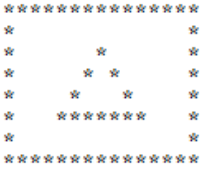
# תרגיל בית 1

תאריך הגשה:



## תיאור כללי:

בתרגיל תממשו כלי אנימציה פרימיטיבי (מאוד) המאפשר לצייר מספר צורות גיאומטריות פשוטות ולבצע עליהן טרנספורמציות.

להלן רשימת הצורות:

**קו רגיל, ריבוע** (מקביל לצירים), **פלוס** (שמן) **ומדרגות חיוביות**.

הסבר מפורט לגבי כל אחת מהצורות ינתן בהמשך המסמך.

## מה עליכם לעשות:

עליכם לממש 4 מחלקות: Stairs , Plus,  Square, Line

המחלקות צריכות להיקרא בדיוק בשמות האלה וכל אחת צריכה להיות מוגדרת בקובץ "h." בשם תואם. הממשק מוגדר בצורה מדויקת – **אין להשמיט או להוסיף פונקציות ציבוריות, השמות של הפונקציות הציבוריות צריכים להיות בדיוק כפי שהם מופיעים בהגדרת התרגיל ואין לחשוף משתנים פנימיים.**

בכל מובן אחר אתם חופשיים לממש את המחלקות כרצונכם (בכפוף כמובן לכללי התכנות הנאות). ז"א שאתם יכולים (וצריכים) להחליט בין השאר:

- איזה מידע יש להחזיק בתוך האובייקטים?
- איך לייצג מידע זה?
- איזה פונקציות פרטיות יש לממש?
- .....

## מה אתם מקבלים מאיתנו:

1. אתם מקבלים את הקובץ *Vertex.h* שבו הגדרה של struct שנקרא Vertex שמייצג קודקוד במישור עם קואורדינטות (x,y).שימו לב לכך שמערכת הקואורדינאטות פה אינה זהה לזו הנהוגה בשעורי המתמטיקה. ראשיתה בפינה השמאלית העליונה. ציר ה-X מתקדם ימינה וציר ה-Y מתקדם למטה, **אין לשנות את קובץ זה**.
2. קובץ שנקרא *macros.h* ובו מוגדרים המשתנים MAX_X ו- MAX_Y. אלה ערכי ה-X וה- Y המקסימליים שניתן לקבל. ערכי ה- X האפשריים בתרגיל הם המספרים בין 0 ל MAX_X וערכי ה-Y האפשריים בתרגיל הם המספרים בין 0 ל MAX_Y. כאשר מקבלים בבנאים נתונים שיגרמו לכך שהאובייקטים יחרגו מהתחום הנ"ל צריך ליצור אובייקט עם ערכי ברירת מחדל שיוגדרו בפירוט בהמשך, אתם יכולים להניח שערכי ברירת המחדל כפי שיוגדרו בהמשך נמצאים בתחום החוקי. **שימו לב לא להתייחס בקוד שלכם למספרים הספציפיים שמופיעים בקובץ אלא לקבועים הנ"ל**. הקובץ *macros.h* שאתם מגישים צריך להיות זהה לזה שקיבלתם.
3. קובץ שנקרא testPaint.cpp, המכיל תוכנית שמקבלת נתונים מהמשתמש ועל פיהם יוצרת אובייקטים מהמחלקות שאתם צריכים לממש, מוציאה פלט שמתאר את האובייקטים שנוצרו ומציירת אותם בקונסולה. אתם לא חייבים להגיש את הקובץ הזה ולכן כמובן שאתם יכולים לשנות אותו. הקובץ הזה הוא לא הtester שאיתו התרגיל ייבדק אבל מן הסתם הוא דומה לו. התוכנית נועדה רק להמחיש ולהדגים את צורת השימוש באובייקטים שאתם צריכים לממש. בתחילת התוכנית יש קבועים בוליאנים שמשפיעים על הפעולה של התוכנית:
   LINE,SQUARE,STAIRS,PLUS קובעים איזה אובייקטים לבדוק.

