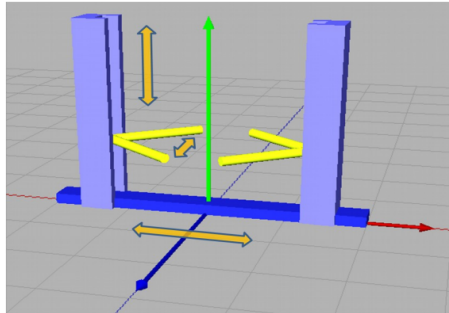


Grafo de escena completo con código y ejecución incluido

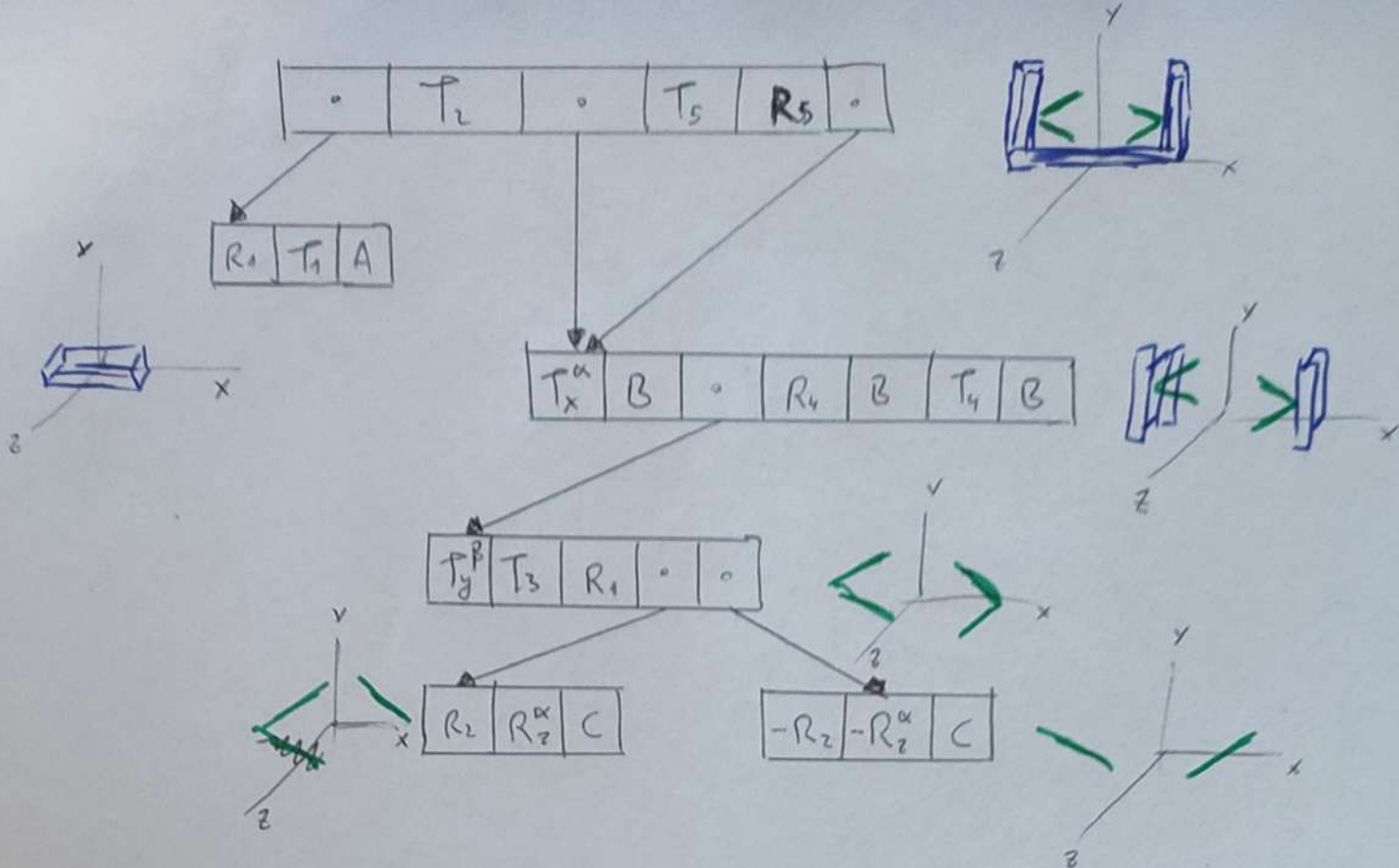
4. (4 puntos) Obtenga el grafo de escena, incluyendo las transformaciones, de modo que partiendo de los objetos básicos cubo unidad y de cilindro unidad centrados en el origen, se pueda obtener un modelo de un elevador para automóviles (ver imágenes de abajo). Haga dibujos del posicionamiento y dimensiones de las piezas como apoyo.

Hay tres grados de libertad: movimiento horizontal de las barras verticales (azules), movimiento vertical y cierre y apertura de los soportes (cilindros amarillos).



<https://drive.google.com/file/d/1EV8CjhLLeeExJod-EV7jFeERW-d1u3a0/view?usp=sharing>

Al final del doc tenéis todo.



