UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

REPORTE DE PRÁCTICA Nº 09

NOMBRE COMPLETO: Barragán Pilar Diana

Nº de Cuenta: 318147981

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 19 de octubre de 2024

CALIFICACIÓN: _____

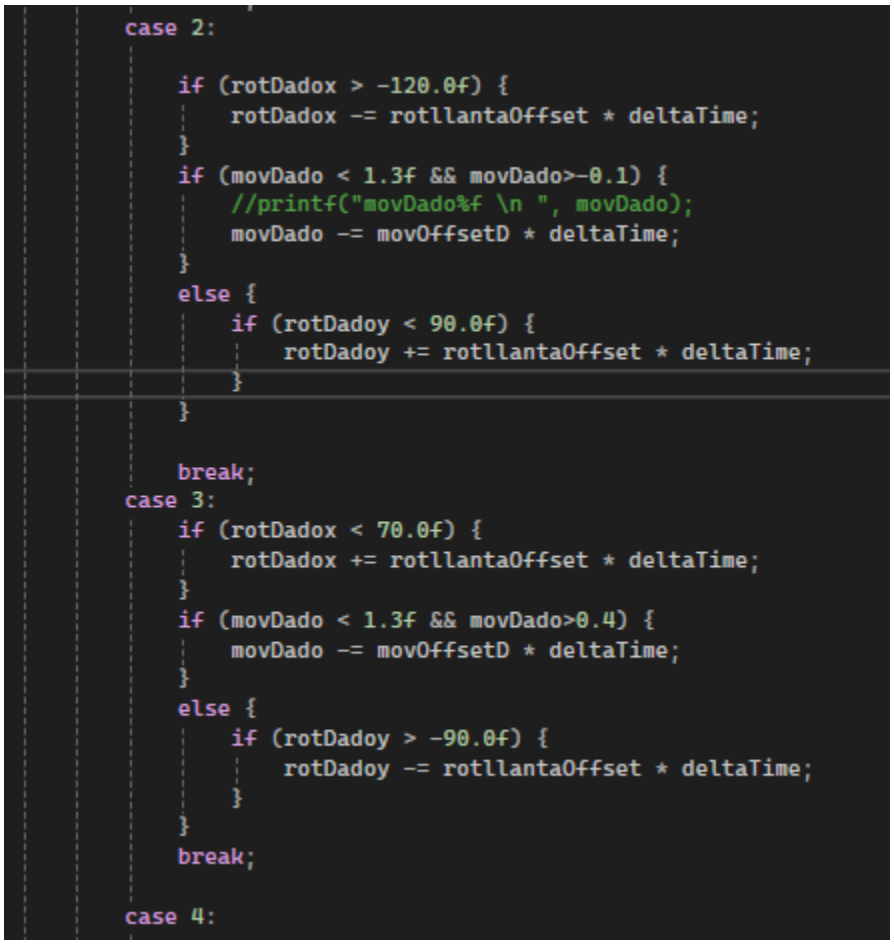
Práctica 9: Animación Básica

1. **Su dado de 10 caras cae sobre el piso, gira y cae en un número "random", se repite la tirada al presionar una tecla.**

En este ejercicio lo primero que hice fue volver a colocar el código de mi dado para a partir de este instanciarlo en el código y aplicarle las transformaciones pertinentes para que cayera un número random. Lo primero fue poner lo el traslate con una variable en y para la animación cuando cae, además coloque rotate con variables en x y y para que se vea una cierta cara dependiendo de la tirada al azar. Posteriormente generé un número aleatorio con rand() que me de un número del 1 al 10 pues estas son las caras del dado, luego dentro del main con ayuda de mainWindow.getpres() si esta es igual a true es porque se esta presionando una tecla en este caso la tecla T y entonces comienza la animación con el primer if para que caiga el dado al piso, después dependiendo de la cara aleatoria entra en distintos case en donde se crea el movimiento de ciertos ángulos hasta que la cara con el número quede arriba.

