**שלב 4 - חבילת אלמנטים, מצלמה, יצירת קרניים**

לפני הכול נוודא ש"הקפאנו" את השלב הקודם שהגשנו עם תג (כולל שליחת התג למאגר המרוחק בענן בגיט-האב או בגיט-לאב).

בשלב הזה של העבודה נתחיל לבנות תשתית עבור סצנה. נתחיל מבניית מצלמה ויצירת קרניים ממנה דרך ה-**View Plane**.

1. נוסיף חבילה חדשה בשם **elements**, בה נתחיל להוסיף את האלמנטים של סצנה.
2. עבור המצלמה, בחבילה החדשה נוסיף מחלקה חדשה **Camera**. נוסיף במחלקה:

* שדות רלוונטיים בהרשאה פרטית (מיקום מסוג **Po****int3D** ושלושה וקטורים של כיוון המצלמה בהתאם – עבור , ו-, ו-) עם פעולות מאחזרות (גטרים).
* בנאי עם פרמטרים עבור ערכי המיקום ושני וקטורים של כיוון – קדימה ומעלה ( ו-). על הבנאי בין היתר לוודא שהוקטורים ו- מאונכים, וליצור וקטור .
  + יש לדאוג שכל הווקטורים השמורים עם המצלמה יהיו מנורמלים
  + אסור ליצור בנאי ברירת מחדל עבור מצלמה – מצלמה חייבת להיות ממוקמת במחרב במפורש בעת יצירתה
  + אסור ליצור בנאי העתקה עבור מצלמה
* מתודת עדכון (סטר) עבור גודל ה-**View Plane**, המקבלת שני מספרים – רוחב וגובה (לפי הסדר הזה) ומחזירה את אובייקט המצלמה בעצמו (**this**) – זהו אלמנט מועיל מתבנית עיצוב **Builder**. החתימה כדלהלן:

**public Camera setViewPlaneSize(double width, double height)**

* מתודת עדכון עבור מרחק ה-**View Plane** מהמצלמה, המחזירה את אובייקט המצלמה בעצמו (**this**), כנ"ל:

**public Camera setDistance(double distance)**

* מתודה **constructRayThroughPixel** כפי שנלמד בקורס. בשלב הראשון הפונקציה תחזיר **null**, למחלקה החדשה ולכל המתודות שלה (כולל הבנאי, הגטרים והסטרים אם יש כאלה) ייכתב תיעוד בפורמט **javadoc**. החתימה כדלהלן:

**public Ray constructRayThroughPixel(int nX, int nY, int j, int i)**

***נ.ב.1 שימו לב שה-View Plane מיוצג בפועל בשני אפיקים – גודל ומרחק מהמצלמה – במצלמה עצמה, והרזולוציה – ע"י הפרמטרים של המתודה הנ"ל, כאשר האינדקסים של פיקסל מיוצגים על ידי הפרמטרים i ו-j – שורה ועמודה של פיקסל בהתאם. כל קרן מיוצרת דרך מרכז פיקסל מתאים במטריצת ה-View Plane*.**

***נ.ב.2 שימו לב שברזולוציה – Nx מייצג כמות עמודות (רוחב שורה) ו-Ny מייצג כמות שורות (גובה עמודה). לכן לא להתבלבל בסדר הפרמטרים (והארגומנטים) של רזולוציית המטריצה ושל האינדקסים של הפיקסלים!*** *הסיבה לכך שברישום הקואורדינטות אנחנו מקדימים את ציר ה-****X*** *(קואורדינטה אופקית) לציר ה-****Y*** *(קואורדינטה אנכית), אבל בסריקת מטריצות (של פיקסלים למשל) אנחנו מקדימים את אינדקס השורה (אינדקס אנכי) לאינדקס העמודה (אינדקס אופקי)*

***נ.ב.*** *לבונוס נפרד כללי (לא מצורף לתרגיל זה – ניתן לעשותו בכל עת ולקבל נקודת בונוס), ניתן ליצור בקמרה מתודת הזזה וסיבוב, אבל אין לעשות זאת לפני השלמת כל המטלה הנוכחית, המתודה חייבת להיות מבוססת על שימוש בשיטות שנלמדו באלגברה לינארית (בסופו של דבר הבונוס יינתן בפועל לאחר שתגיעו לשלב בניית תמונות, ותוכלו להציג "צילומים" [תמונות] מנקודות מבט וזוויות שונות)*

