

גרפיקה ממוחשבת תרגיל 2

מגישים:

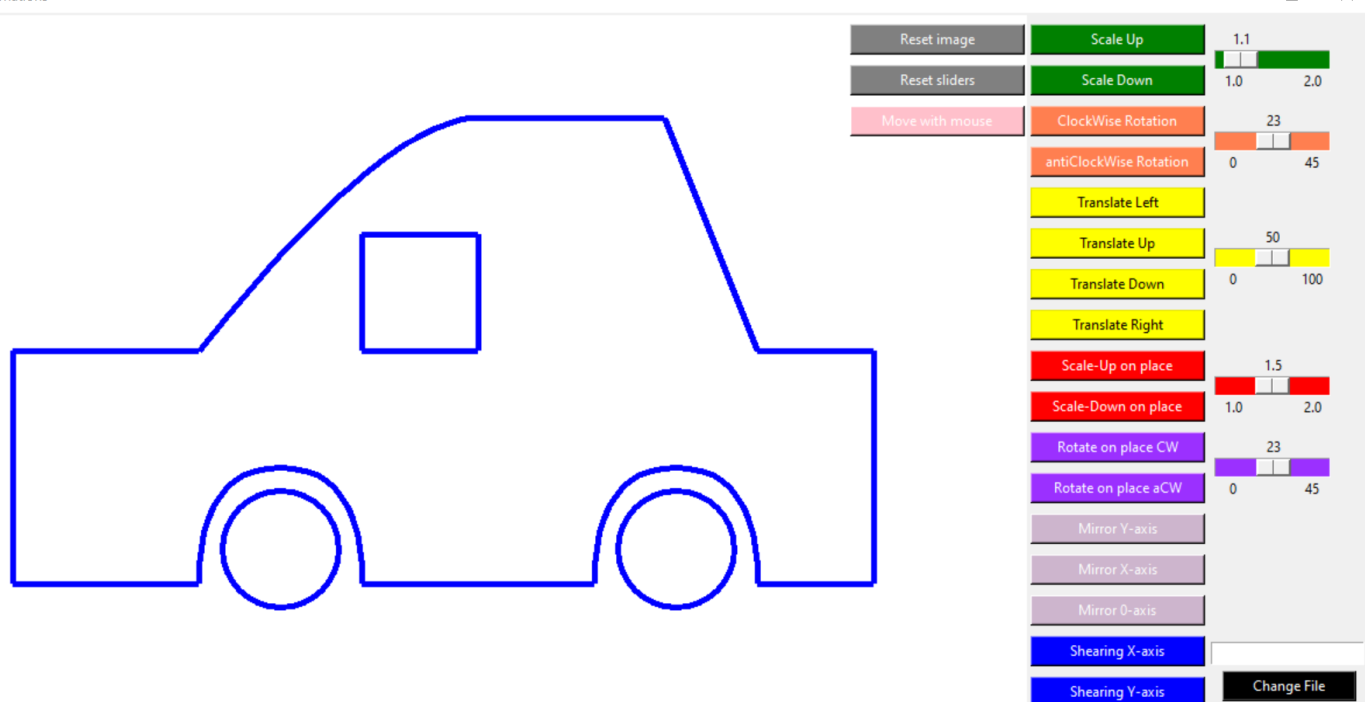
קורל רובילר 316392877

דוד ציבולסקי 309444065

אמרי חכם 308428226

תיאור קצר:

fargil 2 - Transformations



ניתן לבצע פעולות על ידי הכפתורים שמוצגים בממשק המשתמש.

בפעולות כמו סיבוב ניתן לשנות את ערכי הסיבוב על ידי הסליידר לפי הטווח הנתון במידה והציור לא יחרוג מגבולות המסך, כמובן. בפעולות כמו גזירה והזזה עם העכבר, יש צורך ללחוץ על הכפתור ואז לגרור עם העכבר (לפי ההוראות שניתנו).

קבצים מוכנים לשימוש:

- car_params.txt
- Coral_car_param.txt
- new_car.txt

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

מבנה לוגי של הנתונים:

את הנתונים אנו שומרים בתוך טבלת גיבוב (python dictionary) שמכיל מערכים של צורות לפי הקטגוריה, כשכל מערך כזה הוא dictionary.

```
Lines = []
Circles = []
BezierCurves = []
shape = {"lines": False, "Circles": False, "BezierCurves": False}
```