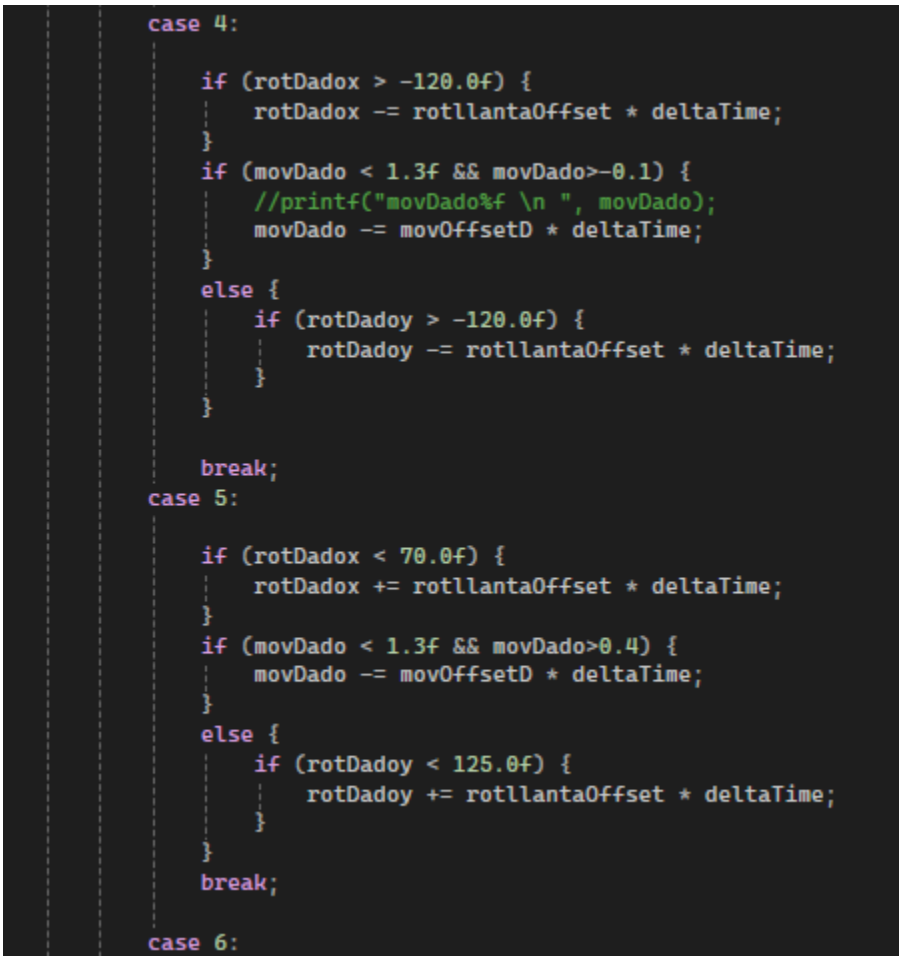
```
//Tirada del Dado
if (mainWindow.getpres() == true) {
    if (caer) {

        if (movDado <= 6.5 && movDado > 1.3f) {
            movDado -= movOffsetD * deltaTime;
        }
        else {
            switch (tiradaDado) {
            case 1:
                if (rotDadox < 70.0f) {
                    rotDadox += rotllantaOffset * deltaTime;
                }
                if (movDado < 1.3f && movDado>0.4) {
                    movDado -= movOffsetD * deltaTime;
                }
                else {
                    if (rotDadoy > -20.0f) {
                        rotDadoy -= rotllantaOffset * deltaTime;
                    }
                }

                break;
            case 2:
```

Figura 1. Case 1 tirada del dado.



```
case 2:

    if (rotDadox > -120.0f) {
        rotDadox -= rotllantaOffset * deltaTime;
    }
    if (movDado < 1.3f && movDado>-0.1) {
        //printf("movDado%f \n ", movDado);
        movDado -= movOffsetD * deltaTime;
    }
    else {
        if (rotDadoy < 90.0f) {
            rotDadoy += rotllantaOffset * deltaTime;
        }
    }

    break;
case 3:
    if (rotDadox < 70.0f) {
        rotDadox += rotllantaOffset * deltaTime;
    }
    if (movDado < 1.3f && movDado>0.4) {
        movDado -= movOffsetD * deltaTime;
    }
    else {
        if (rotDadoy > -90.0f) {
            rotDadoy -= rotllantaOffset * deltaTime;
        }
    }
    break;

case 4:
```