## פירוט הממשק:

עליכם לממש 4 מחלקות:

1. **Line** מייצגת קטע ישר.
2. **Square** מייצגת ריבוע מקביל לצירים
3. **Plus** מייצגת פלוס (שמן)
4. **Stairs** מייצגת מדרגות חיוביות.

## Line API:

### בנאים:

Line(Vertex ends0, Vertex ends1) - בנאי המקבל שני קצוות לקטע.

ends0 הוא תחילת הקטע וends1 הוא סוף הקטע – **לשמור על הסדר ביניהם**.

עבור **כל** הבנאים יש לוודא שערכי הX של שני הקצוות נמצאים בתחום [0,MAX_X] ושני ערכי הY נמצאים בתחום [0,MAX_Y].

אם **אחד מהפרמטרים** לא עונה על הקריטריון הזה עליכם לבנות Line שההתחלה שלו בנקודה (0,0) והסוף שלו בנקודה (10,10).

Line(Vertex ends[2]) – אותו דבר בדיוק כמו הבנאי הראשון אלא שמקבלים את הקצוות במערך(ends[0] הוא תחילת הקטע וends[1] הוא סוף הקטע).

Line(float x0, float y0, float x1, float y1)– בונה Line שבו תחילת הקטע הוא הנקודה (x0,y0) וסוף הקטע הוא (x1,y1).

Line(Vertex start, float length, float angle) – בנאי המקבל נקודת התחלה, אורך וזווית (במעלות) ביחס לציר הX החיובי ובונה על פיהם קטע. גם כאן צריך לבדוק אם שני קצוות הקטע נמצאים בטווח שהוגדר למעלה ואם לא, לפעול באותה צורה כמו בבנאים הקודמים.

### שיטות ציבוריות:

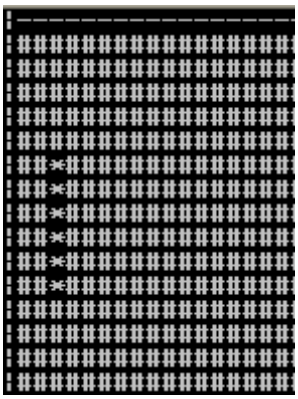
Vertex getEnd1() - מחזיר את ההתחלה של הקטע.

Vertex getEnd2() - מחזיר את הסוף של הקטע.

float getLength()- מחזיר את האורך של הקטע.

void draw(board[][MAX_Y+1]) -מציירת את הקו על המטריצה, עליכם לממש את ציור הקו על פי אלגוריתם (האופטימלי) של Bresenham . שימו לב, כשכתוב בפסאודו קוד from x0 to x1 הכוונה היא **כולל** x1. (על כן ההערה)

**דוגמה:** קו מנקודה (3,6) ועד לנקודה (3,11)



**שימו לב!!!**
**מכיוון שבעולם האמיתי לנקודה אין שטח ואילו כאן נקודה תופסת מקום נדמה כאילו הקו הוא באורך 6 למרות שהיינו מצפים שיתפוס 5 כוכביות. זהו מצב תקין**

## :Square API

### בנאים:

Square(Vertex topLeft,float size) – מקבל את הפינה השמאלית עליונה וגודל צלע ובונה ריבוע מקביל לצירים. יש לבדוק אם כל ארבעת הקודקודים נמצאים בתחום החוקי. אם לא אז צריך ליצור Square שארבעת הקודקודים שלו בנקודות (0,0),(10,0),(0,10),(10,10).

### שיטות ציבוריות:

void move(Vertex shift) – מזיז את הריבוע, ערכי הX של הריבוע צריכים לזוז ב shift._x וערכי הY של הריבוע צריכים לזוז ב shift._y. shift._x ו shift._y יכולים להיות שליליים. אם אחרי ההזזה הריבוע יוצא מהתחום שהוגדר למעלה צריך להשאיר את הריבוע כמו שהיה לפני הקריאה לmove.

