

יש לנו מודל - קובץ .scn, שמורכב מ:

- מספר הנקודות  $n$
- רשימה של  $n$  נקודות, כאשר כל נקודה מורכבת מ  $x, y, z$
- מספר הקשתות  $m$
- רשימת של  $m$  קשתות, כאשר כל קשת מורכבת ממספרים של שני קודקודים (מספור מתחיל מ0)

דוגמה:

```
18
-40 -100
40 -100
100 -40
100 40
40 100
-40 100
-100 40
-100 -40
-60 -40
-30 -40
30 -40
60 -40
-10 0
10 0
-50 40
-20 60
20 60
50 40
14
0 1
1 2
2 3
3 4
4 5
5 6
6 7
7 0
8 9
10 11
12 13
14 15
15 16
16 17
```

יש לנו גם קובץ .viw. שמגדיר את הviewport:

```
Origin -140 -100
Direction 30
```

Size 350 350  
Resolution 250 250

אנחנו רוצים לפתוח חלון עם margin של 20 בכל כיוון - כלומר הגודל של החלון הוא  $(v.width + 40, v.height + 40)$ . אנחנו נחלק את החלון ל-9 ריבועים, ולפי נקודת הלחיצה נעשה טרנספורמציה:

- אם לוחצים באמצע - עושים הזזה
- אם לוחצים בצדדים - עושים שינוי גודל
- אם לוחצים בפינות - עושים סיבוב

R	S	R
S	T	S
R	S	R

כאשר לוחצים, שומרים את נקודת הלחיצה  $S$ , מאזינים לאירוע drag כדי למצוא את נקודת היעד  $D$ , ומחשבים את הטרנספורמציה לפי  $D - S$ . גם את  $S$  וגם את  $D$  נשמור בתור וקטורים מהמרכז  $C$  - כלומר  $v_S = S - C, v_D = D - C$ . נחשב לפיהם את הטרנספורמציות

• במקרה של הזזה -  $D - S = T$ . כלומר  $\Delta x = D_x - S_x, \Delta y = D_y - S_y$ . לא מזיזים על ציר  $Z$  - כלומר  $\Delta z = 0$ .

• במקרה של שינוי גודל, המקדם יהיה  $\frac{|v_D|}{|v_S|}$ .

• במקרה של סיבוב, נרצה לסובב לפי הזווית  $\angle SCD = \theta$ . אם הזווית של  $S$  בהצגה פולרית<sup>1</sup> היא  $\alpha$  ושל  $D$  היא  $\beta$ , אז אנחנו רוצים את הזווית  $\theta = \beta - \alpha$ .

ציר הסיבוב יבחר לפי המקלדת, כאשר נלחץ על  $x$ , על  $y$  או על  $z$ . ברירת המחדל תהיה  $z$ .

- נשים  $\heartsuit$ : כל הטרנספורמציות יהיו לפי מערכת הצירים של הצופה.
- גם שינוי גודל וגם סיבוב יהיו לפי מרכז החלון.

<sup>1</sup>כאשר ציר ה- $X$  פונה ימינה וציר ה- $Y$  פונה שמאלה

## מבנה התרגיל

- מחלקת Scene שמכילה את העולם. נחזיק:
- View שעושה את הטרנספורמציות ממערכת הצירים של העולם למערכת הצירים של הצופה (יהיה  $VM$  - Viewing Matrix)
- תהיה לנו מטריצת  $TT$  - Total Transformation שמכילה את כל הטרנספורמציות עד עכשיו. נאתחל אותה למטריצת היחידה  $TT \leftarrow I$ .
- תהיה לנו מטריצת  $CT$  - Current Transformation שמכילה את הטרנספורמציה שהמשתמש מזין כרגע, לפני שעזב את כפתור העכבר. ברגע שמקבלים אירוע release מכפילים את  $CT$  ב  $TT$  ומאפסים את  $CT$  ל  $I$ .
- יש גם מטריצת  $P$  - Projection בשביל להמיר מתלת מימד לדו מימד. אבל עוד לא למדנו את זה

כאשר נרצה לצייר, נבנה את ה  $FT = CT \cdot TT \cdot VM$  - Final Transformation ונפעיל אותה על כל קודקוד כדי לקבל את ה  $FVL$  - Final Vector List. נעבור קשת קשת, נמצא את הקודקודים שלה לפי ה  $FVL$ , ונצייר:

```
FF = CT.mult(TT).mult(VM);  
FVL = Scene.apply(FT);  
for (i = 0; i < EL.length; i++) {  
    drawLine(v0, v1);  
}
```

## טעינה של קובץ

כאשר לוחצים על L צריך להיפתח dialog של טעינת קובץ, שמאפשר לטעון סצינה חדשה.