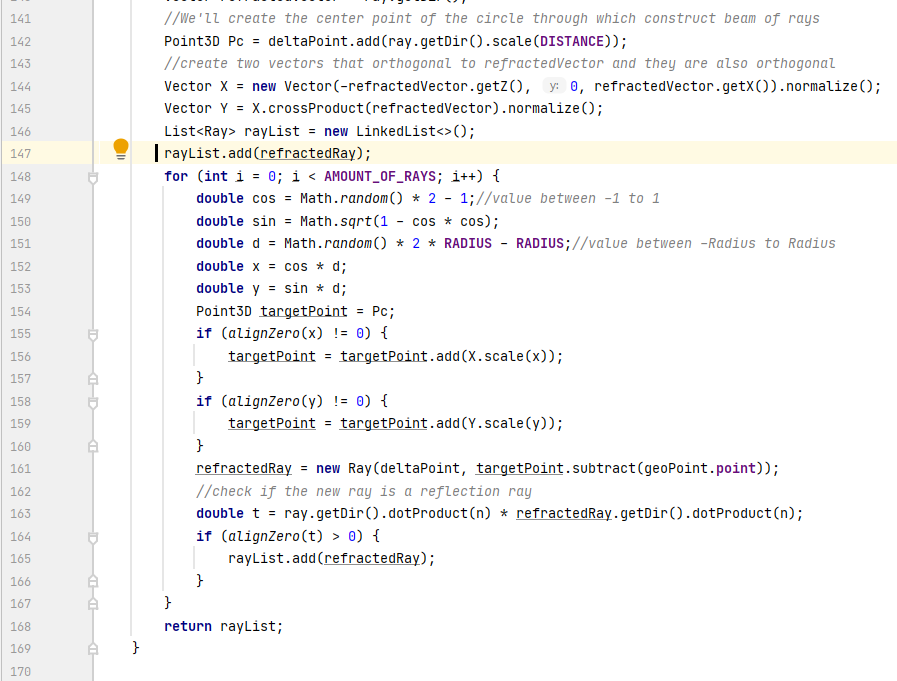
**הסבר על שיפורי התמונה**

יצרנו מחלקות rayTracer ייעודיות עבור כל שיפור אשר יורשות מ BasicRayTracer, ודורסות את הפעולות הנדרשות.

