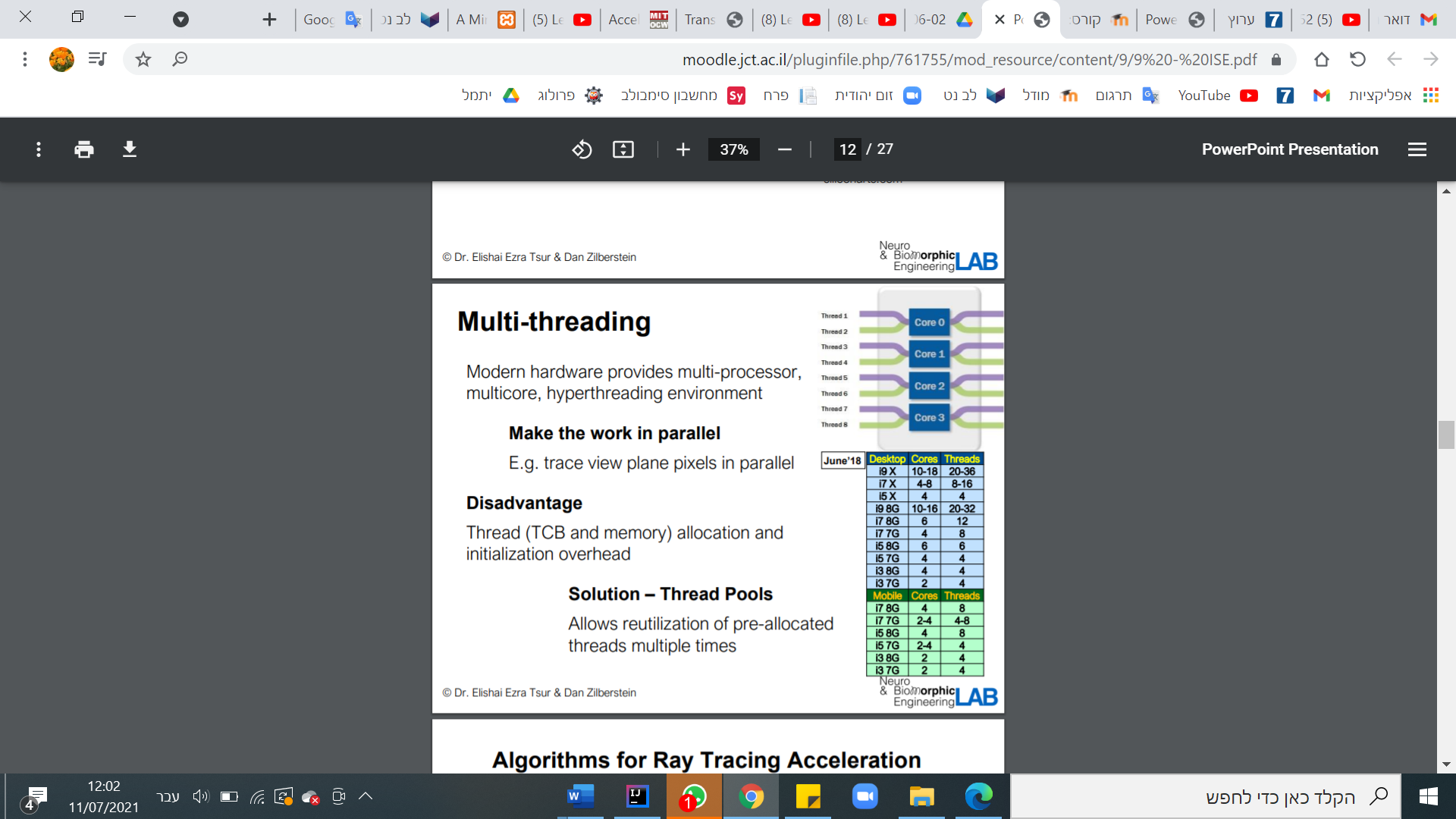
