דו"ח שלב א' במיני פרוייקט

מגישים: עמנואל קואנקה

מבוא

מטרה-מטרת המיני פרוייקט בשלב א' היא לבנות את הבסיס לבניית מיצג גרפי תלת-ממדי, נעשה זאת ע"י החבילה primitives הכוללת מחלקות בסיסיות באלגברה ובגיאומטריה.

Primitives

חבילה שהיא מכילה פונקציות שעושות פעולות בסיסיות בקואורדינאטות ובוקטורים במישור ובמרחב.

-החבילה כוללת את המחלקות הבאות:

1. Coordinate- יחידה על ציר המספרים. המחלקה מכילה:

א. משתנה coordinate מסוג double.

ב. בנאים ופונקציות טריוויאליות.

ג.פונקציית Boolean equals המשווה בין וקטורים ובעת הצורך משתמשת בפונ' ubstract שמצויה במחלקת utils לזיהוי ערך שקרוב לאפס כאפס.

ד.פונ' toString שמדפיס את הערך המוחזר בעשרוני.

ה. פונ' compareTo-שמחזירה 0 אם שווים 1 אם גדול ו1- אם קטן ממנו.

ו.פונ' substract-חיסור קואורדינטה מהנוכחית.

ז.פונ' add- מקבלת קואורדינטה ומשנה את הנוכחית ע"י תוצאת החיבור ביניהם.

ח.פונ' scale- מקבלת מספר ומשנה את הנוכחית ע"י תוצאת הכפל ביניהם.

ט.פונ' multiply- מקבלת קואורדינטה ומשנה את הנוכחית ע"י תוצאת הכפל ביניהם.

2.point2D-נקודה במישור המכילה שתי קואורדינטות, מסוג coordinate.

א.בנאים ופונקציות טריויאליות.

ב. compareTo-משווה בין נקודות, מחזירה 0 אם שווים 1 אם גדול ו1- אם קטן ממנו.

ג. toString-מחזירה את ערך הנקודה במישור בפורמט הדפסה.

ד -distanceBetween2DPoints.מקבלת נקודה ומחזירה את המרחק ביניהן.

3.point3D-נקודה במרחב המכילה שלוש קואורדינטות, מסוג coordinate.

תיאור-המחלקה point3Dיורשת מpoint2D ומוסיפה ערך של קואורדינטה שלישית.

א. compareTo-משווה בין נקודות, מחזירה 0 אם שווים 1 אם גדול ו1- אם קטן ממנו.

ב. toString-מחזירה את ערך הנקודה במרחב בפורמט הדפסה.

ג.divide-מחלק את הוקטור במספר עשרוני.

ד.add-הוספה של וקטור.

ה.subtract-חיסור וקטור.

ו.distance- מקבלת נקודה ומחזירה את המרחק ביניהן.

4.vector-ישר היוצא מראשית הצירים מקבל את כיוונו ע"י נקודה. מכיל משתנה \_head מסוג point3D.

א.בנאים ופונקציות טריויאליות.

ב. compareTo- ההשוואה בפונקציה זו מתבססת על ההשוואה ב Point3D

לפי אורך הוקטור.

ג. toString-משתמשת בtoString של point3D.

ד.add- הוספת וקטור לוקטור הנוכחי.

ה.subtract-חיסור וקטור מהוקטור מהנוכחי.

ו.scale-הכפלה של הוקטור בקבוע.

ז.crossProduct-מבצע כפל וקטורי ומחזיר את התוצאה.

ח. length-מחזיר אורך וקטור.

ט. normalize-נרמול וקטור.

י. dotProduct-מחזיר תוצאה סקאלרית בכפל בין שני וקטורים.

5.Ray-וקטור שלא עובר בראשית הצירים והוא מוגדר ע"י נקודה וכיוון.

מכיל Point3D \_POO ו Vector \_direction-

א.בנאים ופונקציות טריויאליות.

ב. compareTo-משווה בין קרנות, ומחזירה 0 אם שווים 1 אם גדול ו1- אם קטן ממנו.

