Trabajo Práctico 1
Pirámide de Palitos

[75.40] Algoritmos y Programación I

Alumno	Pecuch Agustina
Padrón	111560
DNI	40720147
Email	apecuch@fi.uba.ar
Fecha de Entrega	24 de Octubre de 2023

Índice

1. Introducción

El presente informe reúne la documentación solicitada para el juego “Pirámide de Palitos” propuesto en el 1er Trabajo Practico de la materia [75.40] Algoritmos y Programación I.

2. Estructuras utilizadas

+ Pirámide Palitos

- Lista de listas de listas.
- Cada lista mayor corresponde a 1 fila de la pirámide.
- Por cada palito dentro de la misma fila, se utiliza otra lista menor.
- La lista que corresponde a cada palito está conformada por:
    - str: “|”.
    - bool: True para indicar si el palito es rojo y False para caso contrario.
    - int: Cantidad de turnos que dicho palito estará bloqueado.

Ej:

```
[ [ [ “|”, True, 0 ] ],                          >>  FILA 1
[ [ “|”, False, 3 ], [ “|”, True, 0 ] ] ]        >>  FILA 2
```

+ Jugadores

- Lista de listas.
- Cada lista interna corresponde a la data de 1 jugador
    - str: nombre del jugador.
    - int: cantidad de turnos que el jugador debe saltearse.
    - int: cantidad de palitos que lleva sacado hasta el momento.

Ej:

```
[ [ “Jugador 1”, 0, 0 ],          >>  FILA 1
  [ “Maquina1”, 0, 0 ] ]          >>  FILA 2
```

3. Librerías utilizadas.

- **random** : Utilizado para la elección de palitos de la máquina y otros valores aleatórios.
- **os** import system, name : Utilizado para limpiar la consola.
- **time** import sleep : Utilizado para detener el juego y que no se borre tan rápido la consola.
- **math** import sqrt, ceil : Utilizado para calcular el promedio de los palitos.
- **colorama** import init, Fore, Style : Utilizado con proposito decorativo.

4. Flujo del programa.

El juego comienza consultando al usuario con cuantos jugadores desea jugar. El número ingresado debe ser 2 o más. Luego se le solicita que ingrese su nombre y por último con cuantos palitos desea jugar.

- Dependiendo de la cantidad de jugadores que el usuario ingresó se crean jugadores IA con el nombre de “Maquina*” (* corresponde al número de jugador).
- Dependiendo de la cantidad de palitos que el usuario ingresó se procede a crear una pirámide perfecta (si la cantidad ingresada no corresponde a una pirámide perfecta se utiliza la cantidad mayor mas aproximada).

El primer turno del juego corresponde al usuario. Al inicio de cada turno se evalúa si el jugador actúa no posee ningún “turno bloqueado”. En caso de que lo tenga se continúa con el próximo jugador. En caso contrario, se le muestra la pirámide al inicio del turno y se le solicita ingresar cuantos palitos va a desear retirar en este turno. Dependiendo de la cantidad ingresada se le va solicitando X veces que ingrese las coordenadas del palito a retirar. A medida que se van retirando palitos la pirámide se va reacomodando de modo que los palitos de filas superiores van ocupando los espacios vacíos. Luego de quitar 1 palito, se le muestra al usuario la pirámide al momento ya acomodada.

Al finalizar el turno, en caso de que el jugador haya retirado algún “palito rojo” ocurrirá un evento. Para esto se “arroja un dado” y dependiendo del número que salga (del 1 al 6) se procederá a evaluar que evento debe ocurrir.

El proceso de juego se repite hasta que ya no queden palitos en la pirámide para retirar.

Al finalizar, se indica el jugador perdedor que retiro el último palito de la pirámide y cuantos palitos retiro en la totalidad del juego.