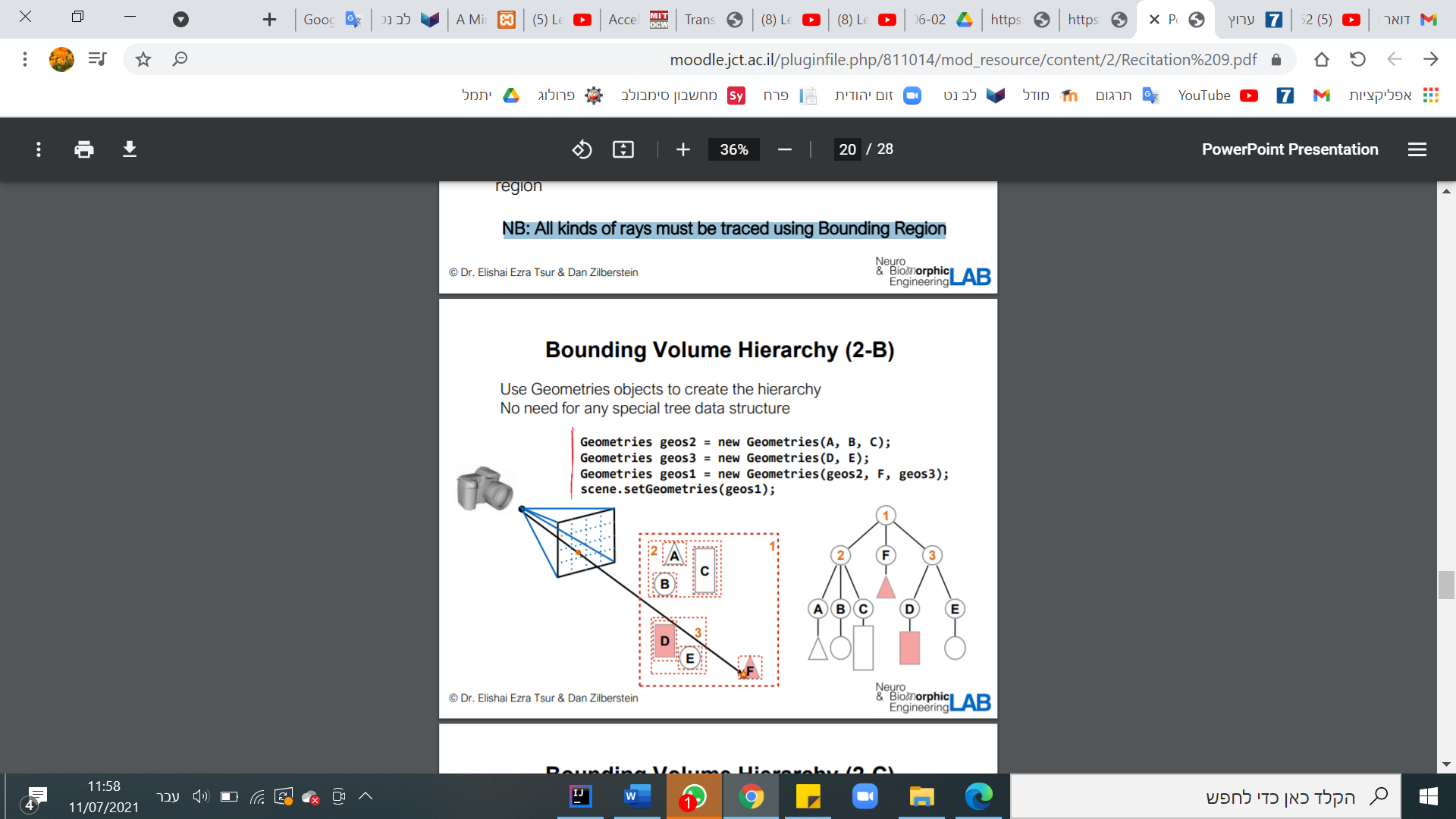
**דו"ח מיני-פרוייקט #2**

