Geometrical Image Processing

Computer Science Faculty

Technion

דו"ח פרויקט רנדור קרניים בשיתוף עם רפאל

- פרטי הסטודנטים:

  אור חי קרן 206018806.

  איתי לוי 205609175.

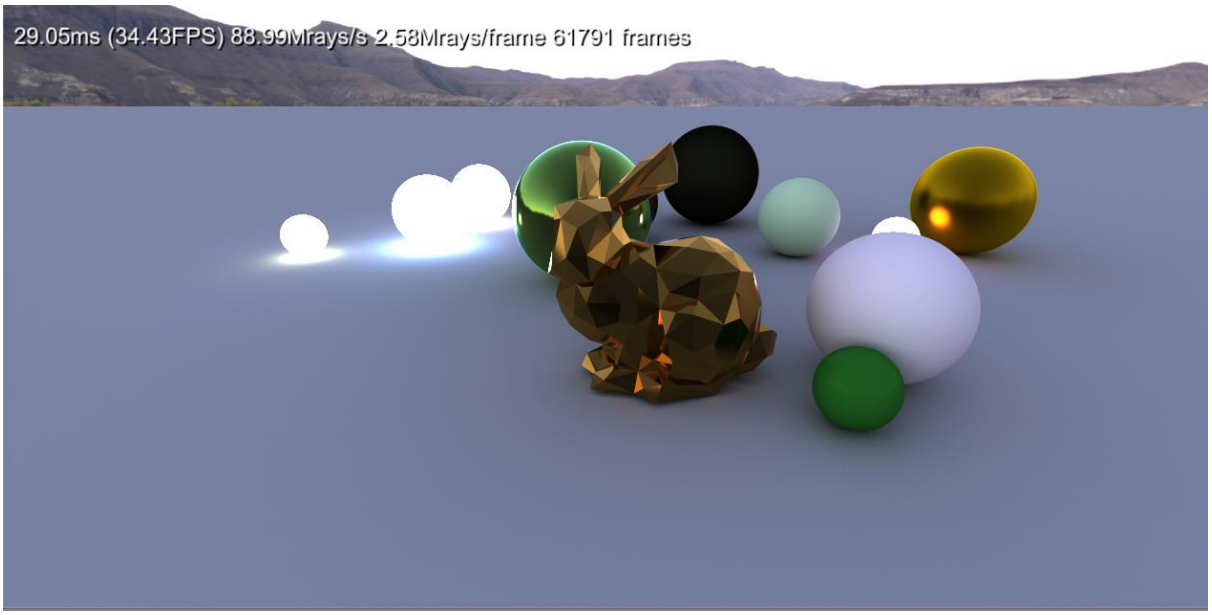
- שמות המנחים:
  טכניון – בועז שטרנפלד וירון חונן.
  רפאל – גיא קופלוביץ'.

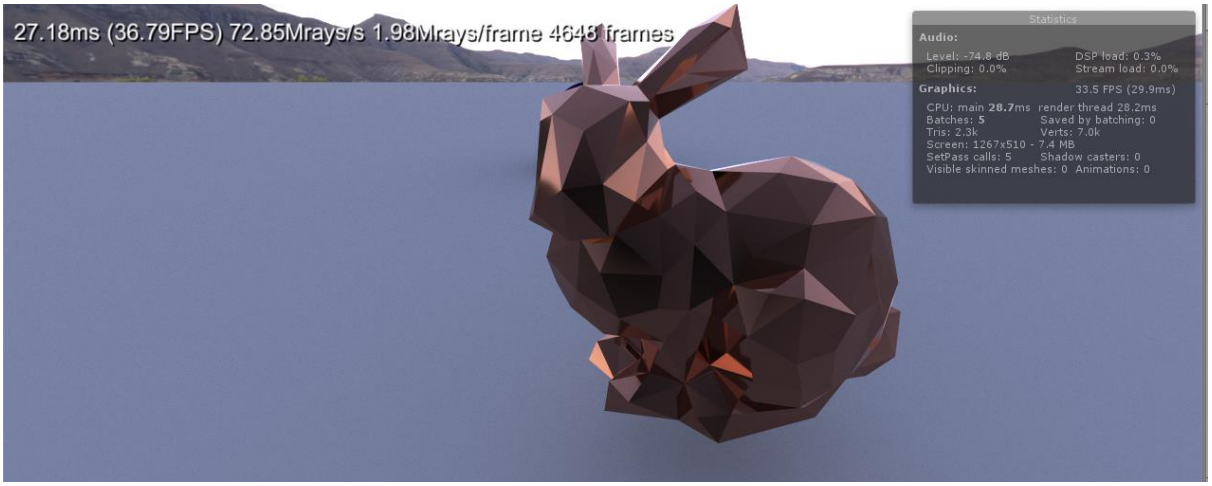
- תקופת הפרויקט: סמסטר חורף.
- תאריך: 2020.

- **רקע:** מטרת הפרויקט הייתה לבדוק האם יש אפשרות לממש השפעת גופים פולטי אנרגיה (כמו מנוע של רכב) על רנדור התמונה, ובפרט לבצע זאת באמצעות טכנולוגיית ray-tracing. חלקו הראשון של הפרויקט מטרתו לבדוק האם ניתן לממש מודל יחסית פשוט של תאורה (cook-torrance) בray-tracing, וחלקו השני של הפרויקט מטרתו לממש את מודל הגוף השחור כפי שתואר קודם.
רפאל מעוניינים בטכנולוגיית ray-tracing כי היא מאפשרת לנו למדל סצנות בצורה מדויקת.

- **התהליך:**
1. בתחילת הדרך, קראנו מאמרים ומסמכים שמסבירים על רנדרים, שיידרים, בפרט מודלים כמו cook-torrance וphong והבנו לעומק את המודלים הפיזיקליים הנדרשים ברמת התאוריה.
2. ניסינו לממש שיידר cook-torrance ע״י שפת השיידרים של blender, אך נתקלנו במספר בעיות. ראשית, שפת השיידרים של blender כמעט ולא מאפשרת לממש פונקציות פיזור חדשות, אלא רק להשתמש בקיימות ולבצע חישובים עליהן. בנוסף, גילינו שהטכנולוגיה שבה blender משתמש לray-tracing (cycles), משתמשת במעבד במקום בכרטיס המסך, דבר שיאט מאוד את החישובים כשנגיע לסצנות האמיתיות שרפאל מנסים למדל.
3. החלטנו לנסות לבדוק תוכנה אחרת, Unity. הצלחנו לממש בה שיידר cook-torrance, אך בדיעבד לאחר פגישה עם גיא מרפאל גילינו שהתוכנה משתמשת בטכנולוגיית rasterization, והתמיכה של התוכנה הבסיסית בray-tracing מוגבלת מאוד.
4. בעקבות השלב הקודם, החלטנו לחזור לבלנדר לנסות לממש פונקציית פיזור חדשה בתוך קוד המקור של בלנדר. אולם, לצערנו, קוד המקור של cycles מורכב מאוד וכמעט חסר תיעוד, וניסיוננו לפנות למפתחי המנוע היו חסרי הצלחה. לאחר שיחה עם בועז, המליץ לנו לעבור לאפיק אחר, והפנה אותנו לray-tracer בunity שמתכנת בשם David כתב.
5. לאחר הבנת קוד הרנדר של David, הצלחנו לממש את מודל cook-torrance ברנדר ולרנדר סצנה בסיסית שכוללת ספירות וארנב שמורכב ממשולשים.
6. לאחר ביסוס אפשרות יצירת מודלים בunity שמחושבים בray-tracing, התחלנו לחקור וללמוד על התיאוריה של גוף שחור (או גוף אפור). קראנו מאמרים שמתארים את אופן החישוב של פליטה של גוף שחור ואפור, והתייעצנו עם גיא מרפאל בנוגע לאופן החישוב שנרצה שיהיה במודל שלנו.
7. לאחר ביסוס הידע התיאורטי, פנינו למודל שלנו ומימשנו בתוכו אפשרות להוספת גופים אפורים פולטי אנרגיה. בנינו לדוגמא, סצינה פשוטה שמורכבת מרכב וגוף ריבועי שמדמה את המנוע החם של הרכב, וכך נוכל לראות את הפליטה של המנוע.
8. לסיום, הכנו קובץ דוקומנטציה שמסביר בדיוק כיצד להתקין את התוכנה עם השיידר שלנו, וכיצד להשתמש בשיידר ולהוסיף גופים חדשים לסצינה.

- **תוצאות ומסקנות:**
כיום, יש לנו מודל cook-torrance משולב עם גופים אפורים מוכן שמחושב בray-tracing על הgpu, עם מס׳ סצנות בסיסיות שמדמות את היכולות של השיידר.
כלומר, מסקנת הפרויקט היא שאכן ניתן לממש שיידר כזה בטכנולוגיית ray-tracing. נשים לב, שאם מוסיפים אובייקטים מורכבים לסצנה, זמן הריצה הוא ארוך בגלל החישובים הכבדים שמתבצעים בray-tracing, ולמרות זאת לדעתנו ניתן גם לבצע בו אופטימיזציות שימזערו את זמן הריצה של השיידר.





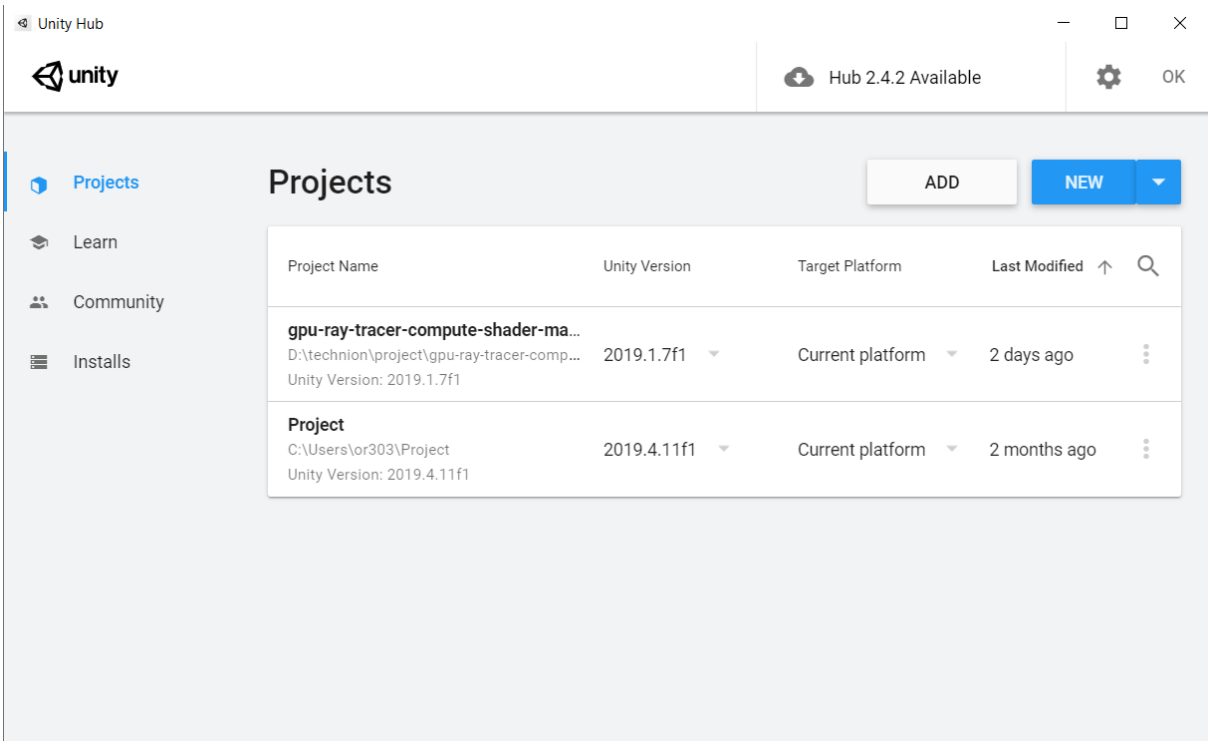
- **ביבליוגרפיה:**

1. המודל הפיסיקלי - https://google.github.io/filament/Filament.md.html.
2. מימוש המודל בטכניקת ray-tracing: http://www.codinglabs.net/article_physically_based_rendering_cook_torrance.aspx.
3. תיעוד ray-tracing של David בunity: https://unitylist.com/p/n7u/gpu-ray-tracer-compute-Shader.
4. תיעוד osl של בלנדר: https://github.com/imageworks/OpenShadingLanguage/blob/master/src/doc/osl-languagespec.pdf.

נספח א' – כיצד להפעיל ולהשתמש בshader:

1. ראשית יש להוריד את תוכנת unity בגרסת 2019.1.7 (לא כל כך קריטי הגרסה הראשונה שמורידים, ברגע שעושים import לפרויקט הוא מאפשר להוריד את הגרסה הרלוונטית).

2. לאחר הורדת הunity, יפתח המסך הבא:

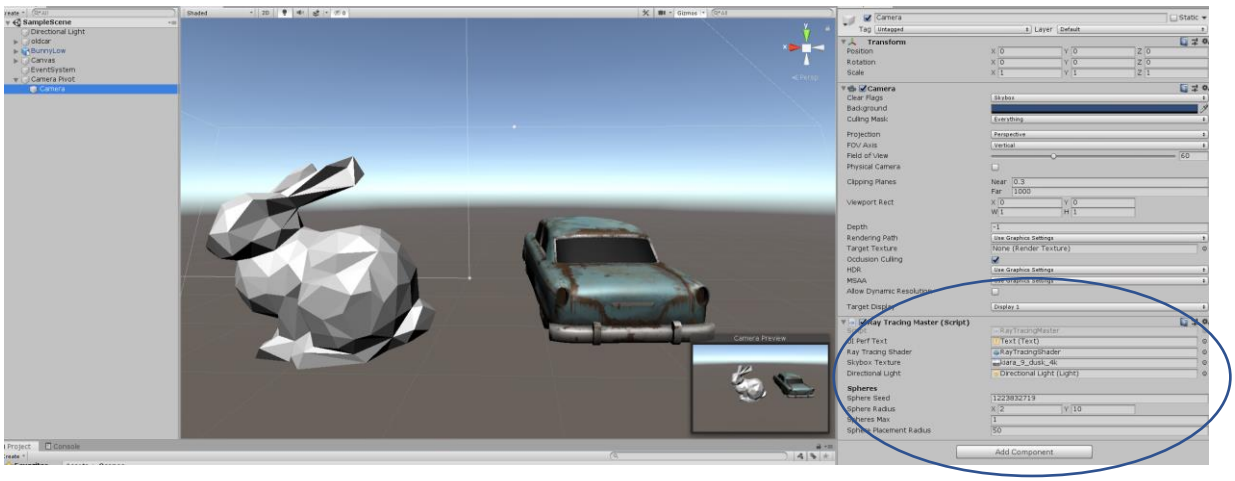


שבו צריך ללחוץ על add ולהכוון לתיקייה של הפרויקט שחילצתם מתוך הzip.

3. לחצו על הgpu-ray-tracer-compute-shader ואז יפתח לכם הפרויקט. שימו לב שאם לא מופיע לכם הארנב והמכונית, יש להיכנס לתיקיית הscenes וללחוץ פעמיים על הscene שנמצא שם.

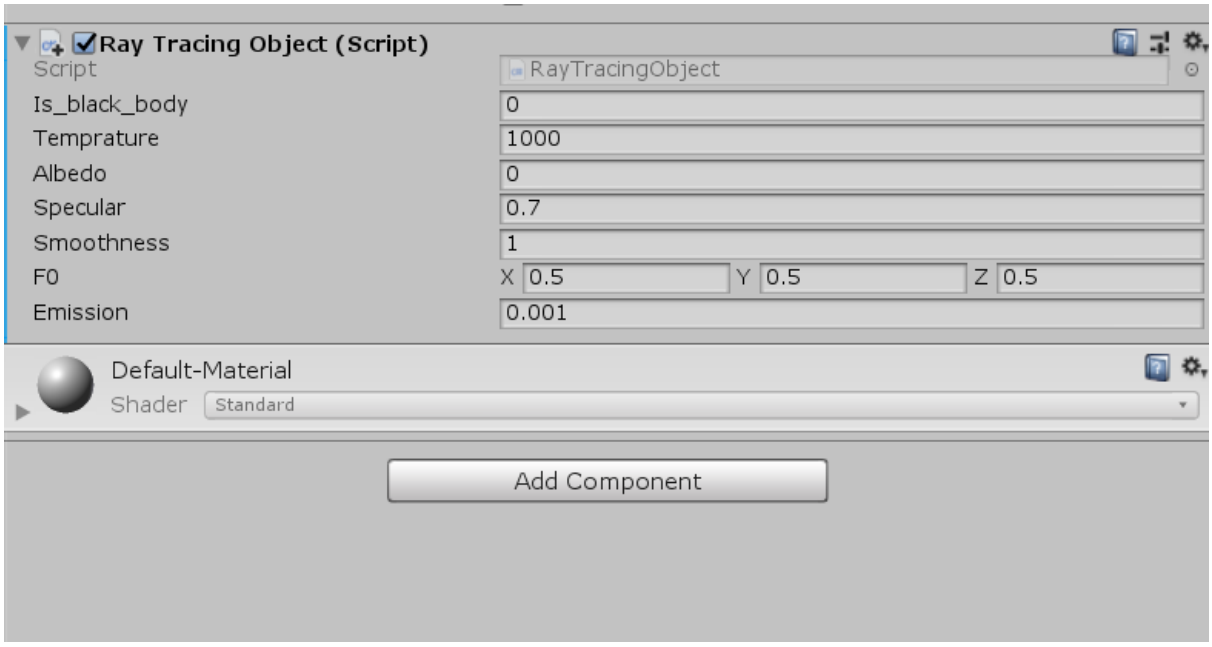
4. שימו לב, בכדי להריץ הרצה ראשונית יש ללחוץ על כפתור הplay, הסצנה שהכנו עבורכם היא יחסית כבדה, לכן כדאי במידה והיא כבדה מידי עבור הgpu, למחוק את המכונית.

שימו לב, שבאובייקט המצלמה יש פרמטרים שמאפשרים לכם לגנרט spheres אקראיים ברחבי הסצנה:



בנוסף, אם תרצו להוסיף אובייקטים נוספים לסצנה שירונדרו ויופיעו בעת הרנדור, יש לגרור עליהם את הקובץ RayTracingObject שנמצא בתיקייה הראשית.

לאחר שתגררו את הקובץ הזה על האובייקט, יופיעו לו הפרמטרים הבאים שישפיעו על רנדורו בסצנה:



הקוד:

הקוד המרכזי של הrenderer נמצא בקובץ RayTracingShader.compute שגם הוא נמצא בתיקייה הראשית, הפונקציה המרכזית שבה מבוצעים כל חישובי הcook-torrance והblack-body נקראת Shade(). הפונקציות בקובץ מתועדות ומובנות לקריאה. שימו לב שספציפית בתוך פונקציית הshade כל חלק וחלק מתועד, החלק האחרון מהווה את ערך החזרה של הפונקציה ואת חישוב הblack-body, ערך החזרה של הפונקציה מייצג את הפליטה של האובייקט שבו פגעה הקרן.

שימו לב שקיים גם קובץ הRayTracingMaster.cs שהוא גם רלוונטי לכל נושא הray tracing renderer ומטרתו להכין את כל הסצנה והפרמטרים לrenderer עצמו, גם במובן של ניהול הזיכרון.

הפרויקט מבוסס על ray tracing renderer קיים, כך שהתיעוד המעמיק לשאר הפונקציות נמצא בלינק הבא: https://unitylist.com/p/n7u/gpu-ray-tracer-compute-Shader.