הקובץ מופרד לקווים, מעגלים ועקומות בצורה הבאה:

```
lines:
100, 300, 100, 400; 100, 300, 180, 300; 100, 400, 180, 400;
300, 200, 380, 200; 250, 400, 350, 400; 380, 200, 420, 300;
420, 400, 470, 400; 420, 300, 470, 300; 470, 400, 470, 300;
250, 250, 300, 250; 250, 250, 250, 300; 250, 300, 300, 300;
300, 300, 300, 250;
Circles:
215, 385, 25; 385, 385, 25;
BezierCurves:
180, 300, 220, 250, 280, 200, 300, 200; 180, 400, 180, 350, 250, 350, 250, 400;
350, 400, 350, 350, 420, 350, 420, 400;
```

בתוך כל קטגוריה, כל צורה מיוצגת לפי מספר הפרמטרים הנדרשים (קו-2 נקודות, מעגל – נקודה ורדיוס, עקומה – 4 נקודות) ומפריד נקודה פסיק (;) בין כל צורה וצורה בקטגוריה.

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 מרכז והגדלה (ע"י פונקציה שמתוארת למעטה)

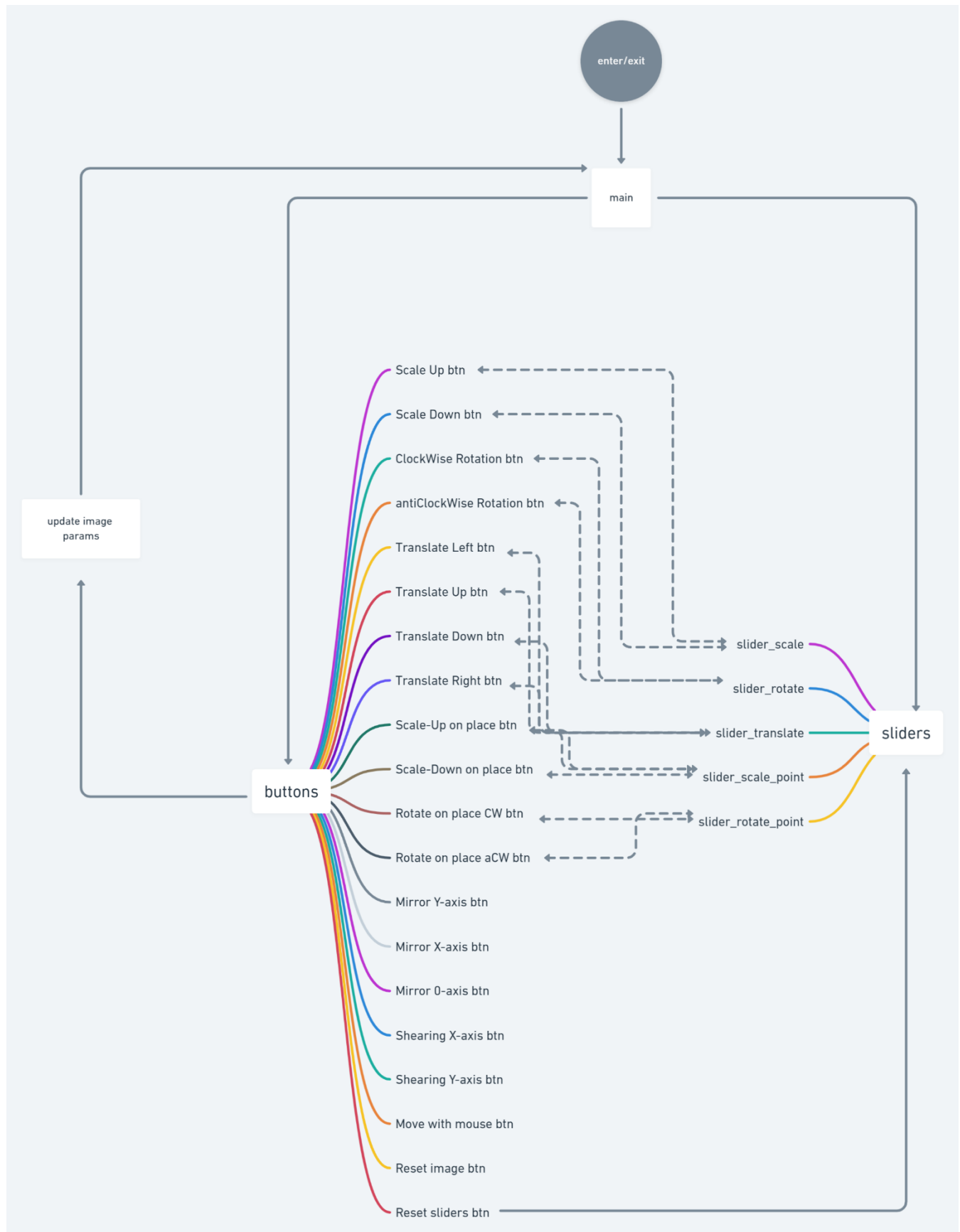
פונקציות שנכתבו על ידנו:

(הסבר מפורט יותר בקובץ הקוד)

1. write_to_the_screen(my_canvas, my_text, color) – כותב טקסט בצבע נדרש למסך
2. def clear(my_canvas) – מנקה את המסך והופך את כולו ללבן
3. create_circle(x, y, r, my_canvas) – יוצר מעגל במסך לפי נקודת המרכז ורדיוס
4. getImageParams() – מעבד את המידע מהקובץ ומחזיר את הצורות כפי שהוסבר
5. find_image_location_params() – פונקציה שמעבדת את המידע ומחזירה נקודות מרכז וקצה
6. draw_the_shapes(shapes_from_file, my_canvas) – מצייר את הצורות על המסך
7. check_border_overflow(point_coordinate_key, point_coordinate_value, msg) – בודקת האם נקודה יצאה מגבולות המסך המוגדרים, ובמידה וכן יוצאת שגיאה
8. scale(my_canvas, mult_by, nkudat_ihus=False) – הגדלה של ציור על ידי הכפלה בערך שנבחר בסליידר
9. translation(my_canvas, Tx=0, Ty=0, nkudat_ihus=False, complex_transformation=False) – הזזה של הציור במרחב לפי הערך שנבחר בסליידר
10. rotation(my_canvas, angle, nkudat_ihus=False) – ביצוע סיבוב לפי המעלות שנבחרו בסליידר

11. **rotate_nekudat_ihus(x_ihus, y_ihus, angle, my_canvas)** – וולפי הערך ביצוע סיבוב של הציור על ידי הזה לראשית הציורים והחזרתו למיקום המקומי ובנוסף לפי ערך הסליידר.
12. **scale_nekudat_ihus(x_ihus, y_ihus, mult_by, my_canvas)** – ביצוע הגדלת הציור על ידי ההזזה לראשית, הגדלה לפי ערך הסליידר והחזרה למיקום המקורי.
13. **mirror(mirror_type, my_canvas)** – ביצוע אפקט מראה על הציור לפי ציר הנבחר
14. **point_in_image_border(point_x, point_y)** – בדיקה האם פיקסל בגבולות התמונה
15. **translate_with_mouse(my_canvas, my_window)** – שינוי מיקום הציור על ידי לחיצה על הכפתור וגרירה עם העכבר
16. **drag_with_mouse(event, my_canvas)** – תת פונקציה שבודקת האם הלחיצה הראשונה הייתה בתוך גבולות התמונה, ובמידה וכן היא מעדכנת את ערכי הקורדינות בהתאם להזזה
17. **mouseButtonReleased(event)** – פונקציה עבור שיחרור הלחצן של העכבר
18. **Shearing(shearing_type, my_canvas, my_window, nkudat_ihus=False)** – פונקציה שמבצעת גזירה בהתאם הגרירה-למרחק הגרירה לאחר לחיצה על הכפתור המתאים במידה והפעולה לא גורמת לתמונה לצאת מהמסך.
19. **reset_canvas(canvas)** – ניקוי המסך וציור מחדש של הציור מהקובץ המבוקש
20. **print_slider_value(slider)** – הדפסת ערך הסליידר המבוקש
21. **get_slider_values(slider)** – החזרת ערך הסליידר המבוקש
22. **get_mouse_pos_on_click(event)** – קבלת קורדינטות העכבר
23. **get_angle(num)** – הפיכת ערך הסליידר לזווית שניתן לבצע בעזרתה סיבוב
24. **set_all_sliders(all_sliders)** – החזרת ערכי הסליידר לערך האמצעי
25. **change_file_name(my_canvas, text_box)** – שינוי קובץ הפרמטרים ועדכון המסך בהתאם
26. **fit_car_to_screen(canvas)** – שינוי של ערכי הצורות המרכיבות את הציור כך שיתאים לגודל המסך
27. **__main__** – אתחול המסך, הציור המבוקש ויצירת ממשק המשתמש כאשר אנו מחברים בין הכפתורים לפונקציות המתאימות

מבנה לוגי של התוכנית:



שגיאות:

1. יציאה מגבולות המסך (Out of boarder) – בהתאם לפעולות.
2. קובץ לא נמצא (invalid file name) – במידה והמשתמש ייתן שם של קובץ שלא נמצא
3. אין אפשרות להתאים את הציור למסך (can't fit image to the screen) – השגיאה תקרה במידה והערכים בקובץ היו מחוץ לגבולות המסך.
4. תו לא תקין (כמו אות במקום מספר) – תקרה בעת עיבוד הקובץ
5. הקטנה מוגזמת (can't scale- too small) – במידה והמשתמש ירצה להקטין בצורה שלא תהיה אפשרות להגדיל אחר כך.

תודה רבה קבוצה G !