ג. toString-מחזירה את ערך הנקודה והכיוון בפורמט הדפסה.

6.material-מחלקה שאחראית על שימוש בחומר. מכילה משתנים כדלהלן:

א. double \_Kd-מקדם פיזור

ב. double \_Ks-מקדם הנחתה נייטרלי.

ג. double \_Kr-מקדם השתקפות (מראה אחת).

ד. -double \_Kt מקדם השתקפות(עבור שקף)

ה. double \_n- מקדם השבירה.

בנוסף ישנם בנאים ופונ' טריויאליות.

**Primitives:**

1. Coordinate - יחידה על ציר המספרים.

2.point2D-נקודה במישור.

3.point3D- נקודה במרחב.

4.vector-ישר היוצא מראשית הצירים מקבל את כיוונו ע"י נקודה.

5.Ray-וקטור שלא עובר בראשית הצירים והוא מוגדר ע"י נקודה וכיוון.

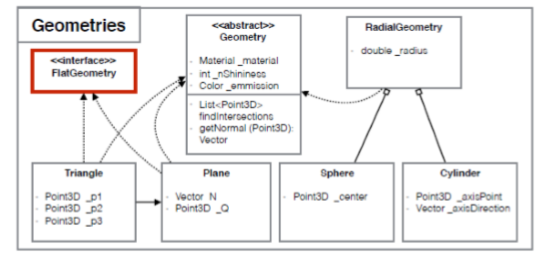
6.material-מחלקה שאחראית על שימוש בחומר.

Geometries

בעזרת חבילת הפרימיטיביים שהגדרנו נוכל עכשיו להגדיר אוסף של מחלקות בעזרתם נוכל לתאר צורות וגופים הנדסיים.

נגדיר חבילה בשם **geometries** שיש לה אוסף מחלקות המתארות צורות וגופים גיאומטריים, ע"מ לבנות סצנה גרפית מגופים מגוונים נבנה את החבילה בצורה היררכית.

מבנה:



מחלקות חבילת geometries:

מחלקות אבסטרקטיות:

1. -Geometryמחלקה אבסטרקטית עבור גוף גיאומטרי מסויים (מחלקות שונות של הגופים השונים ירשו ממנה). המחלקה תכיל בנאי ברירת מחדל והעתקה, 3 משתנים של סוג החומר, רמת ההברקה והצבע. כמו כן סטרים וגטרים.

פונקצייה אבסטרקטית **getNormal** המקבלת פרמטר יחיד של נקודה ומחזירה וקטור אנך לנקודה זו.

פונקצייה אבסטרקטית **findintersection** שמקבלת משתנה מסוג Ray ומחזירה רשימה של נקודות החיתוך עם הצורה הגיאומטרית.

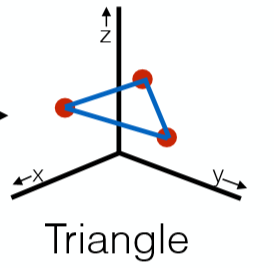
2. RadialGeometry-מחלקה אבסטרקטית היורשת מgeometry וכוללת בנאים (בנאי ברירת מחדל שמגדיר את הרדיוס באורך 0 ובנאי שמקבל משתנה עשרוני) שדה רדיוס וגטר עבורו. תפקידה לייצג צורה גיאומטרית מעגלית.

ממשק FlatGeometry-ממשק שמייצג צורות שטוחות לא מכיל פונקציות.

מחלקות הגופים המגוונים:

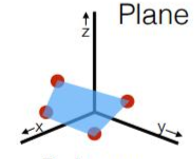
המחלקות שנמנה יורשות מהמחלקות האבסטרקטיות. **צורות שטוחות**:

1.משולשtriangle/-3 נקודות במרחב. מממשת את הממשק השטוח. ישנם 3 בנאים (בנאי ברירת מחדל, ההעתקה ובנאי המקבל שלוש נקודות) גטרים,סטרים, **findintersection** ו-GetNormal.