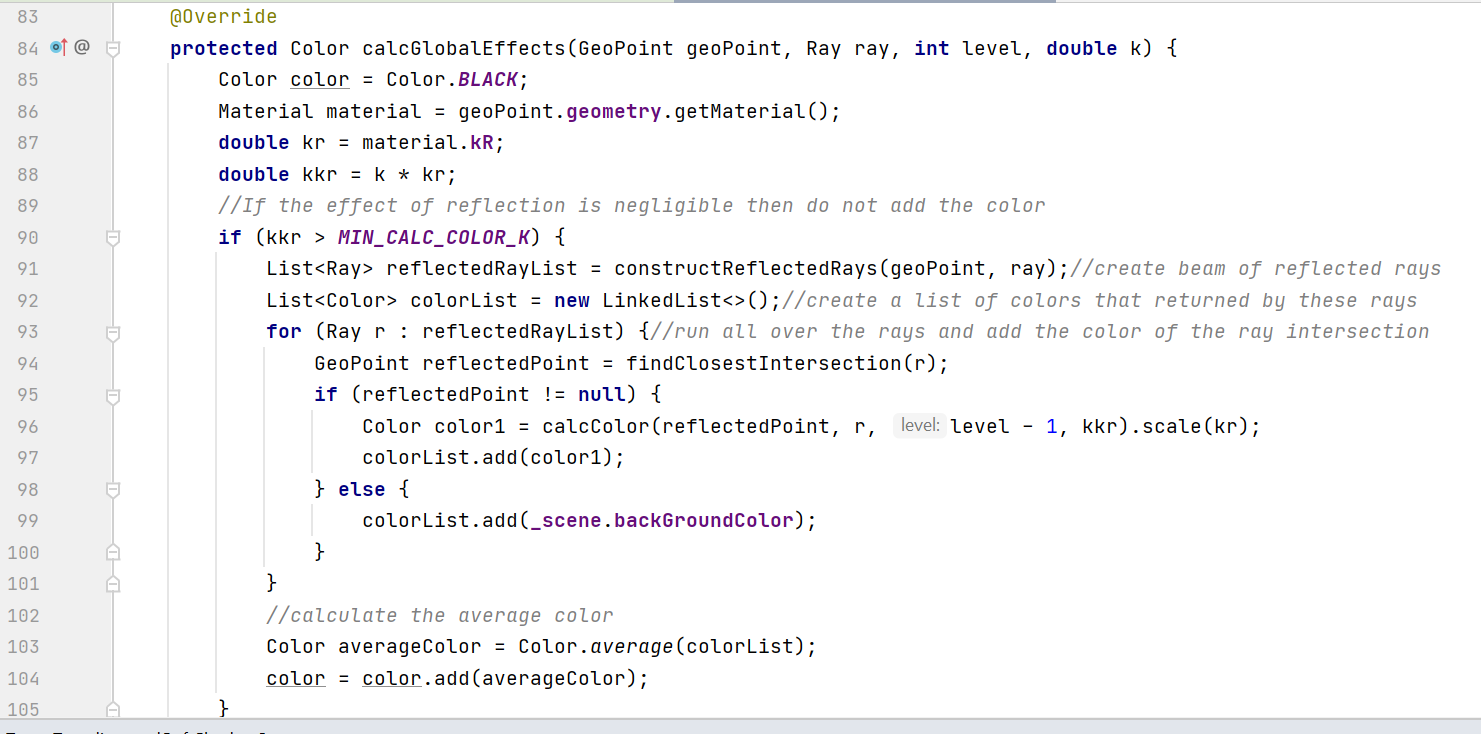
**glossy surface**

הבעיה הייתה שהשתקפות הגופים הייתה מלאכותית, כלומר – זה לא מציאותי שהגוף המשתקף נראה בדיוק כמו הגוף המקורי.

תיקון: שינינו את הפונקציה constructReflectedRay שבמקום לשלוח קרן השתקפות אחת עבור כל נקודה, שלחנו אלומה של קרניים בכמות שניתנת לבחירת המשתמש באמצעות setter.

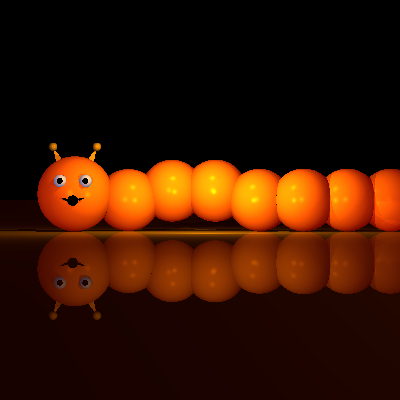
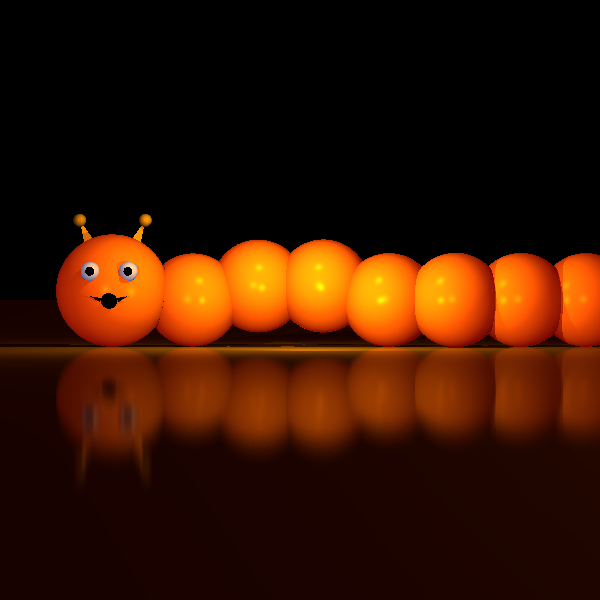
אופן יצירת הקרניים הוא יצירת עיגול מטרה וירטואלי, שדרכו נורות הקרניים בצורה רנדומלית כפי שמתואר בקוד הבא .