Figura 2. Case 2 y 3 tirada del dado.



```
        case 4:

            if (rotDadox > -120.0f) {
                rotDadox -= rotllantaOffset * deltaTime;
            }
            if (movDado < 1.3f && movDado>-0.1) {
                //printf("movDado%f \n ", movDado);
                movDado -= movOffsetD * deltaTime;
            }
            else {
                if (rotDadoy > -120.0f) {
                    rotDadoy -= rotllantaOffset * deltaTime;
                }
            }

            break;
        case 5:

            if (rotDadox < 70.0f) {
                rotDadox += rotllantaOffset * deltaTime;
            }
            if (movDado < 1.3f && movDado>0.4) {
                movDado -= movOffsetD * deltaTime;
            }
            else {
                if (rotDadoy < 125.0f) {
                    rotDadoy += rotllantaOffset * deltaTime;
                }
            }
            break;

        case 6:
```

Figura 3. Case 4 y 5 tirada del dado.

```
case 6:

    if (rotDadox > -120.0f) {
        rotDadox -= rotllantaOffset * deltaTime;
    }
    if (movDado < 1.3f && movDado>-0.1) {
        //printf("movDado%f \n ", movDado);
        movDado -= movOffsetD * deltaTime;
    }
    else {
        if (rotDadoy < 165.0f) {
            rotDadoy += rotllantaOffset * deltaTime;
        }
    }
    break;

case 7:

    if (rotDadox < 70.0f) {
        rotDadox += rotllantaOffset * deltaTime;
    }
    if (movDado < 1.3f && movDado>0.4) {
        movDado -= movOffsetD * deltaTime;
    }
    else {
        if (rotDadoy < 55.0f) {
            rotDadoy += rotllantaOffset * deltaTime;
        }
    }

    break;

case 8:
```

Figura 4. Case 6 y 7 tirada del dado.

```
        case 8:

            if (rotDadox > -120.0f) {
                rotDadox -= rotllantaOffset * deltaTime;
            }
            if (movDado < 1.3f && movDado>-0.1) {
                //printf("movDado%f \n ", movDado);
                movDado -= movOffsetD * deltaTime;
            }
            else {
                if (rotDadoy < 17.0f) {
                    rotDadoy += rotllantaOffset * deltaTime;
                }
            }

            break;

        case 9:
            if (rotDadox < 70.0f) {
                rotDadox += rotllantaOffset * deltaTime;
            }
            if (movDado < 1.3f && movDado>0.4) {
                movDado -= movOffsetD * deltaTime;
            }
            else {
                if (rotDadoy > -160.0f) {
                    rotDadoy -= rotllantaOffset * deltaTime;
                }
            }

            break;
```

Figura 5. Case 8 y 9 tirada del dado.