2. מישורplane/- נקודה במרחב ווקטור אנך. ישנם שני משתנים:vector המייצג נורמל למישור ונקודה במישור. בנאי ב"מ, העתקה והשמה.גטרים וסטרים.

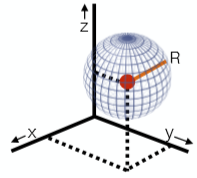
פונ' **findintersection** המייצגת את נקודת החיתוך של הקרן עם המישור.



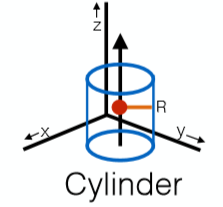
צורות תלת ממדיות:

1.כדורsphare/-נקודה עם רדיוס. המחלקה יורשת מ **RadialGeometry**- יש משתנה אחד המייצג את מרכז המעגל, ישנם שלושה בנאים גטרים סטרים. פונקציות: GetNormal-המחזירה וקטור נורמל לנקודה בכדור שהתקבלה כפרמטר.

**Findintersection**-המחזירה רשימה של נקודות חיתוך עם הכדור.



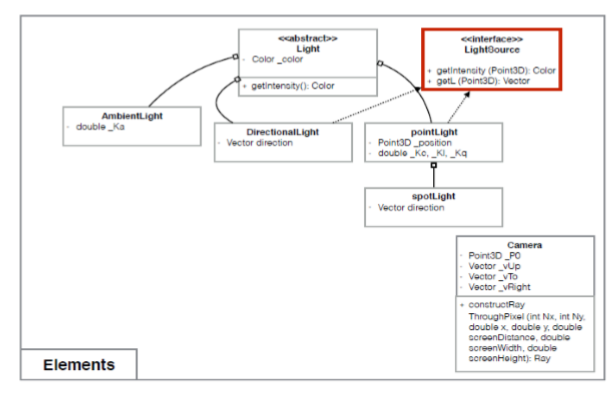
2.גלילcylinder/-קרן גובה ורדיוס. המחלקה יורשת מ **RadialGeometry-** ישנם שני משתנים אחד המייצג וקטור של כיוון הגליל ושני המייצג נקודה בגליל.ישנם סטרים, גטרים ושלושה בנאים. כמו כן פונקציות GetNormal ו- **Findintersection.**



Elements

אחרי שהגדרנו את המודל התלת-ממדי שאותו רוצים לייצג ,בשלב זה של הפרויקט נרצה להגדיר את המצלמה שהיא נקודת המבט שלנו על המודל שלנו ,על הצורות הגיאומטריות ומקורות האור שמשתמשים בהם בסצנה.

מבנה:



מחלקות חבילת **elements**:

1.- Cameraהמצלמה מכילה:

א. Point3D \_P0-נקודה תלת ממדית שהיא מרכז המצלמה המשקיף על המודל מראשית הצירים.

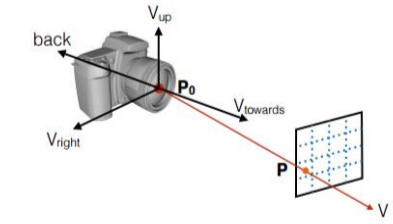
ב. ממרכז המצלמה יוצאים שלושה וקטורים שהם הצירים של המערכת- וקטורי הכיוון:

1.Vup- וקטור בכיוון החיובי של המערכת כלפי מעלה, ההמשך שלו בכיוון ההפוך הוא הווקטור שמצביע כלפי מטה.

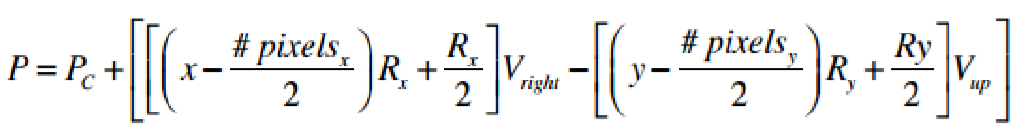
2.Vright-וקטור לכיוון הצד הימני של המערכת, ההמשך שלו בכיוון ההפוך הוא הווקטור שמציין את הצד השמאלי.