$R_1: -90^\circ$ Eje Z

$T_1: -1, -6.5, 0$ Nota: -1 sube la plataforma en el eje Y
-6.5 es la mitad de la base de B

$T_2: -7.5, 0, 0$ Nota: -7.5 \rightarrow 6.5 de media base + 1 del tamaño de columna

T_x^x : Mueve Eje X. Rango de 0 a 1.5

T_y^y : Mueve Eje Y. Rango de -3.5 a 5 (-3.5 \leq atraviesa la base)

$T_3: 1, 5, 0.75$ Nota: 0.75 centra en la columna $\square \rightarrow \square$
5 centra altura de columna de 10 $\square \rightarrow \square$
1 centra respecto al tamaño de columna $\square \rightarrow \square$

$R_2: -20, 1, 0$ Nota: Inclinación del soporte

R_z^x : Mueve eje Z. Rango: de 0 a 15

$R_y: 90, 0, 1, 0$ Nota: Giro para colocar columna $\square \rightarrow \square$

$T_4: -3, 0, 0$ Nota: $2(\frac{\text{ancho}}{\text{base}} + 1(\text{longitud columna}))$. Desplaza a la columna

$T_5: 15, 0, 2$ Nota: $13(\text{longitud de base}) + 2(\frac{\text{ancho}}{2 \text{ por simetría}} \text{ de columna } (2 \text{ y } 2)) = 15$

$R_5: 180, 0, 1, 0$ Nota: Espejo de las columna y soporte de la izquierda

"0" significan estructuras:
`glPushMatrix();`
`glPopMatrix();`

Donde A:
`glScalef(1, 13, 2)`
`cubo.draw();`
Donde B:
`glScalef(1, 10, 2)`
`cubo.draw();`

~~`glScalef(5, 5, 5)`~~
~~`cilindro.draw();`~~

Donde C:
`glScalef(0.5, 5, 0.5)`
`cilindro.draw();`

```

1: //Variables de animacion
2: int colsigno = 1, rsopsigno = 1, ysopsigno = 1; //Para cambiar la direccion de movimiento
3: float xcolumna = 0, rsoporte = 20, ysoporte = 0;
4:
5: void columnasysoportes() {
6:     glPushMatrix();
7:     glTranslatef(xcolumna,0,0); //Movimiento columna Talfa
8:     ortoedro(0.5,0,1,0.5,10,1,1,2); //B (Columna frontal) Equivale a un cubo unidad escalado 1x10x2
9:
10:    glPushMatrix();
11:    materialOrtoedro(light_yellow);
12:    glTranslatef(0,ysoporte,0); //Movimiento soportes en y (Tbeta)
13:    glTranslatef(1,5,0.75); //T3
14:    glRotatef(-90,0,0,1); //R1
15:    glPushMatrix();
16:    glRotatef(-20,1,0,0); //R2
17:    glRotatef(rsoporte,1,0,0); //Rotacion soporte
18:    cilindro(0.25,0,0.25,0.25,5,0.25,0.5,0.5); //C(soporte izquierdo) Equivale a un cilindro unidad
escalado 0.5x5x0.5
19:    glPopMatrix();
20:    glPushMatrix();
21:    glRotatef(20,1,0,0); //R3
22:    glRotatef(-rsoporte,1,0,0); //Rotacion soporte
23:    cilindro(0.25,0,0.25,0.25,5,0.25,0.5,0.5); //C(soporte derecho) Equivale a un cilindro unidad es
calado 0.5x5x0.5
24:    glPopMatrix();
25:    materialOrtoedro(light_blue);
26:    glPopMatrix();
27:
28:    glRotatef(90,0,1,0); //R4
29:    ortoedro(0.5,0,1,0.5,10,1,1,2); //B(columna lateral 1) Equivale a un cubo unidad escalado 1x10x2
30:    glTranslatef(-3,0,0); //T4
31:    ortoedro(0.5,0,1,0.5,10,1,1,2); //B(columna lateral 2) Equivale a un cubo unidad escalado 1x10x2
32:    glPopMatrix();
33: }
34:
35: Dibuja() {
36:     Inicializar luz, escena, etc.....
37:
38:     materialOrtoedro(light_blue); //Establece el material de la columna
39:     glPushMatrix();
40:     glRotatef(-90,0,0,1); //R1
41:     glTranslatef(-1,-6.5,0); //T1
42:     ortoedro(0.5,0,1,0.5,13,1,1,2); //A(Base) Equivale a un cubo unidad escalado 1x13x2
43:     glPopMatrix();
44:
45:     glTranslatef(-7.5,0,0); //T2
46:     columnasysoportes(); //Funcion que dibuja 3 columnas en forma de U y sus soportes
47:
48:     glTranslatef(15,0,2); //T5
49:     glRotatef(180,0,1,0); //R5
50:     columnasysoportes(); //Funcion que dibuja 3 columnas en forma de U y sus soportes
51: }
52:
53: //Rangos de animacion
54: void idle(int v) {
55:     //elevador
56:     xcolumna += 0.01*colsigno;
57:     if(xcolumna > 1.5)
58:         colsigno = -1;
59:     else if(xcolumna < 0)
60:         colsigno = 1;
61:
62:     ysoporte += 0.1*ysopsigno;
63:     if(ysoporte > 5)
64:         ysopsigno = -1;
65:     else if(ysoporte < -3.5)
66:         ysopsigno = 1;
67:
68:     rsoporte += 0.5*rsopsigno;
69:     if(rsoporte > 15)
70:         rsopsigno = -1;
71:     else if(rsoporte < 0)
72:         rsopsigno = 1;
73: }
74:
75:

```