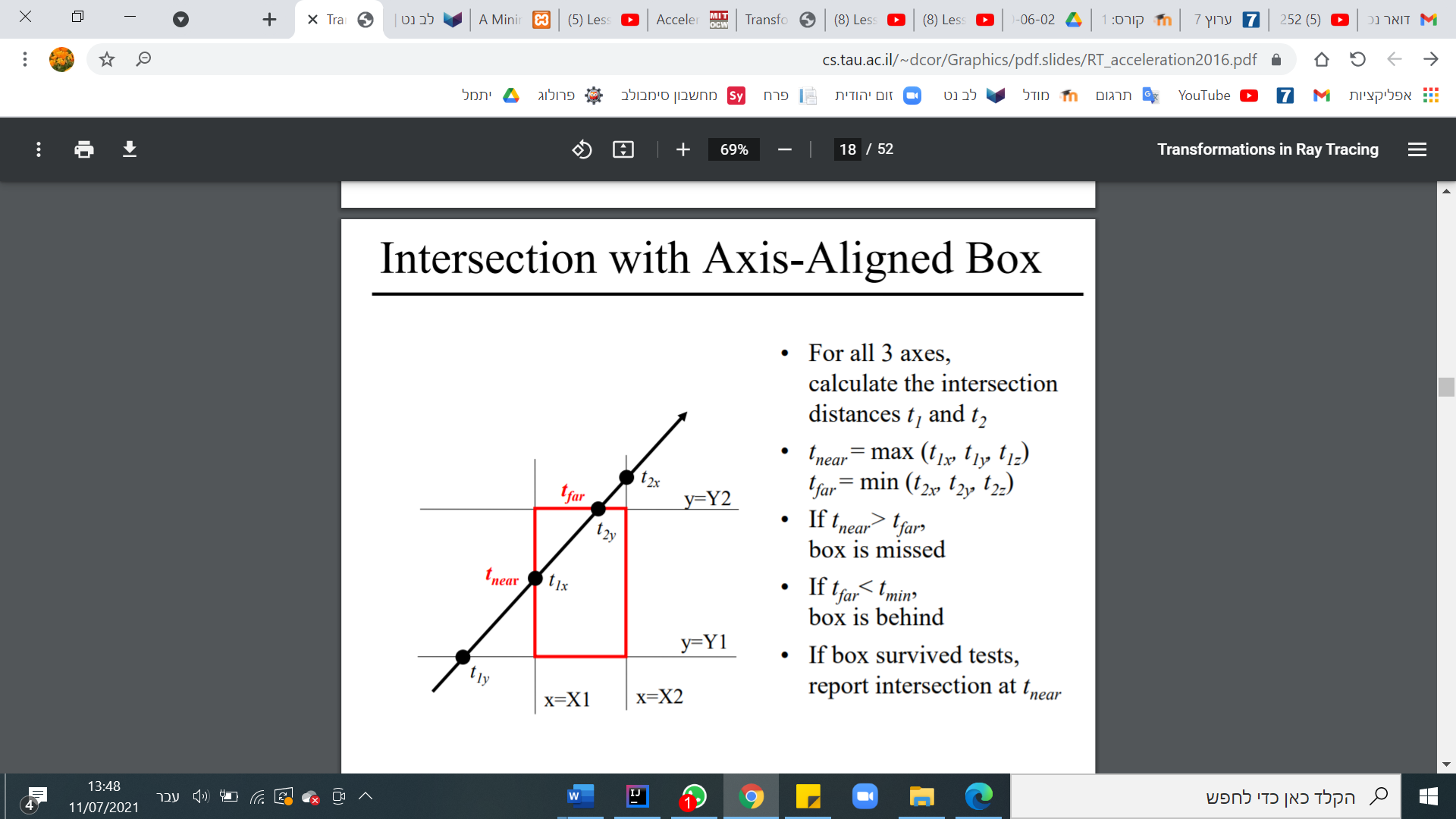
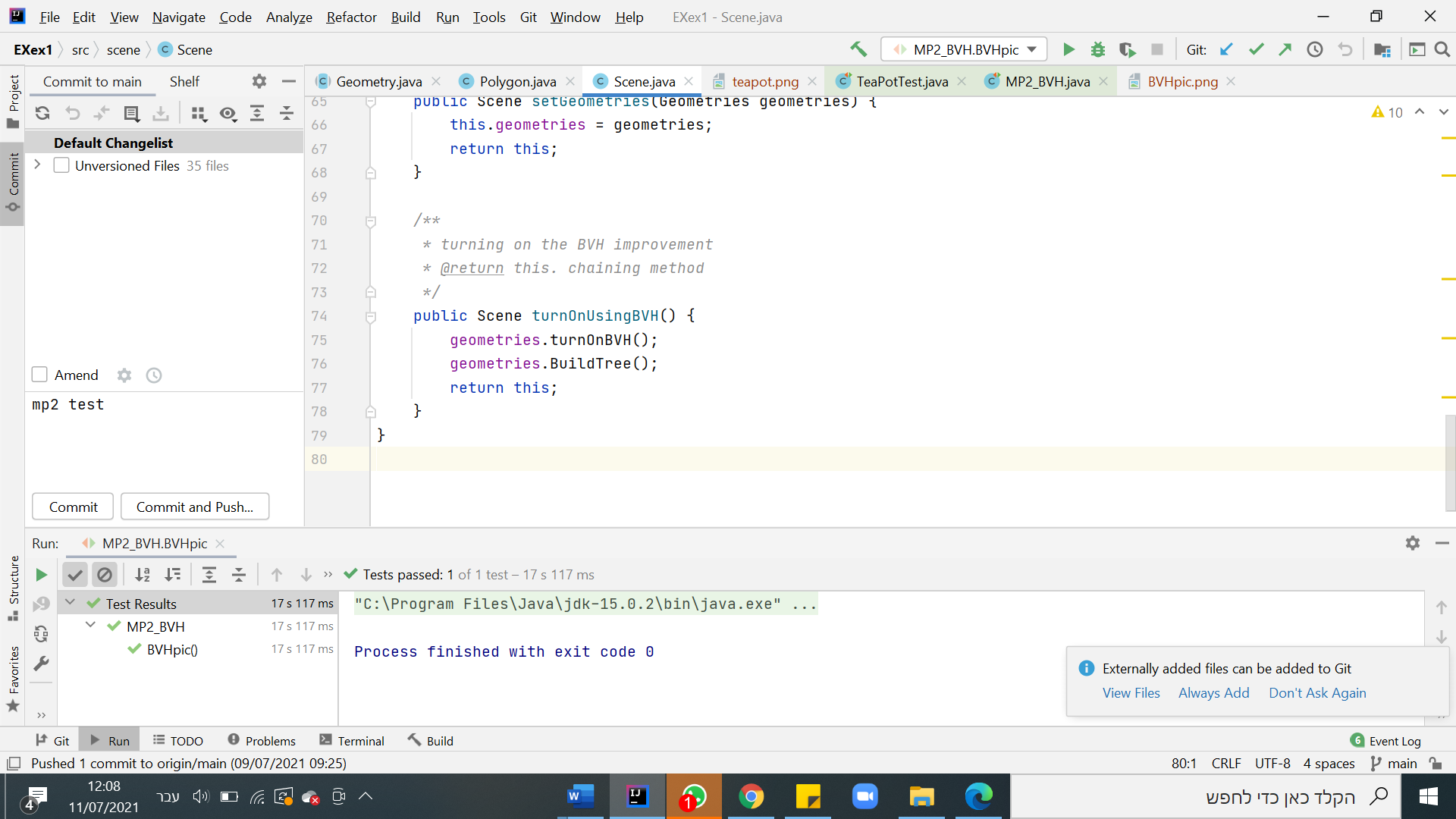
יהודית גוטמן 207812389 ושירה אורבך 315234484

