



Instituto Politécnico Nacional Escuela Superior de Cómputo

Teoría de la computación

Practica 5: Tablero

Profesor: Juarez Martinez Genaro

4CM6

Torres Abonce Luis Miguel
Mayo 2022

Contents

1	Introducción	3
2	Marco Teórico	4
3	Desarrollo	5
3.1	Código	5
3.2	Explicación	17
3.3	Capturas de ejecución	19
4	Conclusiones	21
5	Bibliografía	22

1 Introducción

Descripción del problema:

EL problema a consiste en utilizar un tablero tipo ajedrez de 4x4 y mediante una cadena de 'b' y 'n' (ej. bnbnnbnnb) 'b' y 'n' significa entre que casillas se puede mover la pieza, si el carácter es 'b' solo se podrá mover a una casilla blanca y de igual manera con carácter 'n', se podrá jugar con 1 o 2 fichas si es una ficha, para ganar tendrá que llegar a la esquina opuesta a la que inicio, de igual forma con 2 piezas, si en algún momento las fichas se cruzan sus caminos tendrá que recalcular la ruta a partir de ese momento y si esto no es posible se saltara un turno y continuara normalmente; los caminos son generados antes de ver la animación y se elige una entre todas las rutas ganadoras, el programa podrá dar a elegir entre realizar toda la configuración en automático o de forma manual asignando los valores de la cadena o cuantas piezas jugaran. En archivos se deberán guardar todas las rutas generadas de cada pieza así como las rutas ganadoras de cada pieza.

2 Marco Teórico

Para la resolución de este problema se utilizara la recursividad.

Recursividad. Proceso mediante el que una función se llama a sí misma de forma repetida, hasta que se satisface alguna determinada condición

3 Desarrollo

3.1 Código

```
from tkinter import CENTER
import turtle
import sys, os
import random
import time

#Definimos la ruta de las pieza
def generarRuta(cadena,estado,camino,ficha):
    if(estado==0):#si el estado actual es ese entra a la condicion
        camino=camino+"00"#Se escribe el camnio que va llevando
        if len(cadena)== 1 or 0:
            if ficha ==0:#Dependiendo de la ficha escribe en el documento
                caminos0.write(camino+",")
            else:
                caminosX.write(camino+",")
            return#retornar a la ultima funcion
        else:
            cadena=cadena[1:]
            if(cadena[0]=="b"):#2 opciones mover a blancas
                generarRuta(cadena,1,camino,ficha)#mover al estado
                generarRuta(cadena,4,camino,ficha)
            else: generarRuta(cadena,5,camino,ficha)#1 opcion mover a negras
    elif(estado==1):
        camino=camino+"01"
        if len(cadena)==1:
            if ficha ==0:
                caminos0.write(camino+",")
            else:
                caminosX.write(camino+",")
            return
        else:
            cadena=cadena[1:]
            if(cadena=="b"):
                generarRuta(cadena,4,camino,ficha)
                generarRuta(cadena,6,camino,ficha)
            else:
                generarRuta(cadena,0,camino,ficha)
                generarRuta(cadena,5,camino,ficha)
                generarRuta(cadena,2,camino,ficha)
    elif(estado==2):
        camino=camino+"02"
        if len(cadena)==1:
            if ficha ==0:
                caminos0.write(camino+",")
            else:
                caminosX.write(camino+",")
            return
        else:
            cadena=cadena[1:]
            if(cadena=="b"):
                generarRuta(cadena,1,camino,ficha)
                generarRuta(cadena,6,camino,ficha)
```

```

        generarRuta(cadena,3,camino,ficha)
    else:
        generarRuta(cadena,5,camino,ficha)
        generarRuta(cadena,7,camino,ficha)
elif(estado==3):
    camino=camino+"03"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"): generarRuta(cadena,6,camino,ficha)
        else:
            generarRuta(cadena,7,camino,ficha)
            generarRuta(cadena,2,camino,ficha)
elif(estado==4):
    camino=camino+"04"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,1,camino,ficha)
            generarRuta(cadena,9,camino,ficha)
        else:
            generarRuta(cadena,0,camino,ficha)
            generarRuta(cadena,5,camino,ficha)
            generarRuta(cadena,8,camino,ficha)
elif(estado==5):
    camino=camino+"05"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,1,camino,ficha)
            generarRuta(cadena,4,camino,ficha)
            generarRuta(cadena,6,camino,ficha)
            generarRuta(cadena,9,camino,ficha)
        else:
            generarRuta(cadena,0,camino,ficha)
            generarRuta(cadena,8,camino,ficha)
            generarRuta(cadena,2,camino,ficha)
            generarRuta(cadena,10,camino,ficha)
elif(estado==6):
    camino=camino+"06"
    if len(cadena)==1:

