**דו"ח מיני פרויקט מבוא להנדסת תוכנה**

במיני פרויקט נדרשנו ליצור תמונה שתשקף את כל מה שלמדנו במשך הקורס.

החלטנו ליצור תמונה בסיסית עם בית, שמיים עם שמש יפה בצידה ואחר כך הוספנו עוד כל מיני אלמנטים על מנת ליצור תמונה יפה וברורה יותר.

זוהי התמונה הסופית:

תמונה שמכילה צעצוע, גרפיקה וקטורית

התיאור נוצר באופן אוטומטי

בתמונה ישנם כמה מקורות אור: קרן אור שמאירה על השמים ויוצרת את האפקט של ההילה של השמש, מנורה בתוך הבית שמאירה על החיה החמודה שבתוכה, אבל נתקעת בגוף ולכן יוצרת רק מעט אור שרואים.

שלבי הכנת התמונה:

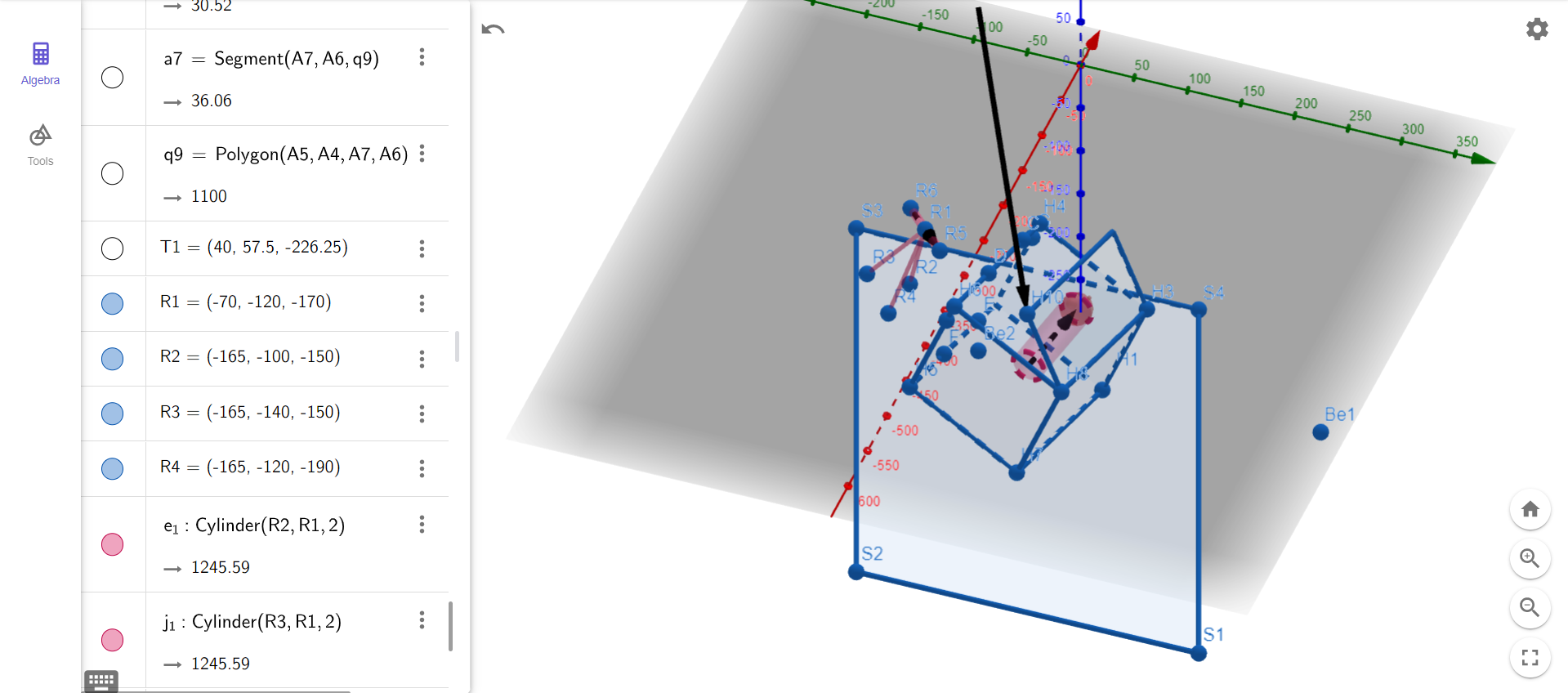
* שרטוט בגאוגברה כל שלב , כל נקודה, כל משטח וכו' על מנת שנוכל להרכיב את התוצר הסופי.

