

Trabajo Práctico N° 1

[75.40] Algoritmos y Programación I

Pirámide de palitos

Introducción

En este documento se detallan las estructuras que elegí para el desarrollo de este trabajo y una breve explicación del flujo del programa.

La ejecución principal se desarrolla en un método *main*. Dentro del mismo se crean las estructuras necesarias para el inicio del juego en sí, como la pirámide y los jugadores.

En cuanto a la pirámide, decidí utilizar una lista de listas de diccionarios. La misma podría verse de la siguiente manera (es solo una imagen ilustrativa con un solo “palito” de ejemplo, teniendo en cuenta la cantidad mínima de palitos esto no sería posible):

```
piramide: list[list[dict]] = [  
    [  
        {  
            'es_rojo': False,  
            'esta_congelado': False,  
            'contador_congelado': 0,  
            'fue_eliminado': False  
        }  
    ]  
]
```

Cada diccionario representa a un palito, el cual contiene sus atributos, los mismos se irán modificando a lo largo de la ejecución. Cada lista dentro de la principal tendrá un diccionario más que la anterior, formando así la pirámide perfecta.

Para los jugadores seguí una lógica similar. Utilicé una lista de diccionarios para representarlos. La lista con dos jugadores podría verse de la siguiente manera:

```
jugadores: list[dict] = [  
    {'palitos_retirados': 0, 'pierde_turno': False, 'es_maquina': False, 'numero_jugador': 0},  
    {'palitos_retirados': 0, 'pierde_turno': False, 'es_maquina': True, 'numero_jugador': 1}  
]
```

El jugador de número 0 representará al usuario, el resto serán controlados por la máquina.

1: Método *main()*

En el método *main()* se le pide al usuario las condiciones principales para dar inicio al programa: cantidad de jugadores y cantidad de palitos que tendrá la pirámide. Con ello se arman las estructuras correspondientes detalladas anteriormente para poder dar inicio al juego (incluye la asignación de palitos rojos sobre el 30% de la totalidad de los mismos). Cada turno se desarrolla dentro del método *jugar_turno()*. Al finalizar el turno, se valida si el juego terminó, es decir si se eliminó la totalidad de palitos, y si es así, se muestran los resultados finales.

2: Método *jugar_turno()*

Dentro del método *jugar_turno()* se desarrolla la lógica principal, ya que representa justamente el turno de cada jugador. En el mismo se piden los palitos a eliminar y se ejecuta la acción correspondiente. Cabe destacar que en todo momento se valida si el jugador es controlado por la máquina y se adapta el juego a ello.

3: Eliminar palitos y eventos

El objetivo principal del juego es ir eliminando palitos. Durante el turno de cada jugador se le piden la cantidad y las coordenadas de el o los palitos que quiere retirar, se valida que sea posible (que no esté congelado, que no haya sido eliminado, que sea una posición válida de la pirámide) y finalmente se elimina. Para ello, no se eliminan elementos de las listas, la misma permanece con la misma estructura, sino que se lo “marca” como eliminado, mediante la clave “*fue_eliminado*” dentro del diccionario correspondiente. Luego, se reacomodan los palitos, para que las filas inferiores estén completas. En este proceso se toman las características del palito que se moverá de lugar y las mismas se mantienen en la nueva posición. Es decir, si un palito estaba congelado, lo seguirá estando, y lo mismo ocurre con sus otros atributos.

Algo muy importante que sucede al eliminarse el palito es la validación de si se trata de uno rojo, ya que de ser así se dispara un evento al azar. Los mismos se desarrollan como se detalla a continuación:

1. El jugador pierde su próximo turno: En el diccionario del jugador se aplica una marca sobre el valor de la clave “*pierde_turno*”. Una vez pasado el siguiente turno, la misma volvería a *False*.
2. Se agregan entre 1 y M palitos: Se pide una cantidad de palitos, de superar el tamaño de la pirámide original, se elimina el sobrante. Los mismos se agregan desde las filas inferiores, para que no sea necesario reacomodar los palitos.
3. Se bloquean/congelan el 20% de los palitos: Se calcula la cantidad de palitos que serán congelados sobre los que no hayan sido todavía eliminados de la pirámide y se asignan al azar. Para ello, se modifican sobre el diccionario los valores de las claves “*esta_congelado*”, el cual toma un valor *True*, y “*contador_congelado*” el cual

representa la cantidad de turnos por los cuales permanece congelado el palito y se le suma 3 (si ya estaba congelado, se suman los turnos anteriores). Con el pasar de los turnos este contador va descendiendo y al llegar a 0, deja de estar congelado. Los palitos congelados se pintarán de azul al imprimir la pirámide.

4. Bomba: el jugador debe retirar una fila completa a su elección: Se verifica que la fila tenga algún palito y se eliminan todos los que no estén congelados. Luego, se reacomodan de ser necesario.
5. Nueva pirámide: Reutilizo el método para armar la pirámide al inicio del juego y se asignan palitos rojos de la misma manera.
6. No hace nada: Sigue el flujo normal del juego

Observaciones

- A pesar de que respecto a la experiencia de juego creo que sería mejor no mostrar los palitos rojos porque le agrega un elemento de sorpresa, decidí mostrarlos en todo momento. Me pareció que iba a ser más gráfico de esta forma y además simplificaría el código, ya que mostrarlos solo cuando se los selecciona por ej. implicaría crear un nuevo método o agregar validaciones que no son necesarias para imprimir la pirámide en el resto del flujo.
- En el final del juego, planteé que el jugador que saca el último palito pierde, pero a su vez, quien haya eliminado más durante la partida, gana. Por lo que, el mismo jugador podría ser perdedor y ganador al mismo tiempo.
- Tanto los palitos eliminados durante el turno, es decir, los elegidos por el jugador, como los eliminados por un evento se suman a la cantidad del jugador. Este valor se guarda con la clave *"palitos_retirados"*
- Para poder apreciar mejor las acciones de los jugadores "controlados por la máquina" decidí no limpiar la consola luego de cada turno sino al finalizar la ronda con el método *limpiar_consola()*.
- Los palitos congelados pueden moverse (por ej. al *reacomodar_palitos()*) pero no pueden ser eliminados.
- Si solo quedaran palitos congelados, se saltean los turnos hasta que se puedan eliminar. Para saberlo se compara la cantidad total de palitos con la cantidad de congelados, si son iguales termina el turno. Hay tres escenarios en donde considero esto: al iniciar el turno, antes de eliminar cualquier palito, o bien podría suceder ya iniciado el turno (al eliminar más de un palito o al querer eliminar una fila por un evento)
- Si el último palito eliminado resulta ser rojo, no se dispara un evento y termina el juego.