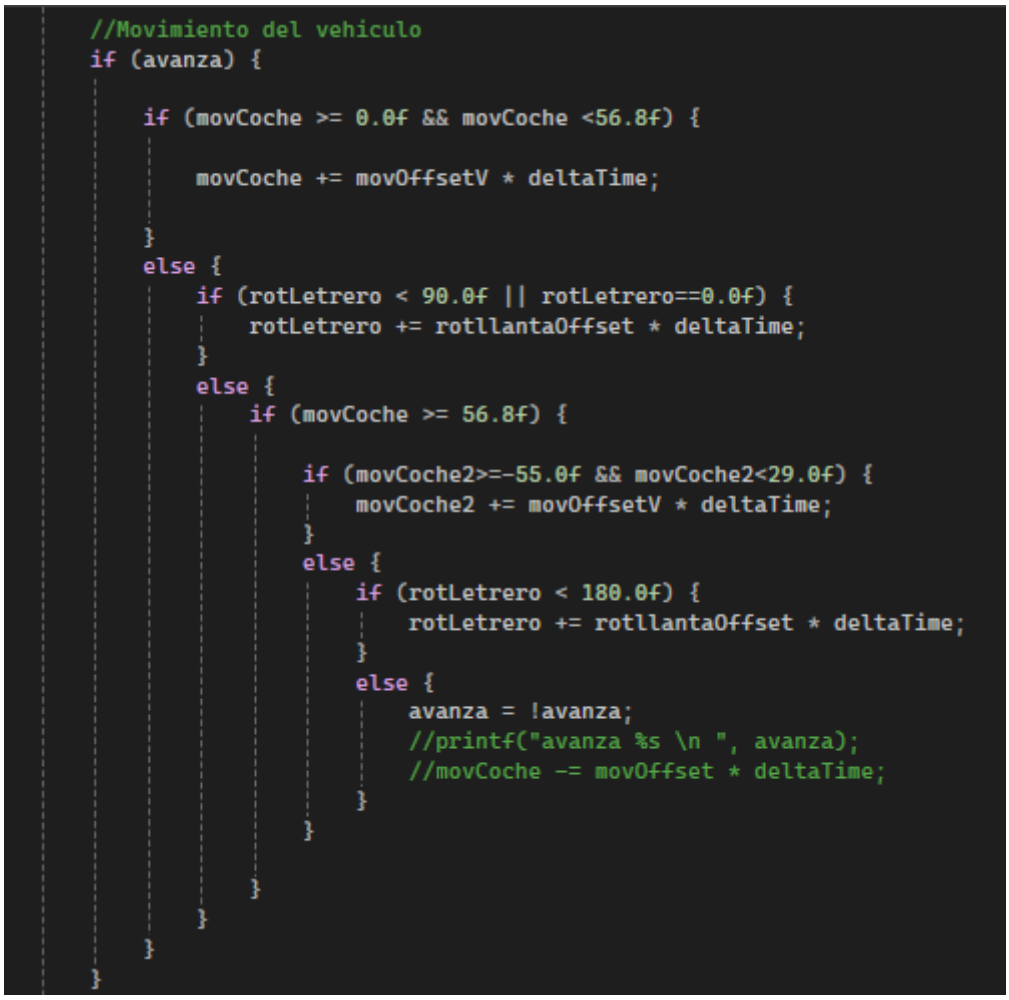
```
                    break;

                case 10:
                    if (rotDadox > -120.0f) {
                        rotDadox -= rotllantaOffset * deltaTime;
                    }
                    if (movDado < 1.3f && movDado>-0.1) {
                        //printf("movDado%f \n ", movDado);
                        movDado -= movOffsetD * deltaTime;
                    }
                    else {
                        if (rotDadoy > -50.0f) {
                            rotDadoy -= rotllantaOffset * deltaTime;
                        }
                    }
                    break;

                default:

                    break;
                }
            }
        }
    }
```

Figura 6. Case 10 tirada del dado.

2. **Por integrante del equipo elegirán un tipo de vehículo: terrestre o aéreo. Cada integrante del equipo creará un recorrido en el cual el vehículo se moverá alrededor de su tablero de Monopoly. Cada vehículo iniciará su recorrido a partir de una esquina diferente. (el vehículo terrestre no puede ser un carro o vehículo similar motorizado de 4 ruedas, se debe de tener movimiento de llantas o de hélices en sus vehículos.)**