בנוסף דרסנו את הפונקציה calcGlobalEffect שתקבל את אוסף קרני ההשתקפות ותחשב את ממוצע הצבעים של כל קרן.



תמונה אחרי השיפור

תמונה לפני השיפור



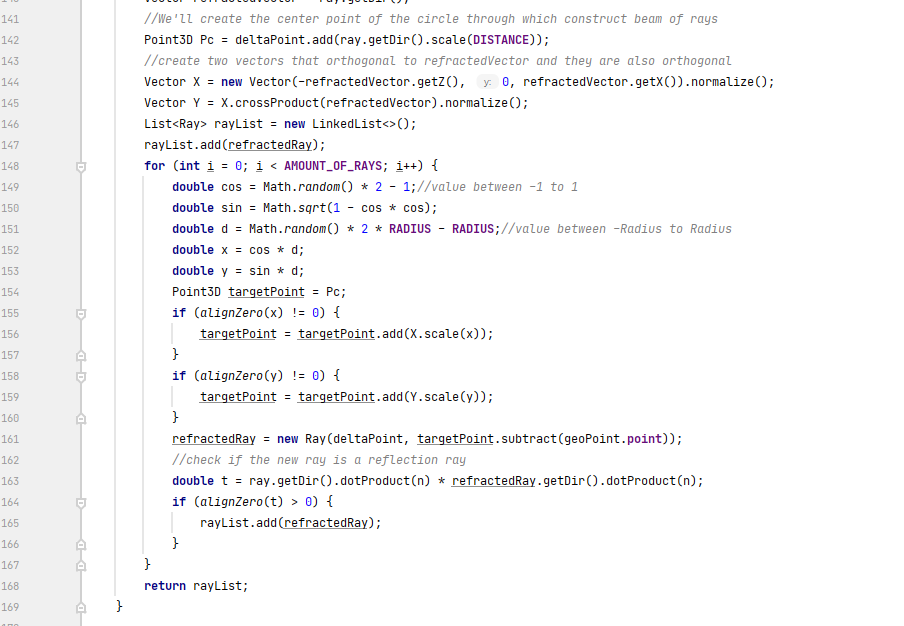
**diffuse glass** –

הבעיה: כאשר נותנים לגוף מראה שקוף (לא אטום) אז הגופים שמאחור נראים טבעי, מה שלא

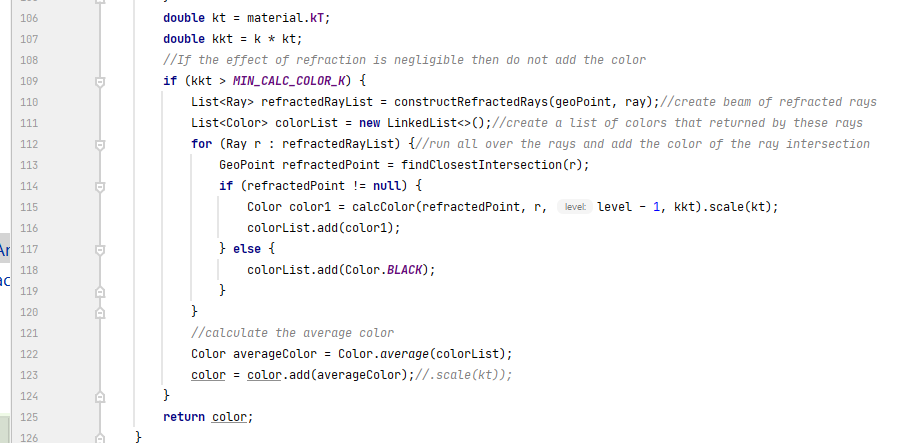
אמור להראות כך, אלא מעט מטושטש .