5. Funciones

clear() -> None	Esta función limpia la consola utilizando la librería “Os”.
iniciarJuego()-> None	Esta función simplemente imprime un título con color al inicio del juego.
finalizarJuego(jugador:list) -> None	Esta función simplemente recibe el jugador perdedor como parámetro y lo imprime junto con la cantidad de palitos que retiro.
elegirJugadores()-> list	Esta función inicializa la lista de jugadores. Dependiendo de la cantidad de jugadores que se ingrese se crearan: - Un jugador real con el nombre del jugador. - El resto de los jugadores serán maquinas con el nombre "Maquina(nro)". A cada jugador también se le asigna dos valores int para indicar la cantidad de turnos que debe perder y la cantidad de palitos que lleva sacados. Al comienzo del juego a todos los jugadores se les asigna "0"
calcularCantidadPalitosMinima (cantidadJugadores:int) -> int	Esta función calcula la cantidad de palitos MINIMA para inicializar el tablero, basada en la cantidad de jugadores.
calcularCantidadPalitos(cantidadJugadores:int) -> int	Esta función consulta al usuario con cuantos palitos desea jugar y chequea que no sea menor a la cantidad mínima requerida.
armarTablero(cantidadPalitos:int) -> list	Esta función arma el tablero inicial dependiendo de la cantidad de palitos. El tablero está conformado por una lista de listas - Cada FILA es una lista. - Dentro de cada fila hay una lista por PALITO, compuesta por: - str: "|" - bool: T si el palito es rojo. F si el palito NO es rojo. - int: indica la cantidad de turnos bloqueados para el palito.
colocarPalitosRojos(tablero:list, cantidadPalitos:int) -> list	Esta función toma el 30% de la cantidad de palitos y se los asigna como "palitos rojos". El segundo elemento de cada lista correspondiente a cada palito es asignado como: True. - La elección de que palito será asignado como rojo se realiza de forma random.
mostrarTablero(tablero:list) -> None	Esta función imprime en pantalla el tablero de una forma más amigable para el usuario. - A la izquierda se imprime el número de fila - Abajo se imprime la posición.
mostrarTableroInicial(tablero:list) -> None	Esta función imprime en pantalla el tablero de una forma mas amigable para el usuario y mostrando los palitos que son rojos.

acomodarTablero(tablero:list) -> list	Esta función reacomoda el tablero para evitar que haya espacios vacios en el medio de una fila. El lugar vacío se ocupa con el primer palito de la fila superior.
inicializarTurno(jugador:str, tablero:list) -> None	Esta función inicializa el turno para cada jugador. Se imprime en pantalla a quien corresponde el turno y el tablero al inicio.
quitarPalitosJugador(tablero:list, cantidadPalitos:int) -> list	Esta función retira los palitos indicados por el jugador.
quitarPalitosMaquina(tablero:list, cantidadPalitos:int) -> list	Esta función retira los palitos indicados por la máquina.
evaluarEventos(tablero:list, cantidadPalitos:int, jugador:list, cantidadInicialPalitos:int) -> None	Esta función se utiliza cuando un jugador retiro algún palito rojo. Se elige un numero random del 1 al 6. Cada número inicializa un evento distinto.
eventoUno(jugador:list) -> list	Esta función corresponde al Evento 1: - El jugador pierde su próximo turno.
eventoDos(tablero:list, cantidadInicialPalitos:int, cantidadPalitos:int) -> list	Esta función corresponde al Evento 2: - Se agregan entre 1 y M palitos, en caso de no poder agregarse a totalidad, se agrega el máximo que se pueda (no superar cantidad inicial de palitos).
eventoTres(tablero:list, cantidadPalitos:int) -> list	Esta función corresponde al Evento 3: - Se bloquean/congelan el 20% de los palitos (en el caso que el valor sea < 1 entonces se toma 1 palito) , durante 3 turnos los jugadores no podrán retirarlos.
eventoCuatro(tablero:list, jugador:str, cantidadPalitos:int) -> list	Esta función corresponde al Evento 4: - Bomba: el jugador debe retirar una fila completa a su elección.
eventoCinco(tablero:list, cantidadInicialPalitos:int) -> list	Esta función corresponde al Evento 5: - Nueva pirámide: se genera una nueva pirámide, del mismo tamaño que la original.
main() -> None	Función principal donde se desarrolla el juego.

6. Dificultades.

En relación con mi experiencia personal durante el desarrollo de este Trabajo Practico, donde encontré mayores dificultades fue en la implementación del acomodo de la pirámide y la jugabilidad por turnos.

En cuanto al acomodo de la pirámide me llevo bastante tiempo descubrir una forma de recorrer el tablero de forma inversa sin perder la estructura original del mismo.

En cuanto a la jugabilidad por turnos, se me dificulto a la hora de implementar que ciertos jugadores debían “perder su turno”.