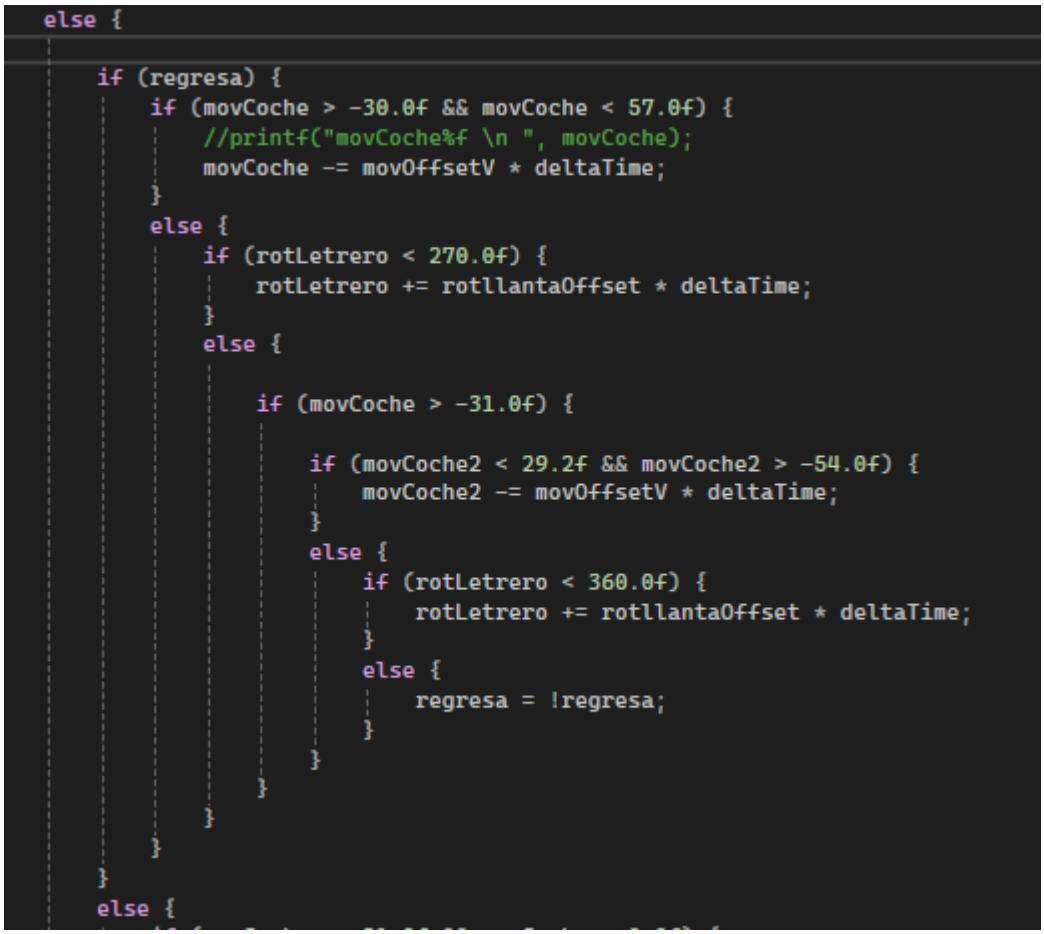
Para este ejercicio mi modelo es una especie de aerodeslizador como si fuera una nave o un avión, como siempre importe el modelo, declare su ubicación y a la hora de instanciarlo con un traslate utilice dos variables movCoche y movCoche2 para el recorrido al redor del tablero cambiando de posición en x y en z, así como rotando en y en cada esquina para girar. Lo mismo hice con las hélices simplemente que cambié para que rotarán en el eje y.

Luego para que se moviera en un if -else coloque la variable booleana avanzar, para después comenzar con el recorrido en el eje z, hasta llegar a la coordenada de la esquina donde dentro con ayuda de otro if es que rote el modelo para que diera la vuelta y sigue avanzando, así hasta llegar a la segunda esquina en donde la variable avanza se pone el false por lo que entraremos en el else donde asigne otra variable booleana para que al entrar al if se realiza el resto del recorrido con las variables ya mencionadas hasta que volvemos al punto de inicio en este caso, la variable regresa está en false por lo que inicialice las variables para que el recorrido fuera cíclico.