Vertex getBoundingCircleCenter() - מחזיר את מרכז המעגל החוסם את הריבוע .

float getSize() – מחזיר את הרוחב של הריבוע.

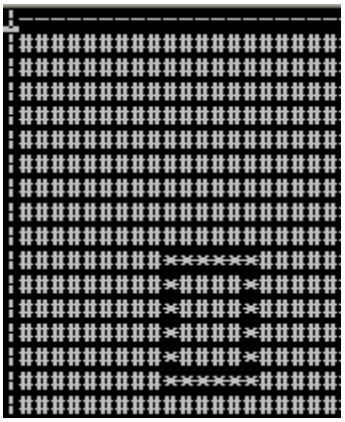
float getArea() – מחזיר את השטח של הריבוע.

float getPerimeter() – מחזיר את ההיקף של הריבוע.

void draw(board[][MAX_Y+1]) - מציירת את הריבוע

(**ניתן להשתמש בשיטת draw שבמחלקה Line**)

**דוגמה:** ריבוע בגודל 5X5 כאשר קודקוד שמאלי עליון בנקודה (10,10)



# Plus API:

## בנאים:

Plus(Vertex topLeft,float size) – מקבל את הפינה השמאלית עליונה (ראו בתחתית העמוד) וגודל צלע ובונה פלוס "שמן" מקביל לצירים. יש לבדוק אם כל קודקודיו נמצאים בתחום החוקי. אם לא יש צריך ליצור Plus בנקודה (10,10) בעל צלע בגודל 3.

## שיטות ציבוריות:

void move(Vertex shift) – מזיז את הפלוס, ערכי הX של הפלוס צריכים לזוז ב shift._x וערכי הY של הפלוס צריכים לזוז ב shift._y. shift._x ו shift._y יכולים להיות שליליים. אם אחרי ההזזה הפלוס יוצא מהתחום שהוגדר למעלה צריך להשאיר את הפלוס כמו שהיה לפני הקריאה לmove.

Vertex getTopLeft() - מחזיר את הנקודה השמאלית העליונה של הפלוס.

float getSize() – מחזיר רוחב צלע בודדת של הפלוס.

float getArea() – מחזיר את השטח של הפלוס.

float getPerimeter() – מחזיר את ההיקף של הפלוס.

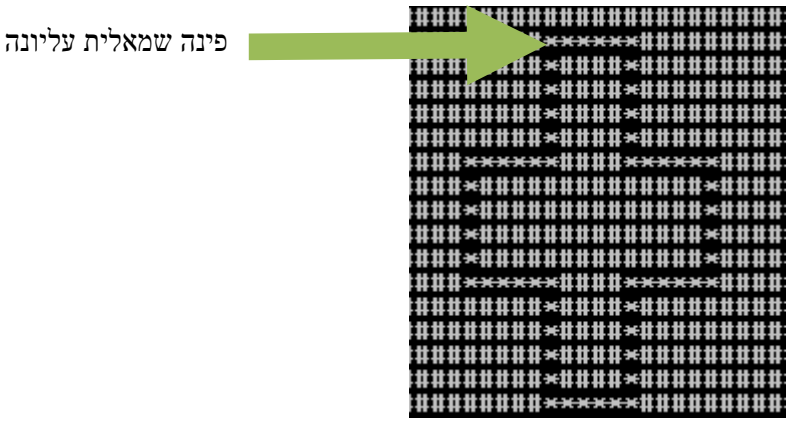
void grow(float delta) – מגדיל כל אחת מצלעות הפלוס ב delta, הערך יכול להיות שלילי. אם אחרי ההגדלה הפלוס יוצא מהתחום שהוגדר למעלה צריך להשאיר את הפלוס כמו שהיה לפני הקריאה ל grow.

void draw(board[][MAX_Y+1]) מציירת את הפלוס

(**ניתן להשתמש בשיטת draw שבמחלקה Line**)