התיקון: שינינו את הפונקציה constructRefractedRay שבמקום לשלוח קרן השתקפות אחת עבור כל נקודה, שלחנו אלומה של קרניים בכמות שניתנת לבחירת המשתמש באמצעות setter.

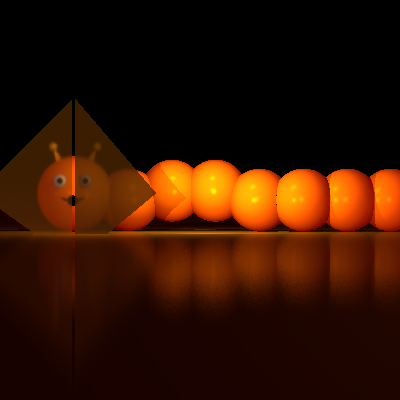
אופן יצירת הקרניים הוא יצירת עיגול מטרה וירטואלי, שדרכו נורות הקרניים בצורה רנדומלית כפי שמתואר בקוד הבא .

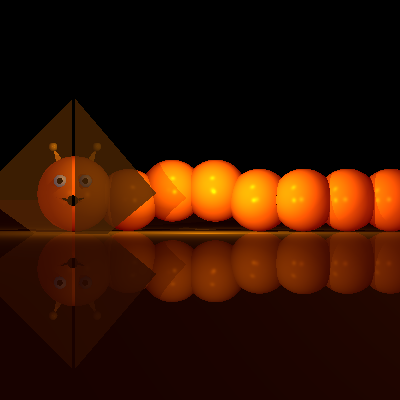


בנוסף שינינו את הפונקציה calcGlobalEffect שתקבל את אוסף קרני השקיפות ותחשב את ממוצע הצבעים של כל קרן.