```

```

        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,1,camino,ficha)
            generarRuta(cadena,3,camino,ficha)
            generarRuta(cadena,9,camino,ficha)
            generarRuta(cadena,11,camino,ficha)
        else:
            generarRuta(cadena,2,camino,ficha)
            generarRuta(cadena,5,camino,ficha)
            generarRuta(cadena,7,camino,ficha)
            generarRuta(cadena,10,camino,ficha)
elif(estado==7):
    camino=camino+"07"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,3,camino,ficha)
            generarRuta(cadena,6,camino,ficha)
            generarRuta(cadena,11,camino,ficha)
        else:
            generarRuta(cadena,2,camino,ficha)
            generarRuta(cadena,10,camino,ficha)

elif(estado==8):
    camino=camino+"08"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,4,camino,ficha)
            generarRuta(cadena,9,camino,ficha)
            generarRuta(cadena,12,camino,ficha)
        else:
            generarRuta(cadena,5,camino,ficha)
            generarRuta(cadena,13,camino,ficha)
elif(estado==9):
    camino=camino+"09"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:

```

```

        caminosX.write(camino+",")
    return
else:
    cadena=cadena[1:]
    if(cadena=="b"):
        generarRuta(cadena,4,camino,ficha)
        generarRuta(cadena,6,camino,ficha)
        generarRuta(cadena,12,camino,ficha)
        generarRuta(cadena,14,camino,ficha)
    else:
        generarRuta(cadena,5,camino,ficha)
        generarRuta(cadena,8,camino,ficha)
        generarRuta(cadena,13,camino,ficha)
        generarRuta(cadena,10,camino,ficha)
elif(estado==10):
    camino=camino+"10"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,6,camino,ficha)
            generarRuta(cadena,9,camino,ficha)
            generarRuta(cadena,14,camino,ficha)
            generarRuta(cadena,11,camino,ficha)
        else:
            generarRuta(cadena,5,camino,ficha)
            generarRuta(cadena,7,camino,ficha)
            generarRuta(cadena,15,camino,ficha)
            generarRuta(cadena,13,camino,ficha)
elif(estado==11):
    camino=camino+"11"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,6,camino,ficha)
            generarRuta(cadena,14,camino,ficha)
        else:
            generarRuta(cadena,7,camino,ficha)
            generarRuta(cadena,10,camino,ficha)
            generarRuta(cadena,15,camino,ficha)
elif(estado==12):
    camino=camino+"12"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        elif ficha==1:
            caminosX.write(camino+",")

```



```

        listaX.append(camino)#Si llega a este punto es el camino ganador
        ganadorX.write(camino+",")#de esa ficha
    return
else:
    cadena=cadena[1:]
    if(cadena=="b"): generarRuta(cadena,9,camino,ficha)
    else:
        generarRuta(cadena,8,camino,ficha)
        generarRuta(cadena,13,camino,ficha)
elif(estado==13):
    camino=camino+"13"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,12,camino,ficha)
            generarRuta(cadena,9,camino,ficha)
            generarRuta(cadena,14,camino,ficha)
        else:
            generarRuta(cadena,8,camino,ficha)
            generarRuta(cadena,10,camino,ficha)
elif(estado==14):
    camino=camino+"14"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,9,camino,ficha)
            generarRuta(cadena,11,camino,ficha)
        else:
            generarRuta(cadena,13,camino,ficha)
            generarRuta(cadena,10,camino,ficha)
            generarRuta(cadena,15,camino,ficha)
elif(estado==15):
    camino=camino+"15"
    if len(cadena)==1:
        if ficha ==0:
            caminos0.write(camino+",")
            lista0.append(camino)#Si llega a este punto es el camino ganador
            ganador0.write(camino+",")#de esta ficha
        else:
            caminosX.write(camino+",")
        return
    else:
        cadena=cadena[1:]
        if(cadena=="b"):
            generarRuta(cadena,14,camino,ficha)
            generarRuta(cadena,11,camino,ficha)
        else: generarRuta(cadena,10,camino,ficha)
#Recalcular ruta