1. בחבילה **unittests** נוסיף מחלקת בדיקות **CameraTests** (המודול של המחלקה מצורף לתרגיל – יש לקחתו ולהציב בחבילת הבדיקות – אם שם החבילה אצלכם שונה – יש לעדכן את שם החבילה בפקודת **package** במודול). במחלקה הזו מופיעות בדיקות יחידה (**unit tests**) עבור הפונקציה הנ"ל בהתאם למה שנלמד בקורס לגבי **TDD** ומפורט במצגת המעבדה, כמו כן בדיקות הבנאי.
2. יש ליצור מחלקה נוספות בחבילה **unittests** לבדיקות אינטגרציה בין יצירת קרניים ממצלמה לבין חישוב חיתוכים של קרן עם גופים גאומטריים מהשלב הקודם. במחלקה תהינה שלוש פונקציות בדיקה (עבור אינטגרציה מול ספירה, מישור ומשולש – הבדיקות כפי שמופיעות במצגת המעבדה). בבדיקות אינטגרציה אלה אין צורך לחלק את הבדיקות לפי **EP** ו-**BVA**.

* כל בדיקה תיצור קרניים עבור כל הפיקסלים ב-**View Plane** 3 על 3 פיקסלים, וגודל 3 על 3, תחשב חיתוך של כל קרן עם הגוף של הבדיקה ותסכם את כמות החיתוכים מכל הקרניים
* כפי שמובן מהנאמר בפסקה הקודמת - אין צורך לחשב את כל נקודות החיתוך במדויק ו-\או לצבור אותם ברשימה כוללת
* חובה להוסיף תעוד למחלקה ולכל פונקציית טסט ולכל בדיקה כמו בכל המודלים של בדיקות
* מיקום המצלמה, נתוני ה-**View Plane**, מיקום וגודל הגוף הגאומטרי של כל בדיקה יילקח מהשקפים של המעבדה, ואם חסרים שם פרטים – יושלמו ע"י הסטודנטים לפי התוצאה הנדרשת בטסט
* שימו לב שכדאי ליצור פונקציית עזר פרטית על מנת לא לחזור על אותו קוד פעמים רבות (לולאה של יצירת קרניים וספירת כמות חיתוכים). תחשבו איך לעשות את זה הכי חכם על מנת לחסוך לעצמכם עד כמה שאפשר עבודה מיותרת של כתיבת קוד שחוזר על עצמו בתוך הטסטים עצמם.
* הבדיקות חייבות לכסות לפחות את המקרים שמופיעים במצגת המעבדה ובמקרה הצורך על פי שיקול דעת הסטודנטים ניתן להוסיף בדיקות נוספות לפי החשיבה שלכם וע"פ דרישות ה-**TDD**

1. **רק לאחר כתיבת כל הבדיקות** יש לממש בפועל את הפונקציה **constructRayThroughPixel** כפי שנלמד בקורס התאורטי ומוצג בשקפים של המעבדה, ויש לוודא שכל הבדיקות מגיעות להצלחה מלאה - הן במודול של בדיקות יחידה שקיבלתם עם התרגיל והן בבדיקות האינטגרציה.

* במימוש הפונקציה יש לתת את הדעת למקרה של פיקסלים מרכזיים (כאשר במטריצה כמות אי זוגית של שורות ו-\או עמודות של פיקסלים), כפי שמופיע בשקף המתאים במצגת המעבדה
* **נ.ב.** האינדקסים של הפיקסלים מתחילים מ-0 עד לכמות הפיקסלים במימד המתאים פחות אחת

**הערה כללית להמשך**: כדאי ליצור מיקומים של מצלמה וגופים, כיוונים ומרחקים כך שיקלו עליכם את יצירת הטסטים ואת החישובים הדרושים לכך.

**בונוסים של התרגיל**:

* טרנספורמציית הזזה וסיבוב (1 נק') – הבונוס יינתן גם אם יתבצע בשלבים מאוחרים יותר

בהצלחה!