**גרפיקה ממוחשבת תרגיל 3**

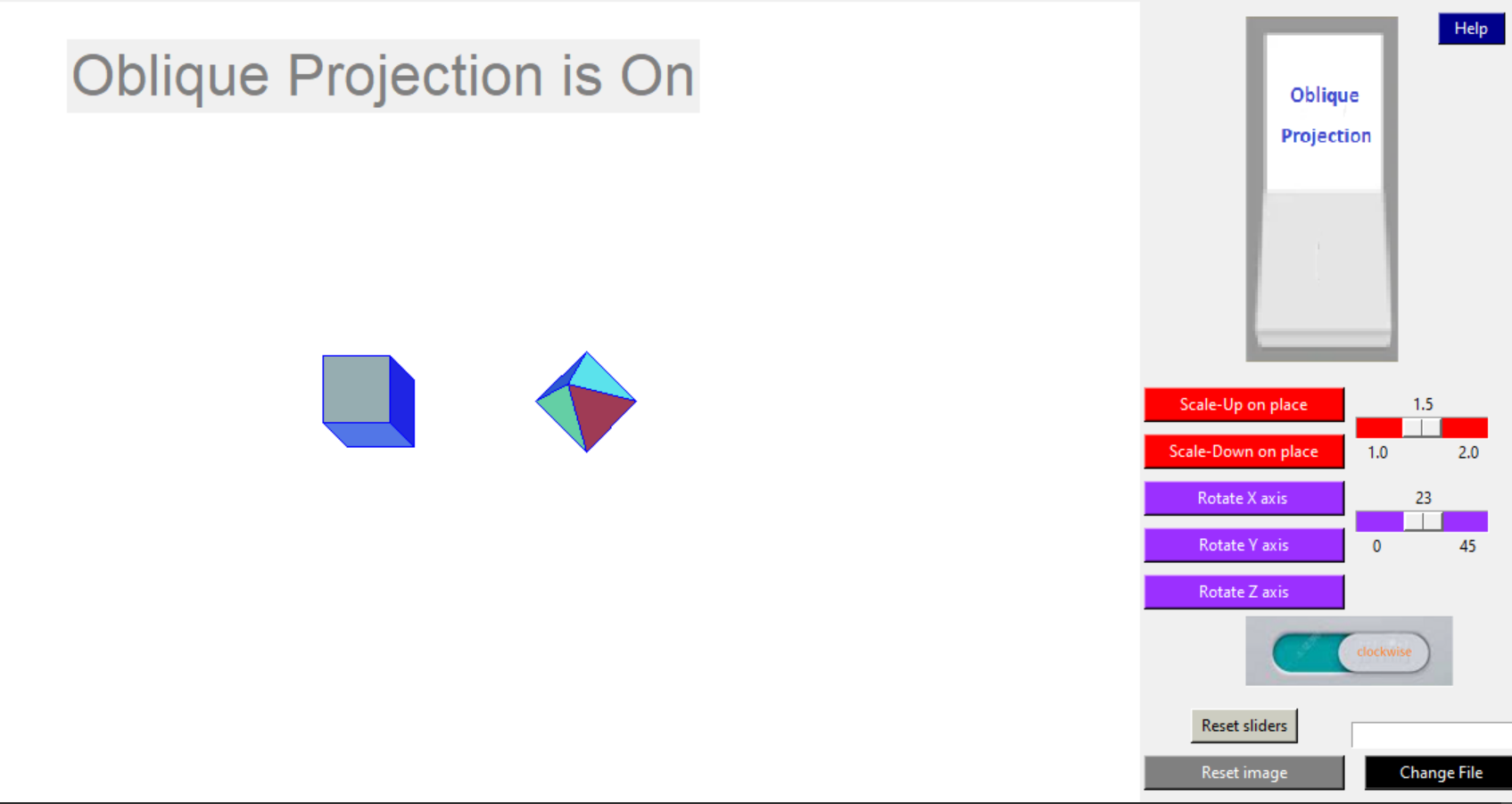
**מגישים:**

קורל רובילר 316392877

דוד ציבולסקי 309444065

אומרי חכם 308428226

**תיאור קצר:**



המערכת מציגה צורות תלת מימדיות ומאפשרת פעולות של הגדה וסיבוב של הצורות התלת מימדיות על פי אלגוריתמים שלנמדו בהרצאה.

ניתן לבצע פעולות על ידי הכפתורים שמוצגים בממשק המשתמש.

בפעולות כמו סיבוב ניתן לשנות את ערכי הסיבוב על ידי הסליידר לפי הטווח הנתון.

המערכת מחולקת למודולים של הטלות, טרנספורמציות, מחלקות של פוליגון ונקודות ומודול ראשי(ex3).

**קבצים מוכנים לשימוש:**

* data.txt

(ניתן ליצור ולהוסיף קובץ נוסף כרצונכם)

**חשוב**-קבצי הפרמטרים צריכים להיות בתוך תיקיה של פרמטרים (Params) – **מצורפת**

**מבנה לוגי של הנתונים:**

את הנתונים אנו שומרים בתוך רשימה (List) של פוליגונים(class ) שמכיל נקודות תלת מימדיות (class).



תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

הקובץ שמתאר את צורתו של האובייקט התלת מימדי מסודר לפי השם לאחר מכן רשימה של פוליגונים ולבסוף רשימה של נקודות מהם מורכבים הפוליגונים (מופרדים בפסיקים ובירידות שורה):

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

כך אנו קוראים את הקובץ בצורה הנכונה על מנת להציג את הצורה למשתמש בהמשך.

התוכנית קוראת את הקובץ, מעבדת את הנתונים ומתאימה את ערכי הציור למסך על ידי מרכוז והגדלה (ע"י פונקציה שמתוארת למעטה)

כמובן שהתוכנית בודקת את נכונות ערכי הקבצים ואם הם קיימים.

**פונקציות שנכתבו על ידנו:**

**Polygon-** מחלקה המחזיקה את הנקודות שמגדירות את הפוליגון

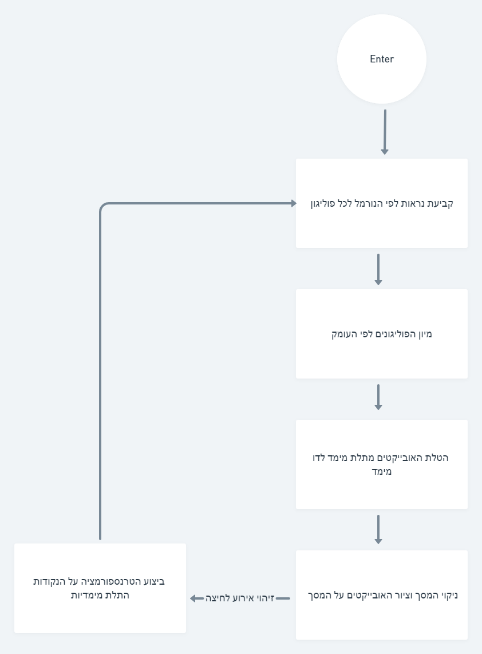
**Point-** מחלקה המתארת נקודה תלת מימדית

**פונקציות שנכתבו על ידנו:**

(הסבר מפורט יותר בקובץ הקוד)

1. **getPoligonsLstFromFile()**- קורא ומעבד את הקובץ הנתון, שומר את התוכן לרשימת הפוליגונים.
2. **change\_file\_name(my\_canvas, text\_box)**– משנה את שם הקובץ במידה והקובץ קיים ואין בו שגיאות
3. **setVisibility()**– פונקציה העוברת על כל הפוליגונים מחשב את הנורמל ומעדכן האם יש להראות את הפוליגון
4. **myPolygonSort(poly)**– פונקציה הממיינת את נקודות הפוליגון לפי עומק
5. **sortPoligons()**– פונקציה הממיינת את הפוליגונים לפי עומק
6. **scale\_func(my\_canvas, S)**– פונקציה המקבלת את ערך הסילום ,שולחת את הערכים לפונקציה ייעודית לסילום ולאחר מכן מציגה את התוצאה
7. **rotate\_func(my\_canvas, theta, axis="x")**- פונקציה המקבלת את ערך הסיבוב ,שולחת את הערכים לפונקציה ייעודית לסיבוב ולאחר מכן מציגה את התוצאה
8. **display\_error(msg: str)**– הצגת שגיאה למסך
9. **reset\_canvas(my\_canvas)**– פונקציה שמאפסת את רשימת הפוליגונים, קוראת אותם מחדש ומציגה מחדש.
10. **Switch\_clock\_wise()**– החלפת כיוון הסיבוב
11. **Switch\_Projection(my\_canvas)** – החלפה בין סוגי ההטלות
12. **draw\_poligons(my\_canvas , list\_2D\_Polygon)**– מעבר על רשימת הפוליגונים הדו ממדיים וציורם על המסך
13. **displayShapes(my\_canvas)**– הצגת הפוליגונים לפי ההטלה הנבחרת
14. **perspective\_projection(list\_3D\_Polygons: list)** – המרת הנקודות התלת ממדיות לדו מימד לפי סוג ההטלה
15. **oblique\_projection(list\_3D\_Polygons:list , angle=45)** – המרת הנקודות התלת ממדיות לדו מימד לפי סוג ההטלה
16. **rotate(list\_3D\_Polygons:list, theta,clockWise, axis="x")** – סיבוב הצורה לפי הכיוון והציר
17. **scale(list\_3D\_Polygons, S)**– סילום לפי הערך שהועבר
18. \_**\_main\_\_ -** אתחול המסך, הציור המבוקש ויצירת ממשק המשתמש כאשר אנו מחברים בין הכפתורים לפונקציות המתאימות

**מבנה לוגי של התוכנית:**

****

**שגיאות:**

1. קובץ לא תקין וקובץ לא נמצא (incorrect file name) – במידה וערכי הקובץ היו לא תקינים (נקודה-float וpolygon-int)

**תודה רבה קבוצה V !**