```

```

def recalcularRuta(camino,estado,posicion,ficha,estado2):
    camino=camino[:posicion]
    listaX.clear()
    lista0.clear()
    if ficha==0:
        generarRuta(cadena1[int(posicion/2):],int(estados),"",0)
        random.shuffle(lista0)#elegir uno alazar
        for i in lista0:
            if i[2:4]!=estado2:#Elegir un estado diferente al que la otra ficha es igual
                camino=camino+i#Escribir el camino recalculado
                return camino
    else:
        generarRuta(cadena2[int(posicion/2):],int(estados),"",1)
        random.shuffle(listaX)
        for i in listaX:
            if i[2:4]!=estado2:
                camino=camino+i
                return camino
    return 1#se debe saltar turno

#Obtener la ruta absoluta de la imagen
def rutaImagen(nombreImagen):#recibe 1 argumento
    pathScript=sys.argv[0]
    carpeta,script=os.path.split(pathScript)
    pathFondo=os.path.join(carpeta, nombreImagen)#Obtiene la ruta absoluta
    return pathFondo#regresa la ruta

def moverX0(X0,estado):
    X0.speed(1)
    if(estado=="00"):
        X0.goto(-240,240)
        estado=0
    elif(estado=="01"):
        X0.goto(-80,240)
        estado=1
    elif(estado=="02"):
        X0.goto(80,240)
        estado=2
    elif(estado=="03"):
        X0.goto(240,240)
        estado=3
    elif(estado=="04"):
        X0.goto(-240,80)
        estado=4
    elif(estado=="05"):
        X0.goto(-80,80)
        estado=5
    elif(estado=="06"):
        X0.goto(80,80)
        estado=6
    elif(estado=="07"):
        X0.goto(240,80)
        estado=7
    elif(estado=="08"):
        X0.goto(-240,-80)
        estado=8
    elif(estado=="09"):
        X0.goto(-80,-80)
        estado=9

```

```

elif(estado=="10"):
    X0.goto(80,-80)
    estado=10
elif(estado=="11"):
    X0.goto(240,-80)
    estado=11
elif(estado=="12"):
    X0.goto(-240,-240)
    estado=12
elif(estado=="13"):
    X0.goto(-80,-240)
    estado=13
elif(estado=="14"):
    X0.goto(80,-240)
    estado=14
elif(estado=="15"):
    X0.goto(240,-240)
    estado=15
    return estado
#Menu del programa
print("Menu del programa:")
print("1. Automatico todo")
print("2. Manual")
opcion= int(input("Elige una opcion: "))
#Variables
caminos0=open("Caminos0.txt","w")
caminosX=open("caminosX.txt","w")
ganador0=open("ganador0.txt","w")
ganadorX=open("ganadorX.txt","w")
lista0=[]
listaX=[]
cadena1=""
cadena2=""
i=0
if (opcion==1):#damos valor aleatorio a la cadena
    numPiezas=2
    #numPiezas=random.randint(1,2)#con cuantas piezas jugara
    print("Num piezas: "+str(numPiezas))
    while (1):
        rango=random.randint(1,10)
        if numPiezas==2:
            while i != rango:
                cadena1=cadena1+random.choice(["b","n"])#generar cadena aleatoria
                cadena2=cadena2+random.choice(["b","n"])#generar cadena aleatoria
                i+=1
            try:#inteta generar rutas
                generarRuta(cadena1,0,"",0)
                generarRuta(cadena2,3,"",1)
                a=lista0[0]
                b=listaX[0]
                break
            except:#si da un error vaciamos las listas
                listaX=[]
                lista0=[]
                cadena1=''#borramos contenido de los strings
                cadena2=''
                i=0
        else:

```

```

        while i != rango:
            cadena1=cadena1+random.choice(["b","n"])#generar cadena aleatoria
            i+=1
        try:
            generarRuta(cadena1,0,"",0)
            a=lista0[0]
            break
        except:
            lista0=[]
            cadena1=''
            i=0
    else:
        numPiezas=int(input("Piezas a jugar (1 o 2): "))#num de piezas a jugar
        aleat=input("generar aleatoriamente las cadenas s/n:")
        if aleat=="s":
            while(1):
                rango=random.randint(1,10)#establecer rango
                while i != rango:
                    cadena1=cadena1+random.choice(["b","n"])
                    if numPiezas==2: cadena2=cadena2+random.choice(["b","n"])
                    i+=1
                try:
                    generarRuta(cadena1,0,"",0)
                    a=lista0[0]
                    if numPiezas==2:
                        generarRuta(cadena2,3,"",1)
                        b=listaX[0]
                    break
                except:
                    lista0=[]
                    cadena1=''
                    if numPiezas==2:
                        listaX=[]
                        cadena2=''
                    i=0
            else:
                print("(solo caracteres: 'b' y 'n' sin espacio)")
                cadena1=input("Primera cadena: ")
                generarRuta(cadena1,0,"",0)
                if os.stat('ganador0.txt').st_size == 0:
                    print("no se puede ganar listas de ganadores vacias")
                    exit()
                if numPiezas==2:
                    cadena2=input("Segunda Cadena: ")
                    generarRuta(cadena2,3,"",1)
                    if os.stat('ganadorX.txt').st_size == 0:
                        print("no se puede ganar listas de ganadores vacias")
#Obtenemos las rutas de las imagenes
ruta0=rutaImagen("0.gif")
rutaX=rutaImagen("X.gif")
tablero=rutaImagen("tablero.gif")
ganadorImg=rutaImagen("ganador.gif")
#Creamos la ventana
ventana = turtle.Screen()#Crear la ventana
ventana.setup(1350,950)#Dar dimensiones a ventana
ventana.bgpic(tablero)#Poner imagen de fondo
ventana.addshape(ruta0)#Insertar imagenes
ventana.addshape(rutaX)

```

```

ventana.addshape(ganadorImg)
#circulo en lugar de inicio
O=turtle.Turtle()
O.penup()
O.speed(0)
O.goto(-240,240)
O.speed(1)
estadoO=0 #iniciar estado de O
O.shape(rutaO)#Muestra la imagen
if numPiezas==2:#cruz en lugar de inicio
    X=turtle.Turtle()
    X.penup()
    X.speed(0)
    X.goto(240,240)
    X.speed(1)
    estadoX=3#iniciar estado de X
    X.shape(rutaX)
#inicializamos variables
tortucad=turtle#tortuga para actualizar cadenas
tortucad.speed(0)
if numPiezas==2:#Asignar turno
    if random.randint(0,1)==1: turnoActivo="O"#asignar primer turno
    else: turnoActivo="X"#asignar primer turno
    tortucad.penup()
    tortucad.goto(-650,220)
    tortucad.write("Empieza:"+turnoActivo,font=("courier",20, "bold"))
#crear cuadro del costado
tortucad.hideturtle()
tortucad.penup()
tortucad.goto(-650,200)
tortucad.pendown()
tortucad.goto(-350,200)
tortucad.goto(-350,115)
tortucad.goto(-650,115)
tortucad.goto(-650,200)
tortucad.penup()
#crear cuadro de abajo
g=-650#no se cambia
h=100
k=-350#no se cambia
n=55
l=-645#no se cambia
y1=70
y2=50
def reiniciarCuadro(cad1,cad2,g,h,k,n,l,y1,y2):
    tortucad=turtle
    tortucad.color("cyan")
    tortucad.speed(0)
    tortucad.hideturtle()
    tortucad.penup()
    tortucad.goto(g,h)
    tortucad.begin_fill()
    tortucad.pendown()
    tortucad.goto(k,h)
    tortucad.goto(k,n)
    tortucad.goto(g,n)
    tortucad.goto(g,h)
    tortucad.end_fill()

```

```

tortucad.color("black")
tortucad.penup()
tortucad.goto(1,y1)
tortucad.write("0:"+cad1,font=("courier",15, "bold"))
tortucad.goto(1,y2)
tortucad.write("X:"+cad2,font=("courier",15, "bold"))
tortucad.penup()
tortucad.goto(-645,165)
tortucad.write("Cadenas de las fichas",font=("courier",15,"bold"))
tortucad.goto(-645,140)
tortucad.write("0:"+cadena1,font=("courier",15, "bold"))
if numPiezas==2:
    tortucad.goto(-645,115)
    tortucad.write("X:"+cadena2,font=("courier",15, "bold"))
#Elegir ganadores:
gan0=random.choice(lista0)
if numPiezas==2: ganX=random.choice(listaX)
#Mover piezas:
j=0
i=0
ganador=""
tortustado=turtle
if numPiezas==2:
    reiniciarCuadro(gan0,ganX,g,h,k,n,l,y1,y2)
    x=0
    while(1):
        try:
            gan0[i]#comprobar que no esta vacio
            ganX[j]
        except:
            break#Termino y se rompe el ciclo
    if turnoActivo=="0":
        x=0
        if gan0[i:i+2]==ganX[j:j+2]:#comprobar si se encuentran en el mismo lugar
            a=recalcularRuta(ganX,ganX[j-2:j],j-2,1,ganX[j:j+2])#recalcular ruta
            if a == 1:
                print("saltar turno")
                x=1
            else:
                ganX=a
                h=h-45
                n=n-45
                y1=y1-45
                y2=y2-45
                reiniciarCuadro(gan0,ganX,g,h,k,n,l,y1,y2)
                x=0
        if moverX0(0,gan0[i:i+2])==15:
            ganador=0
            break
    time.sleep(.5)
    if x==0:
        if gan0[i+2:i+4]==ganX[j:j+2]:
            a=recalcularRuta(ganX,ganX[j-2:j],j-2,1,ganX[j:j+2])
            if a == 1:
                print("saltar turno")
                x=1
            else:
                ganX=a

```

```

        h=h-45
        n=n-45
        y1=y1-45
        y2=y2-45
        reiniciarCuadro(gan0, ganX, g, h, k, n, l, y1, y2)
        x=0
    if x!=1:
        if moverX0(X, ganX[i:i+2])==12:
            ganador=1
            break
        time.sleep(.5)
        j+=2
    i+=2
else:
    x=0
    if gan0[i:i+2]==ganX[j:j+2]:
        a=recalcularRuta(gan0, gan0[i-2:i], i-2, 0, gan0[i:i+2])
        if a == 1:
            print("saltar turno")
            x=1
        else:
            gan0=a
            h=h-45
            n=n-45
            y1=y1-45
            y2=y2-45
            reiniciarCuadro(gan0, ganX, g, h, k, n, l, y1, y2)
            x=0
    if moverX0(X, ganX[j:j+2])==12:
        ganador=1
        break
    time.sleep(.5)
    if x==0:
        if gan0[i:i+2]==ganX[j+2:j+4]:
            a=recalcularRuta(gan0, gan0[i-2:j], i-2, 0, gan0[i:i+2])
            if a == 1:
                print("saltar turno")
                x=1
            else:
                gan0=a
                h=h-45
                n=n-45
                y1=y1-45
                y2=y2-45
                reiniciarCuadro(gan0, ganX, g, h, k, n, l, y1, y2)
                x=0
        if x!=1:
            if moverX0(0, gan0[i:i+2])==15:
                ganador=0
                break
            i+=2
            time.sleep(.5)
        j+=2
else:
    tortustado.goto(-645,70)
    tortustado.write("0:"+gan0, font=("courier", 15, "bold"))
    for i in gan0:
        moverX0(0, gan0[j:j+2])

```

```

        j+=2
#cerrar documentos
caminosX.close()
caminosO.close()
ganadorO.close()
ganadorX.close()

ventana.clear()#limpiamos imagen
ventana.bgpic(ganadorImg)#definimos otro fondo
tortudor=turtle#generamos otra tortuga
tortudor.hideturtle()#ocultamos tortuga

if ganador==1:
    tortudor.speed(0)
    tortudor.color("black")
    tortudor.write("El ganador es X",font=("courier",40, "bold"),align=CENTER)#escribimos el ganador
else:
    tortudor.speed(0)
    tortudor.color("black")
    tortudor.write("El ganador es O",font=("courier",40, "bold"),align=CENTER)

ventana.mainloop()#La ventana corra todo el programa

```


3.2 Explicación

Este programa se desarrollo en el lenguaje python, primeramente definí la función que genera todas la rutas, recibe los parámetros de estado, cadena, camino y la ficha, por ejemplo si entra en el estado 1 agrega a la variable camino que es un string 01 indicando que se encuentra en este estado, después se comprueba que la cadena no sea de tamaño 1 si esto no es correcto entonces le quitamos el primer carácter para ir reduciendo en cada recursividad el tamaño de la cadena y se comprueba si el carácter es 'b' y 'n', dependiendo en que estado se encuentre se moverá a uno del color indicado cada estado es diferente, y se manda a llamara a si mismo pero con los parámetros cambiado como la cadena, el estado, camino. Si el tamaño de la cadena es de 1 significa que termino la recursividad y se escribe en el archivo de la ficha que corresponde y se llama un return para terminar esa función recursiva, de esta forma recorreremos en forma de árbol todos los caminos posibles para esa cadena en especifico. De igual forma se realizo en todos los estados.

En la función de recalcular ruta lo que se hace es recortar la cadena que dio el programa o usuario hasta el estado en que se encuentra y a partir de ese estado recalcular la ruta con la función calcular ruta, una vez que se vuelven a llenar las listas de rutas ganadoras se revuelven y con un ciclo for se recorre la lista y se elige una que no entre en conflicto con la otra ficha si no se pude entonces se regresa un uno indicando que se debe de saltar un turno.

En la función ruta de imagen se obtiene la ruta absoluta de las imágenes que se utilizan en la graficación

Después realice una función para mover las fichas por el tablero que lo único que hacen es recibir un estado y se mueve hacia esas coordenadas

Después de las funciones empieza el menú donde se pregunta si quieren que se realice de forma manual o automática y se abren los archivos que serán escritos, así como se declaran todas las variables a utilizar se genera el tamaño de la cadena con un random y mediante un while se generan las cadenas de 'b' y 'n' de este tamaño después se intenta generar la ruta si esto no se puede se vacían las listas y los strings y se intenta de nuevo. Si esto no se quiere hacer de manera manual tendrá que introducir cada uno

de los valores que se requieren (cadenas, y numero de piezas a jugar).

Después creamos la ventana con todas las imágenes que usaremos estas son las fichas, el tablero y la imagen de ganador, colocamos las fichas en el lugar que le corresponde y asignamos el turno si es el caso que existan dos piezas.

Cree una función para limpiar el cuadro cada vez que se recalcule la ruta

Después se colocan las cadenas de cada ficha para saber que cadena estamos viendo y se eligen al ganador o ganadores. Después mediante un ciclo se empezaran a mover las fichas y dependiendo de quien sea el turno se moverá primero, dentro de esto se comprobara si estos se cruzaran y se recalcula la ruta si esto se puede solo cambia la ruta sino se salta de turno, después tenemos condicionales que comprueba si llego al estado ganador si es así se rompe el ciclo. Después limpiamos la ventana y colocamos la imagen ganadora junto con la ficha ganadora y seria todo el programa.

Camino X:

[illegible]

Ganador O:

```
ganadorO.txt
1 000100050001051015,000100050004051015,000105000500051015,000105000508051015,000105000508131015,000105000502051015,000105000502071015,000105000510051015,
```

Ganador X:

```
ganadorX.txt
1 03070205081312,03070205101312,03070207101312,03071005081312,03071005101312,03071007101312,03071015101312,03071013081312,03071013101312,03020
```

4 Conclusiones

El programa fue bastante difícil ya que no sabia como generar las rutas de cada ficha y estuve unos cuantos días pensando hasta que se me ocurrió con recursividad podría ser resuelto, y estuve en lo correcto, otra cosa que se me complico un poco fue como saltar turnos ya que tuve que estar probando y pensando en las posiciones de la cadena para saber donde se podrían encontrar e intentar corregirlo y sino saltar un turno a la ficha que corresponde, además la graficación como siempre es mas tardado ya que tengo que poner todas las coordenadas exactas y toma mucho tiempo

5 Bibliografia

Hopcroft, J. E. (1979). Introduction to Automata Theory, Languages, and Computation [Paperback] [Jan 01, 1979] John E. Hopcroft (2.a ed., Vol. 2). Pearson.