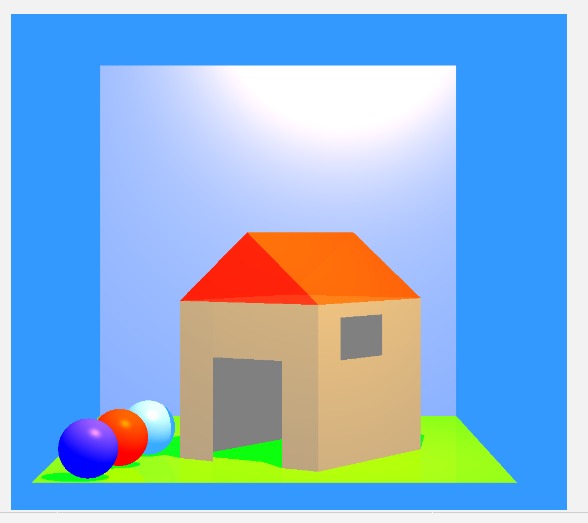
<https://www.geogebra.org/3d/nudvxtnt>

זהו קישור לגאוגברה של התמונה.

* בהתחלה בנינו רק את הרצפה הירוקה ואת המשטח שמאחוריה.
* התחלנו לבנות את הבית ע"י מספר רב של פוליגונים, בתחילה ללא חלון ודלת ואח"כ הוספנו.
* הוספת הכדורים שעל המשטח.
* הוספת שאר האלמנטים על מנת שהתמונה לא תהיה ריקה ויהיה בה מספר רב של חיתוכים עם הקרניים ככל האפשר.

דוגמא מתוך הגאוגברה:





**מיני פרויקט שלב 8:**

החלטנו לעשות את השיפור של ה anti aliasingמכיוון שבתמונה שלנו יש הרבה גופים והקצוות שלהם אינם נראים חדים. עושים זאת ע"י כך ששולחים הרבה קרניים דרך הפיקסל במקום קרן אחת, ואז את הצבע של הפיקסל מחשבים ע"י ממוצע של הצבעים של כל הקרניים.

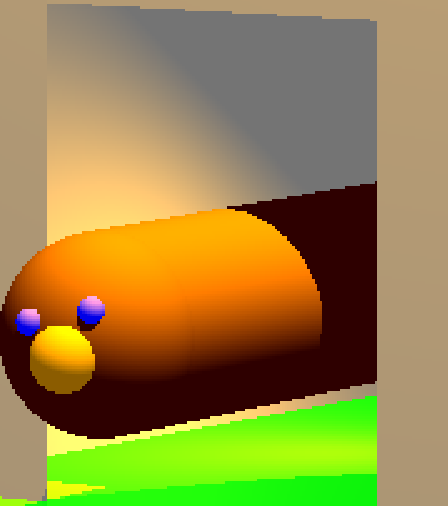
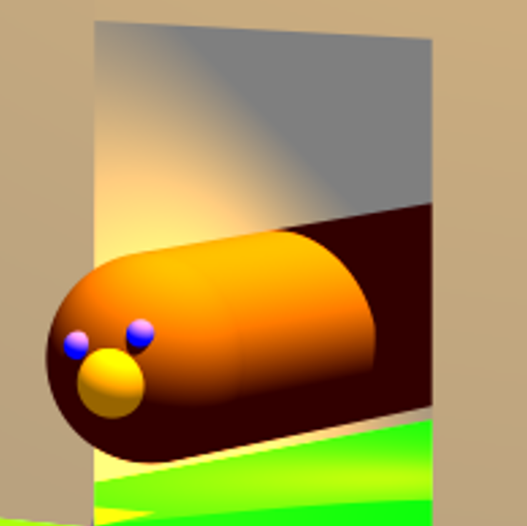
להלן הקוד שמראה זאת:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

ניתן לראות שהפונקציה מחזירה רשימה של כל הקרניים במקום קרן בודדת.

התמונה לפני ואחרי השיפור עם מספר קרניים השווה ל10:



ההרצה ללא השיפור לקחה כ5 דקות, ולאחר השיפור זמן רב יותר, ולכן היינו צריכות בשלב 9 של המיני פרויקט לשפר את זמן הריצה.

**מיני פרויקט שלב 9:**

הצטרכנו לשפר את זמן הריצה של התמונה ע"י 2 שיפורים:

* תהליכונים multi threading :

במקום להריץ את כל התמונה בתהליך אחד השתמשנו בתהליכונים שרצים במקביל וזה שיפר משמעותית את זמן הריצה ע"י שימוש ב4 תהליכונים, כפי שניתן לראות בצילומי המסך הבאים:

לפני השיפור של ה anti aliasing עם שימוש בתהליכונים :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

לאחר השיפור של ה anti aliasing עם שימוש בתהליכונים :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

* Adaptive super sampling:

הרעיון הוא לחלק כל פיקסל לאזורים ולפי זה לקבוע את הצבע של הפיקסל.

בהתחלה בודקים את הצבע של הקרניים שבפינות ובאמצע של כל אזור, אם הצבע זהה מחזירים את הצבע הזה, ואם לא נכנסים ברקורסיה לאותה פונקציה אבל עם אזור קטן יותר. כך חוסכים בדיקות מיותרות של הרבה קרניים, במיוחד באזורים שבהם יש צבע אחד על פני שטח רב יותר.

זמן הריצה לאחר השיפור של adaptive super sampling ללא multi threading :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

זמן הריצה לאחר 2 השיפורים:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי