

Recitations

- *Week 0: Home assignment 0 (not graded): Introduction to OOP with Java.*
- *Week 1: Home assignment 1: Implementation of primitives with operations, geometries*
- *Week 2: Home assignment 2: Implementation of geometries primitives' unit testing for primitives (JUnit). Implementation of normal calculation and their unit testing through geometries.*
- *Week 3: Home assignment 3: Implementation of ray-geometry intersections and their unit testing through geometries.*
- *Week 4: Home assignment 4: Implementation of camera class, rays through view plane construction and unit testing of camera*
- *Week 5: Home assignment 5: Implementation of ambient light, scene, renderer and image-writer classes with their appropriate test units*
- *Week 6-7: Home assignment 6: Adding material support. Implementation of Phong model, with light emission, directional, point and spot lighting, multiple light sources.*
- *Week 8: Home assignment 7: Implementation of shadow rays, reflection and refraction*
- **Week 9: Mini-project 1: Implementation of a picture improvement algorithm**
- *Week 10-11: : Mini-project 2: implementation of a performance improvement algorithm*
- *Week 12-13: Mini-projects presentation*

Mini-Project 1

Super-sampling based picture improvements

In this part of the mini-project each team
will be assigned one of four algorithms,
design it and implement it in their own
ray tracer

בשלב הזה של המיני-פרויקט, יבצע כל זוג אחד מארבעת שיפורי התמונה המתבססים על ריבוי דגימות (super-sampling), או בפשטות – ע"י אלומת קרניים במקום קרן בודדת. השלב הזה כבר לא ייקרא "שלב" או משהו דומה – אלא "מיני-פרויקט ראשון". המרצה יקבע לכל זוג איזה מהשיפורים הזוג יבצע.

Mini-Project 1 – for 1 week

Plan, design and implement algorithm

Create picture with **many** different objects demonstrate the feature

Bonus1: perfect and efficient result: up to 2pt

Bonus2: Each additional feature (from the four features of super-sampling) with correct architecture: up to 2pt (each one)

Bonus3: Jittered super-sampling pattern – 1pt

כמו שכתבנו בשקף הקודם – השלב הזה כבר לא ייקרא "שלב" אלא "מיני-פרויקט ראשון". וזאת מכיוון שלא תקבלו הנחיה מפורטת לביצוע כפי שהיה בשלבים הקודמים. עליכם יהיה לעשות עבודה עצמאית של חקר, חשיבה וקבלת החלטות על ארכיטקטורה, מודל היישום, וכו', ומימוש עצמאי לחלוטין. כמובן שבשלב הזה תידרשו להפגין שליטה מלאה בפתרונכם, כולל נימוקים להחלטות שקיבלתם שמבוססים על העקרונות ועל הגישות שלמדתם בקורס התיאורטי. בשלב הזה, אתם גם נדרשים לבנות תמונה מרובת גופים ומקורות תאורה שתוכל גם להמחיש את השיפור שביצעתם. השלב צריך להתבצע תוך שבוע אחד. מרצה של כל קבוצה יקבע לכל זוג בקבוצתו מה השיפור שהזוג עושה. הפתרון של כל זוג צריך לקחת בחשבון את **הנחת התשתית לביצוע כל ארבעת השיפורים האלה**. זאת אומרת – **הארכיטקטורה חייבת להיות כזו שתתאים לכל אחד מהשיפורים ללא חזרות מיותרות**, ומימוש הפונקציונליות הרלוונטיות בחבילות ובמחלקות מתאימות ע"פ הגישה של עיצוב מונחה אחראיות (RDD).

בשלב הזה יש 3 בונוסים:

1. ביצוע מושלם מבחינה הנדסית (ומכל יתר הבחינות) יכול לזכות את הזוג (בנוסף לניקוד המלא של השלב) בבונוס עד 2 נק' ע"פ שיקול דעת המרצה
2. כל שיפור נוסף מארבעת השיפורים (בנוסף לזה שהזוג קיבל מהמרצה) יכול לזכות את הזוג בבונוס עד 2 נק' לכל שיפור נוסף (1 או 2 נק' לפי איכות הביצוע)
3. במימוש ריבוי דגימות – שימוש בתבנית jittered (ראו בהמשך) תזכה את הזוג בבונוס נוסף של 1 נק'

Single ray → beam of rays

Multiple eye rays

- Jagged edges → **Anti-Aliasing(1)**
- Everything is in focus → **Depth of Field (2)**

Multiple shadow rays

- Hard shadows → **Soft Shadows (3)**

Multiple reflection rays

- Surfaces are perfectly shiny → **Glossy Surfaces (4)**

Multiple refraction rays

- Glass is perfectly clear → **Diffused Glass (4)**

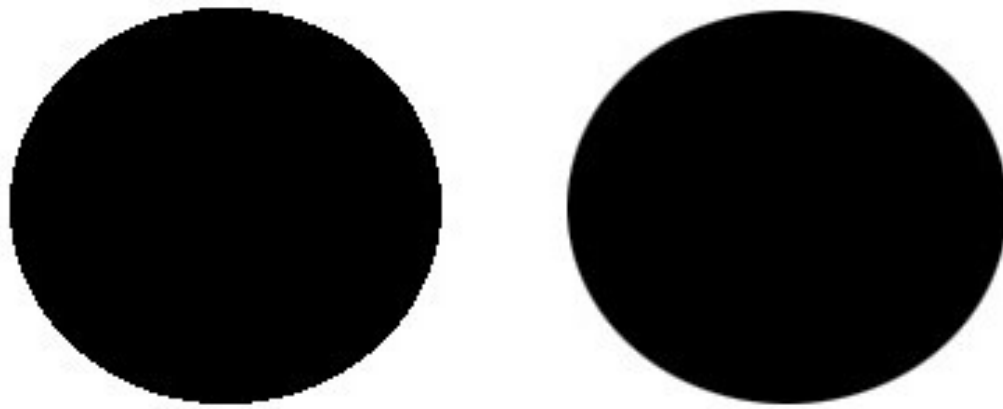
ארבעת השיפורים הם כדלקמן :

1. **החלקת עקומות** – החלקת את המסוריות של העקומות עקב פיקסליזציה (ע"י ריבוי דגימות של קרני מצלמה)
2. **עומק השדה** – יצירת תחושת עומק תמונה ע"י הורדת חדות של האובייקטים שקרובים מדי או רחוקים מדי (ע"י ריבוי דגימות של קרני מצלמה – בשיטה שונה לגמרי מהשיטה של השיפור הראשון)
3. **צללים רכים** – "מריחת" גבולות הצל החדים מדי עקב היות מקורות אור "אידיאליים" (בגודל נקודה) (ע"י מתן גודלוצורה למקורות האור וריבוי דגימות של קרני הצללה)
4. **זכוכית ומראה מאט** – "מרתח" האובייקטים שמאחורי זכוכית (אובייקט שקוף) או אובייקטים שמשתקפים ממראה (ע"י ריבוי דגימות של קרני שקיפות והשתקפות)

בכולם השיפור מתבצע ע"י מיצוע צבע (ברובם) או מקדם שקיפות (בצללים רכים) עבור הקרניים שבאלומה של ריבוי דגימות: נניח שעבור קרן מסוימת יצרנו אלומה של 1000 קרניים, לאחר חישוב הצבע (ברוב השיפורים) של כל קרן באלומה, נחשב את הצבע הממוצע של האלומה. הממוצע הזה יוחזר כצבע הקרן הבודדת המקורית שלפני השיפור. הפרטים על הגישה הננקטת בכל שיפור ושיפור נלמדו עד עומק מסוים בקורס התיאורטי. בשקפים הבאים נזכיר בקצה על כל שיפור ושיפור.

Improving Ray Tracing

Anti-Aliasing

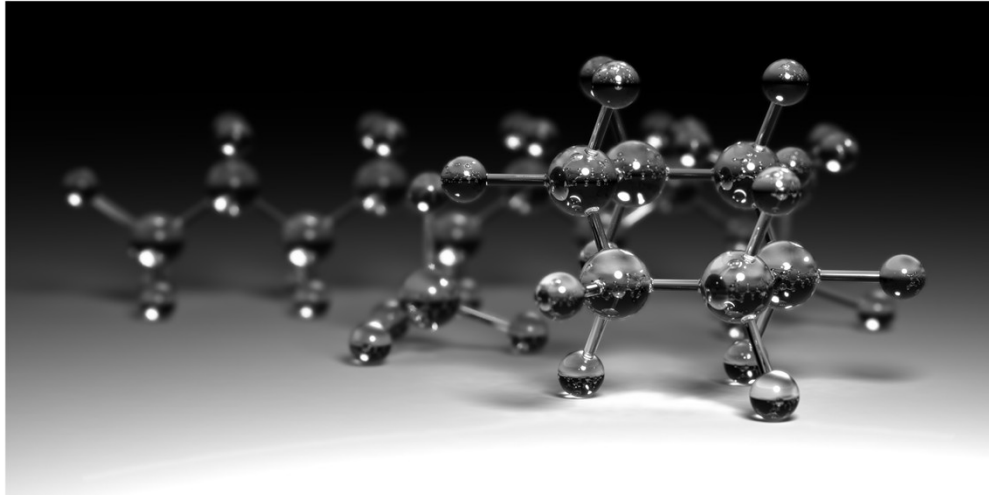


1. החלקים עקומות

משמאל – כדור בתמונה המפוקסלת לפני השיפור. מימין – התמונה המשופרת (עם אותה הכמות של פיקסלים), שבה חלק מהפיקסלים לאורך הגבול של תמונת הכדור צבועים בגוונים של ערבוב צבע הכדור וצבע של מה שמאחורי הכדור (למשל – צבע הרקע). בדוגמה הזו אלו שחור ולבן – ולכן הפיקסלים לאורך הגבול הם בגוונים של אפור. השיטה הזו מייצרת תחושת גבול חלק עבור העין האנושית. מכיוון שמדובר בשיפור בקרניים של מצלמה – ההשפעה של השיפור היא על כל התמונה.

Improving Ray Tracing

Depth of field

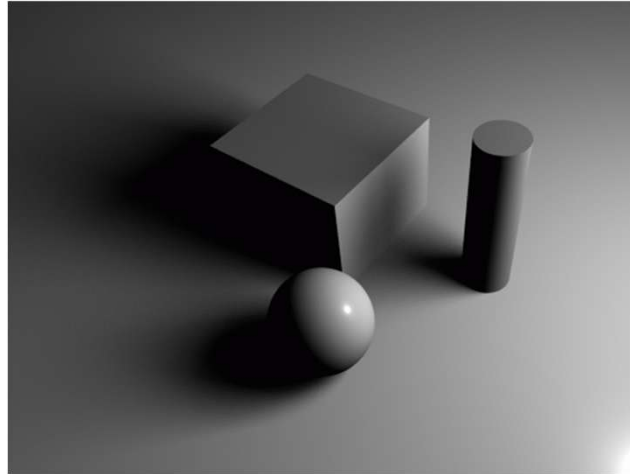


2. עומק השדה

כפי שניתן לראות בתמונה – הכדורים הקרובים למצלמה והכדורים המרוחקים מהמצלמה קצת מטושטשים, והמרוחקים יותר – מטושטשים יותר. קיים איזשהו מרחק שבו הגופים נראים בחדות. מכיוון שמדובר בשיפור בקרניים של מצלמה - ההשפעה של השיפור היא על כל התמונה.

Improving Ray Tracing

Soft shadows

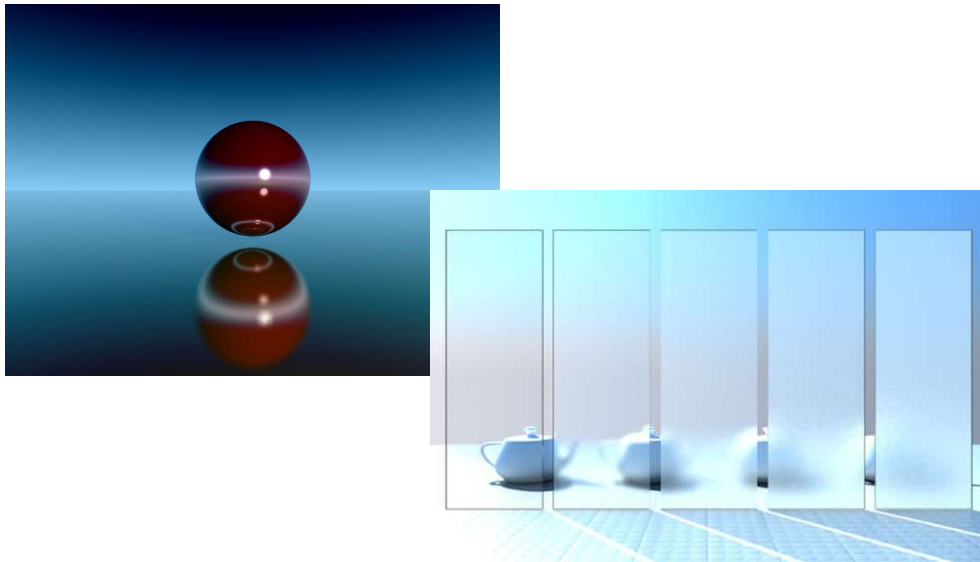


3. צללים רכים

כפי שאפשר לראות בתמונה – הצללים (הצבע הכהה על פני המשטח מאחורי הגופים) מתבהר בהדרגה וללא גבול חד בין האזור המואר לאזור המוצל. התוצאה מושגת על ידי יצירת אלומת קרניים לכיוון גוף תאורה שיש לו גודל. זאת אומרת – האלומה של קרני הצללה לאותו גוף התאורה נוצרים לכיוון אזור המטרה שמסביב למיקום מקור האור. רק מקורות האור נקודתי וספוט (Point/SpotLight) שמושפעים מהשיפור הזה. הצל מאור כיווני (Directional Light) יישאר תמיד חד (לא ניתן לתת גודל למרוא אור שאין לו מיקום). כל אחד ממקורות האור בתמונה יכול להיות בגודל שונה (גודל אפס [0] – מקור אור "אידיאלי" כמו קודם, ללא צללים רכים). זאת אומרת – ההשפעה של צללים רכים תהיה שונה בין מקור אור אחד למשנהו, והיא תהיה תלויה במקור האור הספציפי (לא סוג מקור האור ספציפי אלא מקור האור ספציפי – אפילו יש כמה מקורות אור מאותו סוג – צללים רכים עשויים להיות שונים ביניהם). כך גם במציאות – רכות הצל תלויה בגודל של המנורה ובמרחק מהמנורה.

Improving Ray Tracing

Glossy surfaces and Diffused (Blurry) Glass



4. זכוכית ומראה מאט

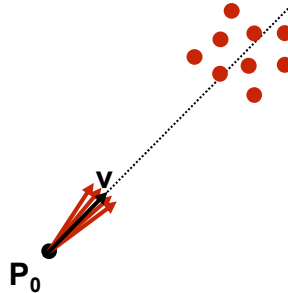
כמו במשטח מתכתי "מבריק" או השתקפות מפני המים, וגם כמו בזכוכית ששמים בדלת למקלחת או בחלון בין משרד מנהל לבין פרוזדור וכדומה. אנחנו רואים השתקפות הכדור מפני המים שהיא השתקפות מעורפלת טיפה, או הקומקום מאחורי הזכוכיות. הזכוכיות מסודרות משמאל לימין – השמאלית שקיפות רגילה, כמה שיותר ימנה – הזכוכית יותר מאט, עד כדי כך שבזכוכית הימנית רואים שיש משהו מאחוריה – אבל אפילו בלתי אפשרי לזהות מה יש שם.

Improving Ray Tracing

How to generate a beam of rays?

There is always a start point and central ray's direction

Need to generate a target point for creating each ray



NB: For depth of field there is a target point and start points need to be generated

נרענן בקצרה את הדיון שהיה בהרצאה לגבי יצירת אלומות הקרניים. הדיון הזה הוא כללי – לכל השיפורים. הוא בעצם מתייחס למימוש התשתית שלא קשורה לשיפור מסוים (מארבעת השיפורים) וממומש בנפרד – עליכם לחשוב איפה ואיך. יחד עם זאת, כל אחד מהשיפורים הספציפיים שיתבצע על ידי כל זוג – ישתמש במימוש הזה.

ברוב השיפורים יש לנו קרן מקורית (שבמקומה אנחנו רוצים ליצור אלומה). הקרן הזו מתחילה בנקודה מסוימת ומכוונת לכיוון מסוים (בדרך כלל לכיוון נקודה נתונה – כמו מרכז פיקסל, או מיקום מקור אור). ביצירת אלומה – במקום נקודת יעד אחת חישוב כיוון הקרן (עבור עומק השדה – נקודת מקור אחת) – נעבוד עם "אזור מטרה" (TargetArea). ניצור את אזור המטרה "מסביב" לנקודת היעד המקורית. באזור המטרה הזה, ניצור הרבה נקודות. אלו נקודות היעד לכל אחד מהקרניים שייכללו באלומה.

בעומק השדה הטכניקה תהיה הפוכה – ניצור אזור "מקור" – סביב מרכז "חלון הצמצם" (מיקום המצלמה), וניצור קרניים מריבוי נקודות בחלון הצמצם דרך אותה נקודת היעד (נקודת פוקוס – נקודת מיקוד).

בסופו של דבר הטכניקה אחידה – יצירת אזור סביב נקודה נתונה, יצירת ריבוי נקודות באזור הזה, ולאחר מכן יצירת קרניים בעזרת הנקודות האלה. בסופו של דבר יחושב צבע (או מקדם שקיפות בצללים רכים) עבור כל אחד מהקרניים – ואז יחושב הצבע (או מקדם השקיפות) הממוצע של כל הקרניים באלומה.


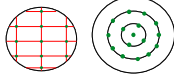
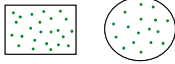
לאחר שאנחנו מבינים מה עלינו לעשות, עולות שלוש שאלות:

1. מה תהיה הצורה של "אזור המטרה"?
2. איך למקם את הנקודות המרובות ב"אזור המטרה"?
3. כמה קרניים ליצור באלומה (או במילים אחרות - כמה נקודות ליצור ב"אזור המטרה")?

Improving Ray Tracing

How to generate a beam of rays (super-sampling)?

Choosing Approach

- Define target area (rectangle? circle? other?) 
- Generate points by ...
- ... a grid? ... a spiral? circles? 
- ... randomization? 
- In any case: it is important to cover the **entire** target area **evenly**

1. מה תהיה הצורה של "אזור המטרה"?

האם שטוח או "נפוח" [התשובה – **שטוח**]? ריבוע? מלבן? עיגול? אליפסה? אזור מטרה מרובע או מלבני מייצר חוסר אחידות בהשפעת השיפור (מריחה יותר גדולה לכיוונים של הפינות מאשר לכיוונים של אמצעי הצלעות). יחד עם זאת, עיגול מייצר קושי מסוים ביצירה אחידה של נקודות (שניתן להתגבר עליו די בקלות ע"י קבלת החלטות אלגוריתמיות נכונות), וגם יוצר בעיה יותר קשה בשיפור הביצועים באמצעות ריבוי דגימות מסתגל (adaptive super-sampling) (בשלב האחרון – מיני-פרויקט שני). המלצתנו: להשתמש בריבוע (יותר קל אך מייצר עיוות אלכסוני) או בעיגול (איכות התוצאה יותר טובה אך קצת יותר מורכב). הזוגות שיקבלו בשלב האחרון לעשות adaptive super-sampling – יצטרכו אז לשנות לריבוע. אם הזוג יאפשר בחירה גמישה של הצורה – תבואנה עליו שתי ברכות – תוצאות יותר יפות בשלב הזה וגמישות לקראת השלב האחרון. יחד עם זאת מי שיעשה עיגול – מסתכן בכך שפיזור הנקודות שלו בעיגול לא יהיה אחיד – מה שעלול ליצור בעיות מסוימות באיכות התמונה. תקדישו מחשב בקבלת ההחלטה שלכם עניין הזה.

2. איך למקם את הנקודות המרובות ב"אזור המטרה"?

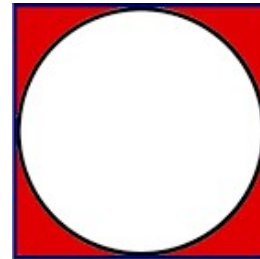
יש שיטות רבות. ניכנס יותר לעומק בשקפים הבאים. אך בכל שיטה – יש לדאוג ל**כיסוי אחיד ומלא** של אזור המטרה ע"י הנקודות. בכל שיטה בעצם נתבסס על שני וקטורים מאונכים בתוך אזור המטרה (בדומה לווקטורים של מצלמה "למעלה" ו"ימינה" שהשתמשו בהם לחישוב מרכזי הפיקסלים במשטח צפיה – View Plane). הווקטורים האלה הם מעין צירים X ו-Y במערכת קואורדינטות דו-ממדית של אזור המטרה.

Improving Ray Tracing

How to generate a beam of rays?

NB: If circle approach is chosen

- Generate points according to the square in which the circle is inscribed
- "Filter out" the points not in the circle (how to do it efficiently?)
- Choose number of points taking in account that certain percent of points will be filtered out – thus the final amount of points will be close to the planned one



$$\frac{S_s}{S_c} = \frac{4}{\pi} \approx 1.27324$$

בהקשר להערה לגבי אזור מטרה עגול בשקף הקודם :

אל תנסו להמציא שיטה מתוחכמת מתמטית. עד כה רוב מה שהציעו סטודנטים – היה לא כל כך טוב, כי פיזור הנקודות לא היה אחיד בתוך העיגול. השיטה הפשוטה היא ליצור ריבוע שחוסם את עיגול (העיגול בתוך הריבוע), וליצור נקודות בריבוע בכמות נקודות שהיא קצת יותר ממה שצריך (פי 1.27324 בערך – כיחס שטח הריבוע לשטח העיגול). הדיון לגבי כמויות הקרניים באלומה יהיה בהמשך. לאחר מכן, "נסנן" (לא לנכלול) את הנקודות שמחוץ לעיגול.

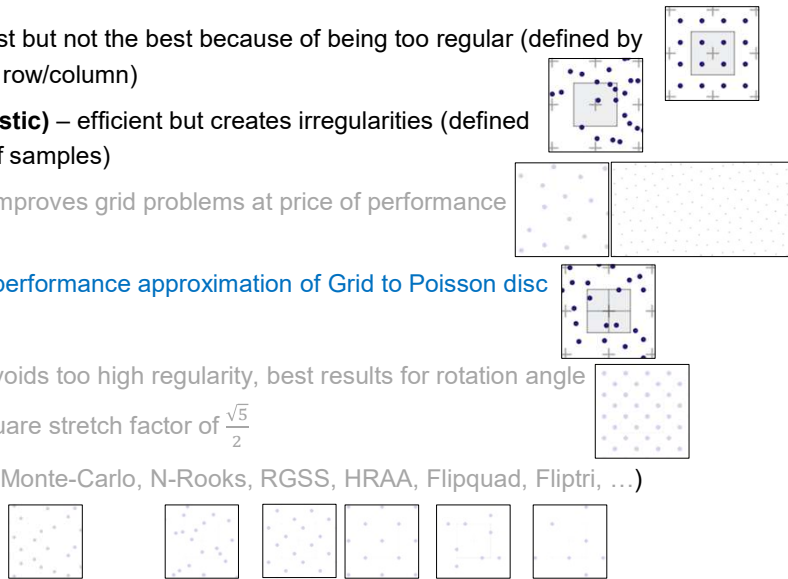
אם משתמשים בשיטה של רשת (גריד) – מייצרים רשת גדול יותר (בערך פי 1.27324) ממה שהתבקשנו, ומסננים את הנקודות שמחוץ לעיגול.

אם משתמשים בשיטה רנדומלית – מייצרים נקודות בלולאה תוך כדי סימון תוצאות מחוץ לעיגול, עד שנקבל את הכמות הנדרשת של הקרניים.

Improving Ray Tracing

Different super-sampling patterns

- **Grid** – the simplest but not the best because of being too regular (defined by step or amount in row/column)
- **Random (stochastic)** – efficient but creates irregularities (defined by total number of samples)
- **Poisson Disc** – improves grid problems at price of performance
- **Jittered** – better performance approximation of Grid to Poisson disc
- **Rotated grid** – avoids too high regularity, best results for rotation angle $\arctan\left(\frac{1}{2}\right)$ and square stretch factor of $\frac{\sqrt{5}}{2}$
- And more (Quasi-Monte-Carlo, N-Rooks, RGSS, HRAA, Flipquad, Fliptri, ...)



עכשיו נדון בתבניות אפשריות של פיזור הנקודות באזור מטרה. יש שיטות רבות כאלה שנידונו בקורס התיאורטי. לצורך הפרויקט, נתמקד בשלושה מהם:

1. רשת (גריד)

דומה מאוד למה שעשינו עם מרכזי הפיקסלים ביצירת קרניים של מצלמה. אזור המטרה שם היה משטח הצפייה – View Plane. שיטה קלה מאוד למימוש ומהירה מבחינת זמן ריצה. יש רק ביעה אחת – היא מייצרת הפרעות בתמונה – מעין גבולות "מפוספסים" בכיוונים מסוימים בגלל שהתבנית "מסודרת יותר מדי". לכן בדרך כלל לא משתמשים בשיטה הזו אלא אם כן יש אילוצים.

2. רנדומלי (סטוכסטי)

השיטה קלה למימוש ומהירה מבחינת זמן ריצה. יחד עם זאת, היא מייצרת הפרעה "מנומרת" (מריחה "מגורענת" – כמו בתמונות ישנות, לפעמים). ונדרש ליצור הרבה מאוד קרניים באלומה על מנת לצמצם את ההפרעה הזו.

3. "רוקד"/"עצבני" (jittered) (בונוס)

זהו שילוב של שיטת הרשת והשיטה הרנדומליות – במקום נקודה מדויקת בגריד מוגרלת הזזה קטנה (תצטרכו להקדיש מחשבה איך לעשות את זה נכון וטוב?)

השיטה קלה יחסית למימוש, והביצועים שלה יחסית טובים מאוד. יחד עם זאת, שימו לב שבביצועים שלה היא תעשה גם גריד וגם רנדומליזציה. כך שזה יהיה קצת יותר איטי משתי השיטות הראשונות לחוד. אבל מבחינת איכות התוצאה – היא הרבה יותר טובה (אם השיטה נעשתה נכון). מימוש השיטה הזו לא מורכב מדי ומאפשר לקבל 1 נק' בונוס.

Improving Ray Tracing

How many rays (quality vs performance)?

- 4 – too few (2x2 for grid) – start from it in the beginning and for debugging
- 10 – may be? Still too few (3x3 for grid)
- 25 – may be? Still quite a few (5x5 for grid)
- **80+ recommended for demo (9x9 for grid) – for test pictures**
- ~300 to ~1000 in final pictures (17x17 or 33x33 for grid) – for the bonus picture

והשאלה האחרונה – היא לגבי כמות הקרניים באלומה. ולצורך פשוטות נדבר על אלומה שנוצרת בעזרת אזור מטרה מרובע בשיטת רשת (גריד).

4 קרניים (רשת 2x2) תיצור תוצאה לא כל כך איכותית. אבל הכמות הזו מתאימה בהתחלה, כשצריך לדבג.

9 קרניים (3x3) – מתחיל לתת תוצאה סבירה בהחלקת העקומות, אך עדיין רחוק מלהיות מספיק על מנת לקבל איכות סבירה בשיפורים האחרים. עדיין אפשר להשתמש בכמות הזו עם התקדמות הפיתוח להרצות ניסוי.

לאחר שהכול עובד - בשביל תמונות פשוטות להדגמה (ושלא ייקח יותר מדי זמן ליצור אותן) מתאים לעשות 81 קרניים (9x9) [בהחלקת העקומות אפשר להסתפק ב-25 קרניים (5x5).

אך בתמונה הגדולה והמרשימה שנדרש לעשות לבסוף, על מנת לקבל תוצאה באיכות גבוהה – יש לבחור בכמות בין 300 ~ (17x17) עד 1000 ~ (33x33) קרניים.

שימו לב: המספרים שאתם רואים לעיל מבוססים כולם על מספרים שהם 2^k+1 (כאשר k חל מ-0 והלאה: 2, 3, 5, 9, 17, 33, ...). מי שיעשה ריבוי דגימות מסתגל בשלב האחרון – יבין למה.

Improving Ray Tracing Design

- Create blackboard entity for target area (to generate list of 2D points)
 - Can be used in all the features of MP1
 - In which package? Which data, functions?
 - 2D value entity? How to hold list of 2D points (which collection)?
 - Generate list of point once (on configuration change) or every time (grid vs. stochastic)?
- Generate or choose two orthogonal normalized vectors to provide blackboard X/Y axes
- After having list of 2D points, blackboard location and X/Y vectors – generate list of target 3D points
- Then generate vectors => rays of the beam
- Don't forget to decided whether to include the original (central) ray in the beam (if needed)

ההמלצות הכלליות לשלב זה הם :

1. **כדאי ליצור ישות [מחלקה] (למשל בשם Blackboard) עבור הפונקציונליות של אזור מטרה (שבתפקידו יהיה ליצור אזור מטרה ואוסף נקודות שמפוזרות בו)**
 - הישות הזאת תוכל לשמש את כל השיפורים
 - אובייקט של הישות הזאת תייצג אזור מטרה מסוים (עבור קרניים מסוימות) – תחשבו מה הכוונה ואיך לעשות את זה.
 - RDD: תחשבו באיזו חבילה לשים את הישות הזאת? איזה נתונים היא תחזיק? איזה מתודות יהיו לה? איזה נתונים המתודות האלו תקבלנה? וכו'
 - תחשבו איך ליעל את עבודת האובייקטים, למשל אם אתם מייצרים גריד בגודל מסוים – למה ליצור אותו כל פעם מחדש? כמובן זה תלוי בשיפור הספציפי, לא בכולם אפשר להשתמש באותה תבנית כל פעם.
 - האם ליצור אוסף זוגות מספרים עבור קואורדינטות פנימיות (x,y) של הנקודות באזור המטרה? ואז מהן ליצור את הנקודות במרחב תלת הממד של הסכינה? תחשבו האם זה מתאים למודל שבחרתם.
2. **עבור אזור מטרה יש ליצור זוג ווקטורים מאונכים (מעין צירים X ו-Y במערכת קואורדינטות דו-ממדית של אזור המטרה). בדרך כלל (ברוב השיפורים) אזור המטרה יהיה מאונך לקרן המקורית (שבמקומה מייצרים את האלומה). למעט צל רך ממקור אור מסוג ספוט, ששם הם יהיו מאונכים לכיוון הספוט.**
 - הווקטור הראשון (וקטור כלשהו שמאונך לווקטור הנתון – כיוון הקרן המקורית או כיוון הספוט) נוצר כפי שלמדתם באלגברה לינארית – זה "תרגיל" פשוט מאד (RDD – איפה לשים את הפונקציונליות הזו)
 - הווקטור השני – הוא נוצר כמובן ע"י מכפלה וקטורית בין הווקטור הנתון לבין הווקטור הראשון
3. **RDD: אם במודל שלכם יש צורך בכך, תחשבו איזו מחלקה ואיזו מתודה תהיה אחראית על יצירת אוסף נקודות תלת הממד (מתוך נקודות דו ממד) באזור המטרה ואיך, כמו כן – מי יהיה אחראי על יצירת הקרניים לאלומה – ואיך**
4. **תחשבו האם אתם רוצים לכלול את הקרן המקורית באלומה או לאו (לא ניתן לכם תשובה עכשיו), תחשבו על יתרונות ועל מגרעות, תדעו לנמק...**

Improving Ray Tracing Design

- Who is responsible for ray beam generation? New entity (if yes – where?)? Ray class?
- How and where to use the ray beam?

When doing ray tracing:

- Don't filter out calculation because of the central ray - do it for every ray in the beam, for example following checks:
$$n \cdot v = 0, \quad n \cdot l = 0, \quad \text{sign}(n \cdot v) \neq \text{sign}(n \cdot l), \quad \text{etc.}$$
- For averaging the color, do we divide by real number of rays in the beam (it may vary from time to time) or by the original number of rays?

נחזור לשאלה מהשקף הקודם – מי אחראי ליצירת אלומה עבור שיפור כזה או אחר? במחלקה חדשה של אלומה? במחלקת Ray? איזו מחלקה ואיזו מתודה יבקשו ליצור אלומה וישתמשו בה על מנת ליצור את השיפור הדרוש? כל אלו הן השאלות שעליכן לענות לעצמכם ולדעת לנמק למרצה שלכם.

הערה נוספת שקשורה לשיפורים של צללים רכים, שקיפות או השתקפות. בסריקת קרניים – באפקטים לוקליים (מקומיים) ואפקטים גלובליים – אנחנו בודקים כל מיני תנאים שעל פי התנאים האלה אנחנו מחליטים האם תהיה השפעה של מקור אור או של שקיפות או של השתקפות או של צל, על המשך החישוב או לאו. למשל בדיקות המבוססות על התנאים שמופיעים בשקף הזה:

$$n \cdot v = 0, \quad n \cdot l = 0, \quad \text{sign}(n \cdot v) \neq \text{sign}(n \cdot l), \quad \text{etc.}$$

אם תעשו את הבדיקות האלה על הקרן המקורית (שבמקומה רוצים ליצור אלומה) – עלול להיווצר מצב שיהיה אפקט חלקי בלבד. ברוב המקרים עלינו לבדוק את התנאי המתאים על כל קרן וקרן באלומה. כאשר אנחנו באים לסכם ולמצע את התוצאה – האם אנחנו לוקחים בחשבון את כל כמות הקרניים המקורית – או רק את הקרניים שלקחנו בחשבון אם התנאים האלה? כמוכן שהתשובה יכולה להיות תלויה בשיפור הספציפי שאתם עושים. עליכם להקדיש מחשבה ולקבל החלטות. כמוכן לאחר שלא שכחתם להעביר את בדיקת התנאים כפי שהסברנו לעיל.

שימו לב שהדיון הזה הוא בעצם תשתית לקבלת תאורה חלקית וצל חלקי במצב שהוצג בהרצאה כדוגמה של "שמש שוקעת" (רואים רק את החלק העליון של השמש – מרכז הכדור של השמש "מתחת לאופק"). לכן הדיון הזה קשור לצללים רכים, זכויות ומראה מטושטש.

Improving Ray Tracing Design

- Feature can be enabled/disabled from the test (mandatory requirement)
 - Modularity
 - Interface or Abstract class + setter choosing appropriate implementation
 - Configuration parameter + setter
- Hardcoded parameters are **prohibited**
 - Use configuration parameters + setters
 - Use default initialization values wherever possible
- RDD: data, actions, "communication" – where? how?
- Avoid code smells, refactor where needed
- Architecture, correct design and implementation – all affect the grading

יש לאפשר בניית תמונה עם/בלי שיפור בלי לגעת בקוד הגרפי (אלא רק ע"י שינוי בקוד של הטסט שבונה את התמונה)

- בין היתר, זה נדרש על מנת ליישם את הדרישה למודולריות
- עושים זאת ע"י פרמטר קונפיגורציה (פרטי כמובן) עם סטר (עדיף שיאפשר שרשרת אתחול של הישות הכוללת אותו)
- איפה יהיו הפרמטרים המתאימים? באיזה ממשק או באיזו מחלקה (אבסטרקטית או לא אבסטרקטית)? יתכן שתמצאו לנכון לשים את הסטר לא רק ביחד עם השדה אלא גם במחלקה שניתן להשתמש בה בטסט, ומשם הסטר יאציל את תפקידו למחלקה המתאימה.

יש למנוע אנטי-תבנית של hard code !!!

- כל הפרמטרים של השיפור חייבים להיות נתונים לשינוי מתוך הטסט שבונה את התמונה – כמות קרניים באלומה, מרחק פוקלי, קוטר של מקור תאורה, וכו'
- על מנת לאפשר בניית כל התמונות ללא שיפורים – אי הפעלה של סטר אמורה ליצור מצב של אי הפעלה של השיפור – זאת אומרת לכל הפרמטרים צריכים להיות ערכי אתחול מתאימים
- הנתונים החשובים שיש לאפשר את הקינפוג (ההתאמה) שלהם מהטסט:
 - כמות קרניים באלומה (ערך 0 או 1 אומר שאין ריבוי דגימות) – לכל אחד מהשיפורים בנפרד
 - מי שמאפשר גמישות – בחירת הצורה של אזור מטרה – בנפרד לכל שיפור
 - בצללים רכים, רדיוס מקור האור (לכל אחד רדיוס אחר, אם רדיוס=0 זה אומר שאין צל רך למקור האור הזה)
 - בזכוכית ומראה מטושטשת, זווית פתיחת החרוט של אלומת הקרניים (או יותר מדויק – מחצית גודל אזור המטרה, כאשר המרחק מראש הקרן עד אזור המטרה נקבע מראש):
- * מוגדר בנפרד לכל חומר של גוף שיש לו שקיפות ו-לאו השתקפות (כדאי שיהיה נפרד בין שקיפות להשתקפות באותו הגוף)
- * אם מחצית גודל אזור המטרה שווה ל-0 – לחומר הזה יש שקיפות ו-לאו השתקפות חדה (לא מטושטשת)
- * המרחק מראש הקרן עד אזור המטרה יינתן בקבוע ברמת האפליקציה
- 0 אם המרחק הזה יהיה 1 – מחצית גודל אזור המטרה הנו טנגנס של זווית הפתיחה של האלומה
- 0 אם המרחק הזה יהיה 100 – מחצית גודל אזור המטרה הנו השיפוע כמו בכבישים - באחוזים

כל ההחלטות של מיקום הנתונים, הפעולות וצורת התקשורת בין האובייקטים חייבים להיות על פי הגישה של עיצוב מונחה

אחריות – RDD

לפני הגשת השלב, חובה לעשות ריפקטורינג – לסלק את סירחונות הקוד, לייעל את הקוד מבחינת הביצועים
ארכיטקטורה, עיצוב הקוד, איכות ויעילות המימוש – הכול משפיע על הציון של השלב