3.Vtoward-וקטור שכיוונו לקראת מישור הצפייה, ההמשך שלו בכיוון ההפוך הוא הווקטור שפונה לכיוון ההפוך ממישור הצפייה.

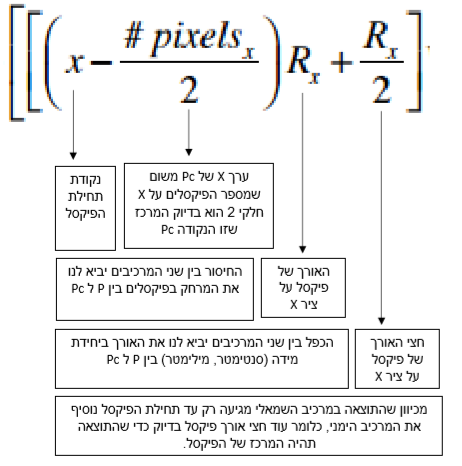
המחשה:



ג. constructRayThroughPixel- פונקציה שהיא בונה קרן דרך פיקסל, כלומר מישור הצפייה שלנו בנוי ומחולק לפי פיקסלים כאשר השטח הוא H\*W. אנו נרצה לשלוח וקטור ממרכז ההקרנה בכיוון מסוים לעבר מישור הצפייה כאשר הוא עובר דרך פיקסל מסוים.

הנוסחה היא:   

הסבר:



ד.בנאי שמקבל שלושה ערכים ומיישם אותם במשתנים המקומיים למחלקה.

ה.פונקציות גטרים סטרים toString שמדפיסה את נתוני המצלמה.

2.אור/light- מחקה אבסטרקטית המייצגת אור.

למחלקה יש משתנה אחד מטיפוס color כמו כן ישנם שני בנאים: א.בנאי ברירת מחדל שמגדיר צבע שחור. ב. בנאי שמקבל צבע ומבצע השמה למשתנה הצבע של המחלקה.

פונקציית **getIntensity**- מחזירה ערך מטיפוס color, שמייצג את האור ע"פ צבעו.

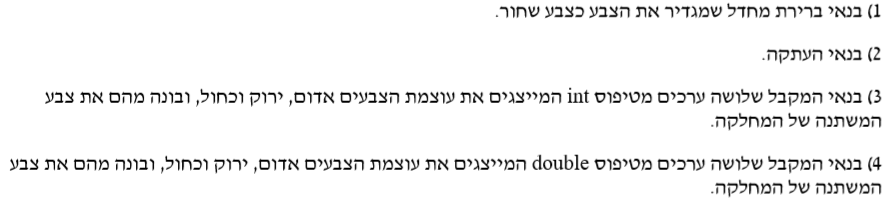
Interface: LightSource-ממשק שמייצג מקורות אור חיצוניים.

getIntensity-פונקצייה שמקבלת טיפוס מסוג point 3D ומחזירה ערך מטיפוס color-שמייצג את עוצמת האור.

getL- פונקצייה שמקבלת טיפוס מסוג point 3D ומחזירה וקטור ממקור האור לנקודה זו.

AmbientLight– מחלקה שמייצגת אור סביבתי, יורשת מlight-. ישנו משתנה שמייצג קבוע שמחליש את האור.

למחלקה ישנם ארבעה בנאים:



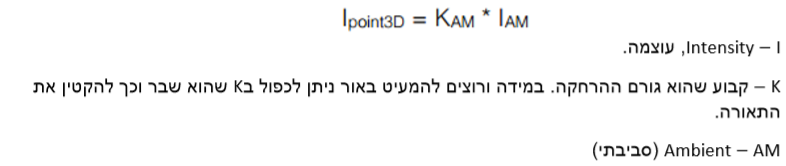
פונקציות גטרים וסטרים, ומימוש לממשק equal.

getIntensity-פונקצייה שמקבלת טיפוס מסוג point 3D ומחזירה ערך מטיפוס color-שמייצג את עוצמת האור ע"י הכפלה.