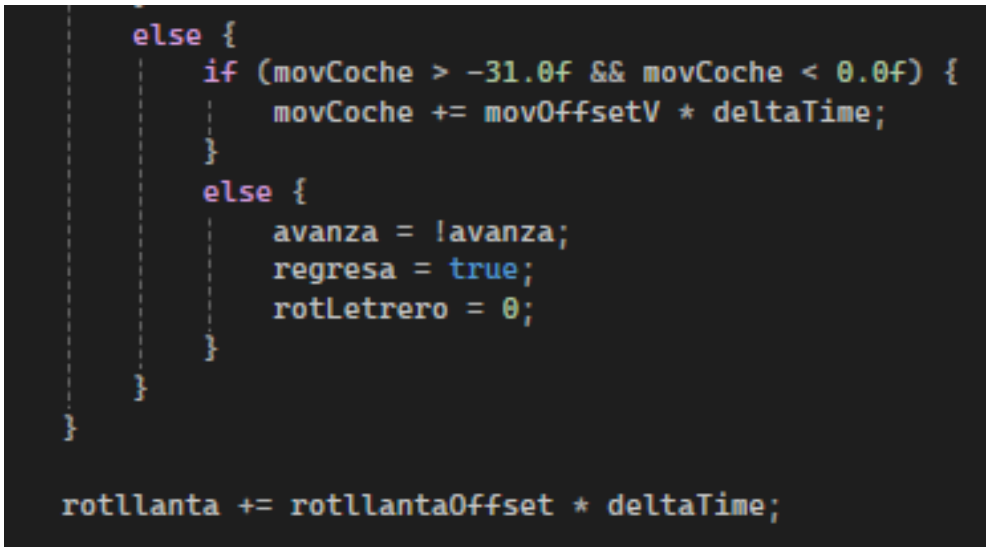
```
//Movimiento del vehiculo
if (avanza) {

    if (movCoche >= 0.0f && movCoche <56.8f) {

        movCoche += movOffsetV * deltaTime;

    }
    else {
        if (rotLetrero < 90.0f || rotLetrero==0.0f) {
            rotLetrero += rotllantaOffset * deltaTime;
        }
        else {
            if (movCoche >= 56.8f) {

                if (movCoche2>=-55.0f && movCoche2<29.0f) {
                    movCoche2 += movOffsetV * deltaTime;
                }
                else {
                    if (rotLetrero < 180.0f) {
                        rotLetrero += rotllantaOffset * deltaTime;
                    }
                    else {
                        avanza = !avanza;
                        //printf("avanza %s \n ", avanza);
                        //movCoche -= movOffset * deltaTime;
                    }
                }

            }
        }
    }
}
```

Figura 7. Código movimiento del vehículo.



```
else {

    if (regresa) {
        if (movCoche > -30.0f && movCoche < 57.0f) {
            //printf("movCoche%f \n ", movCoche);
            movCoche -= movOffsetV * deltaTime;
        }
        else {
            if (rotLetrero < 270.0f) {
                rotLetrero += rotllantaOffset * deltaTime;
            }
            else {

                if (movCoche > -31.0f) {

                    if (movCoche2 < 29.2f && movCoche2 > -54.0f) {
                        movCoche2 -= movOffsetV * deltaTime;
                    }
                    else {
                        if (rotLetrero < 360.0f) {
                            rotLetrero += rotllantaOffset * deltaTime;
                        }
                        else {
                            regresa = !regresa;
                        }
                    }
                }
            }
        }
    }
    else {
```

Figura 8. Código movimiento del vehículo.



```
        else {
            if (movCoche > -31.0f && movCoche < 0.0f) {
                movCoche += movOffsetV * deltaTime;
            }
            else {
                avanza = !avanza;
                regresa = true;
                rotLetrero = 0;
            }
        }
    }

    rotllanta += rotllantaOffset * deltaTime;
```

Figura 8. Código movimiento del vehículo.

Al compilar el código de esta práctica no se me presento ningún inconveniente respecto a la programación de este, puesto que todo funciono con normalidad.

CONCLUSIÓN:

Con la realización de esta práctica comprendí mejor la implementación y el uso de la animación con ayuda de otras variables fuera del while y el como es que se implementan a partir de estructuras como el if – else y switch para lograr el efecto de que la figura se mueve con ayuda de rotaciones y traslaciones, además de permitirme conocer más a fondo el código y su funcionamiento.