בשלב זה ערכנו שיפור בזמן ריצת התוכנית עד ליצירת התמונה, ע"י שימוש בתהליכונים לריצה מקבילית- "Multi-threading", ובנוסף- Conservative Bounding Region: תחימת כל גוף ב'קופסא' לפי גבולותיו ומימוש אלגוריתם BVH החוסך שליחת קרניים ובדיקת נקודות חיתוך מיותרות.

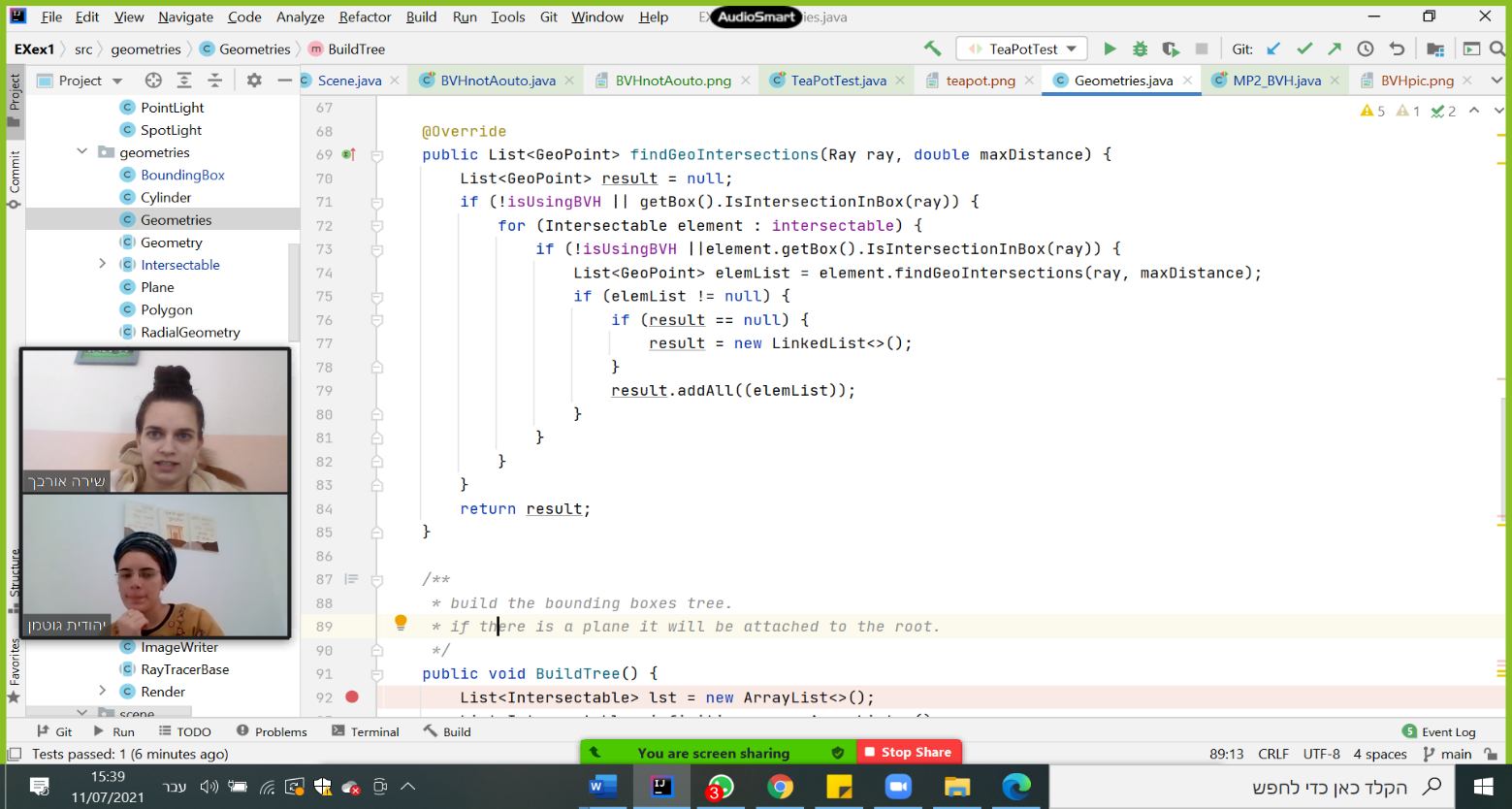


**BVH:**

הוספנו לפרוייקט מחלקת Bounding Box לייצוג הקופסאות העוטפות (מיקמנו אותה בתיקיית geometries בגלל משמעותה הפיזית ולא בגלל שלב השימוש בה). הקופסא מקבילה לצירים ומיוצגת ע"י ערכי הX,Y,Z המינימליים והמקסימליים שלה. בממשק Intersectable, שהפך למחלקה אבסטרקטית, נוסף שדה Box כך שלכל גוף תהיה קופסא עוטפת. הגדרת גבולות הקופסא תקרה ברגע שתתבצע קריאת get box (המוגדרת כברירת מחדל בממשק, ומזמנת באופן חד פעמי את פעולת set box של הgeometry הנוכחי). המחלקה מכילה גם פונקציה בשם IsIntersectionInBox המקבלת קרן ובודקת האם הקרן חותכת את הקופסא. לכתיבת פונק' זו נעזרנו במצגת הזו (קישור אליה מובא כהערה בכותרת הפונקציה):

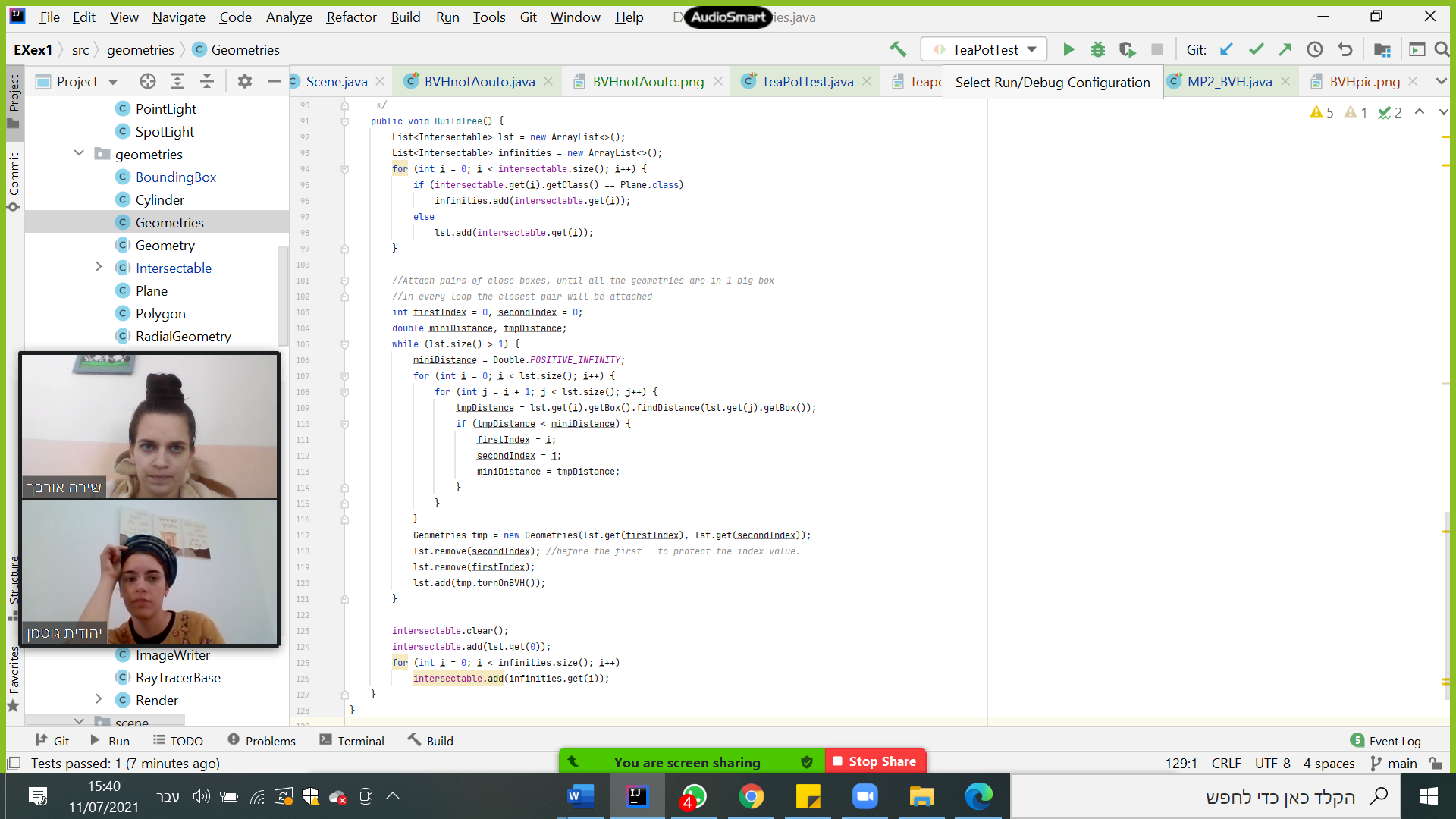


הקפדנו להשאיר את האפשרות להציג סצנות ללא השיפור (כלומר לא פגענו בתפקוד הקוד המקורי), כך שאפשר "להפעיל/לכבות" את האפקט הנ"ל בזמן יצירת סצנה. 'הדלקת' השיפור תיעשה ע"י שימוש בפונקציה turnOnUsingBVH על הסצנה הנשלחת לסורק הקרניים. הפונק' מחזירה את הסצנה (עם השיפור) כך שאפשר לזמן אותה תוך כדי השליחה לRay Tracer.

הדלקת השיפור מפעילה flag ברשימת הגופים של המחלקה(מסוג Geometries) ומזמנת את פונקציית BuildTree (במחלקת geometries) לבניית היררכיית הגופים. הפונקציה מאחדת כל פעם את שני הגופים הקרובים ביותר לגוף אחד, עד שהרשימה מכילה אובייקט אחד המכיל את כל הגופים של הסצנה (בתוך קופסא אחת גדולה). מציאת הגופים הקרובים מתבצעת ע"י הפונקציה findDistance במחלקה box המחזירה את המרחק בין מרכזי 2 קופסאות.

במצב הזה, כשנזמן את פונק' findGeoIntersections על רשימת הגופים, נצטרך להיכנס לעומק העץ על מנת להגיע לגופים עצמם. בכל פעם נתקדם לרמה הבאה בעץ רק אם הקרן הנתונה חותכת את הקופסא של הרמה הנוכחית, אם לא- נעצור ונחזיר null.

פונק' בניית העץ להיררכיית הגופים, במחלקת geometries:



המערכת מאפשרת למשתמש לבנות גם היררכיה ידנית לגופים. במצב כזה עליו לזמן את הפונק' turnOnManuallyBVH שמדליקה את השיפור ללא בניית העץ. בנינו סצנה עם עשרות גופים ויצרנו לה טסט עם בניית עץ ידנית ובניית עץ אוטומטית. ההרצה של הידני לקחה 1.259 שניות, והרצת האוטומטי לקחה 1.570 שניות.

**Multi-threading:**

הוספנו לפרויקט גם שיפור של תהליכונים. ניתן להפעיל את השיפור בזמן יצירת סצנה ע"י שימוש בפונקציה setMultithreading על הrender ושליחת מספר התהליכונים הרצויים.

השארנו את האפשרות להציג סצנות ללא השיפור(כלומר לא פגענו בתפקוד הקוד המקורי), כך שאפשר "להפעיל/לכבות" את התהליכונים.

בפונקציה renderImage שילבנו את הפונקציה שכתבנו לפני השיפור, עם הפונקציה שכתב פרופ' דן זילברשטיין לתהליכונים. בפונקציה זו נגדיר Pixelשיהיה רצף לינארי (בגודל כמות הפיקסלים שיש בתמונה) של הפיקסלים שהיו במטריצה עד עכשיו, בעזרת מחלקת פיקסל.

נייצר מערך של רפרנסים לתהליכונים, ועבור כל אחד נייצר תהליכון חדש, שיעבור על כל הפיקסלים וישלח אותו לפונקציה writePixel. לאחר מכן נפעיל את כל התהליכונים במקביל, ונעשה השהייה כדי שכולם ייגמרו לפני השלב הבא של writeImage .

בנוסף יש לנו בפונקציה זו גם הדפסה לקונסול של אחוזי ההתקדמות של התהליך, כך שהמשתמש יכול לראות בכל רגע נתון את התקדמות התהליך.