פונקציית toString ע"מ לבצע הדפסת האובייקט.

Intersectable interface:

מכיוון שאנו צריכים למצוא נקודות חיתוך עם כל גיאומטריה, נגדיר ממשק(interface) שבו יש פונקציה וירטואלית טהורה בשם FindIntersections (מציאת נקודות חיתוך), שמחזירה רשימה של נקודות תלת-ממדיות. ברשימה נמצאות כל נקודות החיתוך של קרן מסוימת עם הגיאומטריה שמממשת את אותו ממשק.

תאורה סביבתית-מקור האור הסביבתי מייצג מקור אור בעל עוצמת אור גבוהה ומשפיע על כל האובייקטים בסצנה שווה. ישנה נוסחה: 

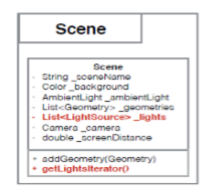
הסבר-הנוסחה אומרת שהעוצמה של התאורה בנקודה מסוימת () שווה לקבוע ההרחקה של התאורה הסביבתית כפול העוצמה של התאורה הסביבתית.

לאחר הביצוע, כל האובייקטים בסצנה מוארים בצבע ובעוצמה שצוינו, הסוג הפשוט ביותר של תאורה מיושמת, ומודלים- ללא תלות בכמה הם דלילים בפיזורם או כמות הפעמים שהם משתקפים- יוצרים אפקט אחיד.

**Scene**

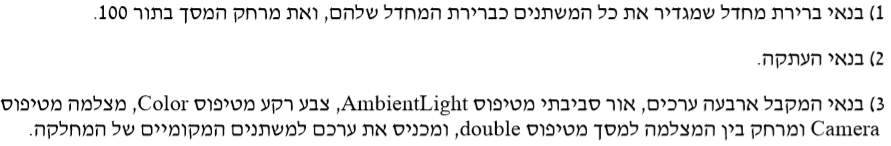
חבילת scene מייצגת את מכלול הסצנה על ידי ריכוז הצורות הגיאומטריות בתמונה, הרקע והתאורה.

מבנה:



מחלקת scene

המחלקה מייצגת סצנה. למחלקה יש שבעה משתנים, אחד מטיפוס צבע המייצג את צבע הרקע של הסצנה, אחד מסוג ambientlight המייצג את התאורה הסביבתית של הסצנה. אחד מטיפוס LIST<GEOMTRY> שמייצג את רשימת הצורות הגיאומטריות,אחד מטיפוס camera, אחד המייצג את המרחק מהמצלמה למסך, אחד מטיפוס LIST<LightSource> המייצג את רשימת מקורות האור ואחרון מטיפוס string שהוא השם של הסצנה.

המחלקה עם שלושה בנאים: פונקציות גטרים וסטרים.

addGeometry- מקבלת פרמטר גיאומטרי ומכניסה אותו לרשימת הצורות הגיאומטריות. getGeometriesIterator- מחזירה איטרטור לרשימת הצורות הגיאומטריות של הסצנה.

מימוש לממשק equal ו-toString.

renderer

מחלקת render- מחלקה זו מייצרת לנו את התמונה ע"י קבלת הנתונים ממחלקת הסצנה ועיבודם עד הצגת התמונה בשלמותה, הצגת התמונה כוללת פרטים רבים כמו הצורות הגיאומטריות הקיימות בסצנה, המרחקים מכל צורה וצורה, מציאת המרחק הקטן ביותר, הצבע לכל צורה, האורות המשפיעים ביחס לצבע ועוד.

במחלקה יש שלושה משתנים: משתנה המייצג את הסצנה של התמונה, אחד מסוג ImageWriter שמייצג את המדפיס של התמונה וקבוע מסוג אינטננג'ר שמייצג את רמת הרקורסיה.

ישנו בנאי אחד המקבל שני ערכים ומכניס את ערכם למשתני המחלקה.

renderImage-פונקצייה שמקבלת את נקודות החיתוך ומוצאת את הצבע המתאים לכל פיקסל.