## דוגמה:

פלוס (שמן) עם צלע בגודל 5 ונקודה שמאלית עליונה ב(15,15)



## Stairs API:

### בנאים:

Stairs(Vertex bottomLeft,float height,float width,int numOfStairs) – מקבל את הפינה השמאלית התחתונה (ראו בתחתית העמוד) את גובה המדרגה ,את רוחבן ואת מספר המדרגות יש לבדוק אם כל הקודקודים נמצאים בתחום החוקי. אם לא יש צריך ליצור מדרגה אחת בנקודה (10,10) בעל צלע גובה 2 ורוחב 2.

### שיטות ציבוריות:

void move(Vertex shift) – מזיז את המדרגות, ערכי הX של המדרגות צריכים לזוז ב shift._x וערכי הY של המדרגות צריכים לזוז ב shift._y. shift._x ו shift._y יכולים להיות שליליים. אם אחרי ההזזה המדרגות יוצאות מהתחום שהוגדר למעלה צריך להשאיר את המדרגות כפי שהיו לפני הקריאה לmove.

Vertex getBotLeft() - מחזיר את הנקודה השמאלית התחתונה של המדרגות.

float getHeight() – מחזיר את הגובה של המדרגות.

float getWidth() – מחזיר את הרוחב של המדרגות.

bool rotate(int angle) – מסובב את המדרגות ביחס לנקודה השמאלית התחתונה של המדרגות אם הסיבוב מוציא את המדרגות מהתחום
השיטה תחזיר false ותחזיר את המדרגות למצב הקודם לפני הקריאה לrotate .
המעלות ניתנות בכפולות של 90 מעלות (ניתן לקבל גם ערך שלילי)

void draw(board[][MAX_Y+1]) מציירת את המדרגות
(**ניתן להשתמש בשיטת draw שבמחלקה Line**)

### דוגמה:

מדרגות חיוביות עם רוחב 3, גובה 3 ומספר מדרגות של 2 בנקודה (15,15)

## קובץ הREADME:

יש לכלול קובץ README שיקרא README.doc או README.txt (ולא בשם אחר)

קובץ זה יכיל לכל הפחות:

1. כותרת
2. פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.
3. הסבר כללי של התרגיל.
4. רשימה של הקבצים שנוצרו ע"י הסטודנט, עם הסבר של שתי שורות לכל היותר לגבי תפקיד הקובץ.
5. מבני נתונים עיקריים ותפקידיהם.
6. אלגוריתמים הראויים לציון.
7. נקודות מעניינות.
8. באגים ידועים.

יש לתמצת ככל שניתן אך לא לוותר על אף חלק (אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת ומתחתיו פסקה ריקה). תכתבו בREADME כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

## אופן ההגשה:

הקובץ להגשה: יש לדחוס כל קובץ הקשור לתרגיל למעט ספריות הdebug לקובץ ששמו exN_ID.zip כאשר N הוא מספר התרגיל וID הוא מספר תעודת הזהות של המגיש כולל כל הספרות.

לפני דחיסת תיקיית הProject שלכם יש למחוק את הפריטים הבאים:

- קובץ בעל סיומת ncb שאמור להיות בתיקית המקור של הפרויקט.
- תיקיה בשם debug שאמורה להימצא בתיקית המקור של הפרויקט.
- כל תיקיה נוספת בשם debug הנמצאת בתוך אחת התיקיות הפנימיות בפרויקט.

עליכם להגיש את הקבצים Vertex.h ו my_macros.h כפי שקיבלתם אותם.

את הקובץ Line המעודכן ואין צורך להגיש את testPaint.cpp.

את הקובץ יש לשלוח במייל לכתובת **oophadassa@gmail.com** כשהנושא זהה לשם הקובץ המצורף.

כל שינוי ממה שמוגדר פה לגבי צורת ההגשה ומבנה הREADME עלול לגרור הורדת נקודות בציון.

## בהצלחה!