תמונה אחרי השיפור





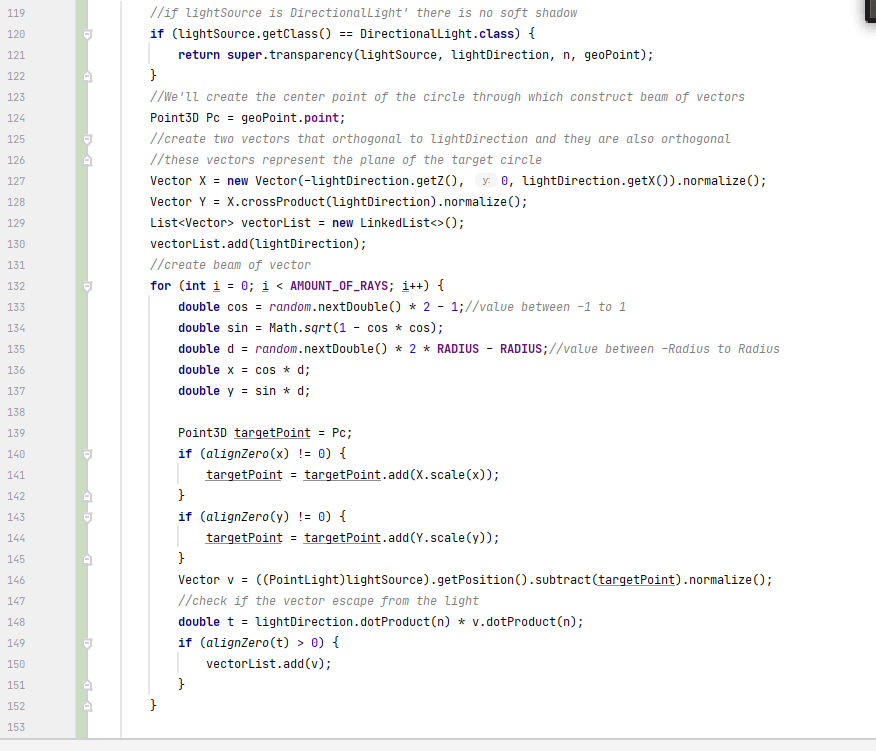
תמונה לפני השיפור

**Soft shadow:**

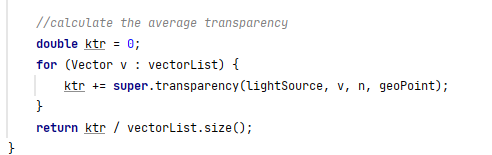
הבעיה: כאשר משתמשים ב- PointLight ו- spotlight , הצללים נראים חדים ולא מציאותיים.

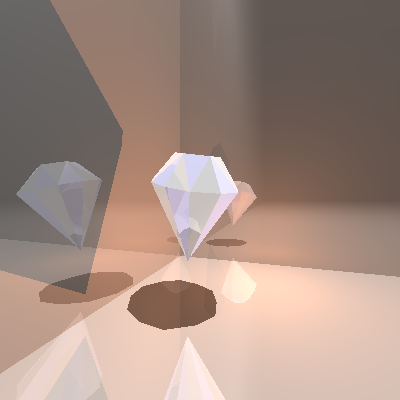
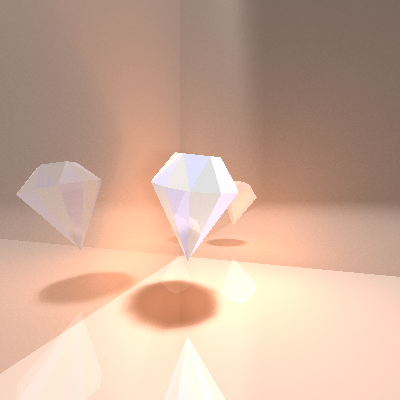
(בDirectionalLight הצללים כן אמורים להיות חדים)

התיקון: דרסנו את הפונקציה transparency שתייצר אלומה של וקטורים מהנקודת אור כלפי עיגול מטרה וירטואלי סביב הנקודה שעבורה מחשבים את הצל כפי שמתואר בקוד הבא.



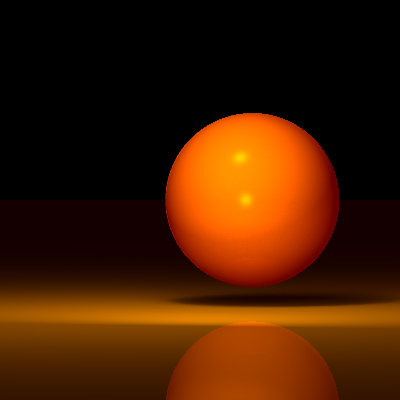
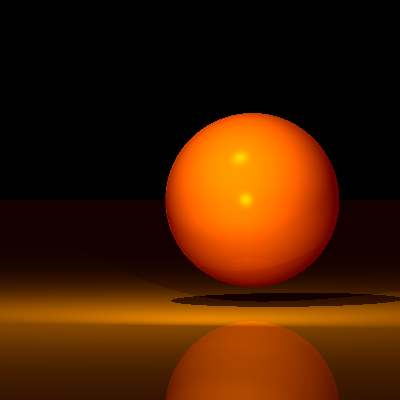
אח"כ קראנו לפונקציה transparency של מחלקת האב עבור כל וקטור, ואז חישבנו את ממוצע מקדם השקיפות.





תמונה אחרי השיפור

תמונה לפני השיפור



תמונה אחרי השיפור

תמונה לפני השיפור

התמונה לאחר השיפור נראית יותר מוארת מכיוון שכל המישורים שבתמונה הם מראות, וככל שהצל רך יותר, יש יותר אור שמוחזר.

בתמונות שלמטה אין תוספת אור לאחר השיפור