פונקציית renderImage המעודכנת:



הרצת טסט הקומקום (teaPotTest) שכתב דן זילברשטיין לקחה לנו:

לפני שיפורי הריצה(MP2): 4.29.302 דקות

עם הפעלת תהליכונים: 2.17.564 דקות

עם הפעלת :BVH 8.590 שניות

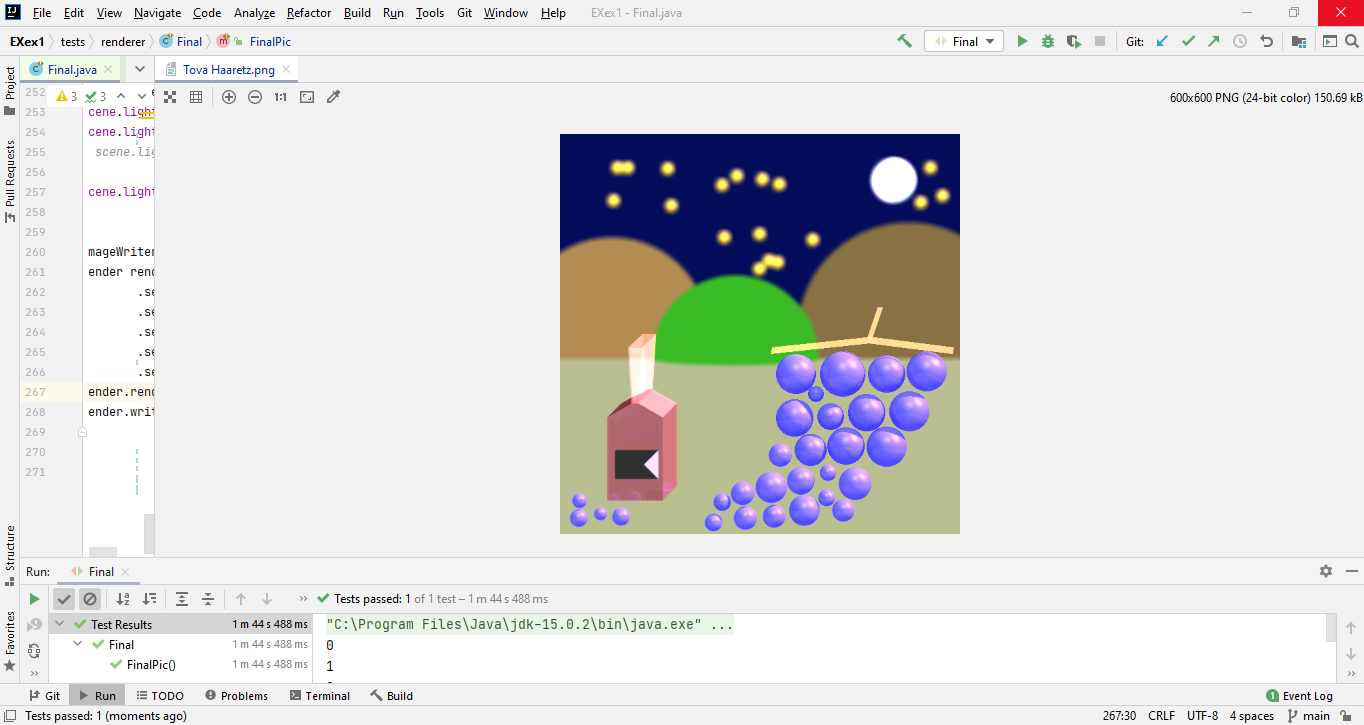
עם כל השיפורים: 6.335 שניות

הרצת טסט הסצנה שלנו (FinalPic) לקחה לנו:

בלי שיפורים: 27.08.963 דקות

עם תהליכונים: 15.11.589 דקות

עם שיפור BVH: 3.50.809 דקות

עם כל השיפורים: 1. 44.488 דקות

כלומר יחס זמן הריצה לפני שלב זה (mp2) השתפר כמעט פי 20.