

Implementación del Juego de la Oca. Valor: 15%

I. DESCRIPCIÓN DEL PROBLEMA

La Oca es un juego de mesa para dos o más jugadores. Cada jugador tira un dado y avanza su ficha (de acuerdo al número obtenido) por un tablero en forma de caracol con 63 casillas. Dependiendo de la casilla en la que se caiga, se puede avanzar o por el contrario retroceder, y en algunas de ellas está indicado un castigo o un premio. En su turno cada jugador tira 1 dado que le indica el número de casillas que debe avanzar. Gana el juego el primer jugador que llega a la casilla 63, "el jardín de la oca".

II. OBJETIVO GENERAL

Implementar de forma individual una aplicación de escritorio en C++ que permita a dos jugadores jugar el juego de la Oca. Ver demo del juego en el sitio: <https://www.fandejuegos.com/juego/juego-oca>

III. OBJETIVOS ESPECÍFICOS

- Utilizar matrices o vectores en el código de la aplicación para gestionar el estado de la aplicación.
- Desplegar interfaces de usuario por medio de consola.
- Aplicar los estándares de programación dentro del código (indentación, nombres significativos, lógica simple, comentarios internos).
- Modularizar el código de la solución mediante la implementación de distintas clases según los siguientes comportamientos: interfaz de usuario, Lógica del negocio, controlador del juego (MVC).

IV. REGLAS DEL JUEGO

- Hay un tablero con 63 casillas y 2 jugadores con una ficha cada uno.
- El objetivo es llegar a la última casilla (63).
- Cada jugador tira el dado en su turno y mueve la ficha el número de casillas que indica el dado.
- Si el jugador cae en una casilla con la oca (8,13,18,23,28,33,38,43,48,53,58) duplica la cantidad de pasos obtenida en su lanzamiento de forma automática. Si cae en otra de las casillas especiales, el jugador puede recibir la instrucción de perder uno o más turnos, o avanzar o retroceder un número determinado de espacios:

Casillas especiales:

- **(O) La oca:** Se encuentran ubicadas en las casillas: 8, 13, 18, 23, 28, 33, 38, 43, 48, 53, 58. Cuando se cae en una de estas casillas, se avanza de forma automática la misma cantidad de pasos obtenida en su lanzamiento. Tradicionalmente se dice "de oca a oca y tiro porque me toca".
- **(P) Los puentes:** Se encuentran normalmente ubicados en las casillas 6 y 12. Si se cae en un puente se mueve la ficha al otro puente y se vuelven a tirar los dados diciendo "de puente a puente y tiro porque me lleva la corriente". Es decir, si se cae en la casilla 6 se avanza a la 12 y si se cae en la 12 se retrocede a la 6, tirando de nuevo los dados en ambos casos.
- **(H) El hospedaje:** Se encuentra normalmente ubicada en la casilla 19. Cuando se cae en esta casilla, suelen perderse 2 turnos.
- **(L) El laberinto:** Ubicado en la casilla 42, cuando se cae en el laberinto, se está obligado a retroceder a la casilla 30. Se suele decir: "Del laberinto al 30".
- **(C) La cárcel:** Ubicada en la casilla 52, cuando se cae en la cárcel, la ficha queda inmovilizada hasta que otro jugador caiga y te rescate, quedando éste atrapado.

- **(M) La muerte:** Ubicada en la casilla 56, cuando se cae en la calavera o muerte, se vuelve a empezar desde la casilla de inicio.
- **(J) Casilla 63 (meta) "jardín de la oca":** Para poder llegar a esta casilla, hay que obtener los puntos exactos con el dado, de lo contrario retrocederás tantas casillas como puntos sobren.
- **(J1 Y J2) Jugadores:** cada jugador se puede representar como j1 y j2 o también se le podría solicitar el nombre a cada jugador y anotarlo como identificador en el juego.

V. ESTRUCTURAS DE DATOS ESPERADA

Matriz de juego

Usar una matriz para representar las posiciones del juego. La matriz será de tamaño 8X8 donde cada posición se representa con su respectivo número y las casillas especiales se representan con su respectiva letra o nombre:

J	62	61	60	59	O	57	M
O	49	50	51	C	O	54	55
47	46	45	44	O	L	41	40
32	O	34	35	36	37	O	39
31	30	29	O	27	26	25	24
16	17	O	H	20	21	22	O
15	14	O	P	11	10	9	O
J1 J2	1	2	3	4	5	P	7

Matriz de estado del juego

Se recomienda usar un segundo arreglo bidimensional para representar el estado del juego. Esta matriz almacena enteros que representan el estado de los jugadores en cada movimiento, por ejemplo:

- 0 representa celdas vacías
- 1 representa a cada oca
- 2 representa el puente de la casilla 6
- 3 representa el puente de la casilla 12
- 4 representa el hospedaje
- 5 representa el laberinto
- 6 representa la cárcel
- 7 representa la muerte
- 8 representa la casilla de gane del juego (63)
- 9 representa el jugador1
- 10 representa el jugador2
- 11 representa los dos jugadores en la misma casilla

8	0	0	0	0	1	0	7
1	0	0	0	6	1	0	0
0	0	0	0	1	5	0	0
0	1	0	0	0	0	1	0
0	0	0	1	0	0	0	0
0	0	1	4	0	0	0	1
0	0	1	3	0	0	0	1
11	0	0	0	0	0	2	0

VI. REQUERIMIENTOS

1. El juego debe solicitar al inicio los nombres de los jugadores y asignarles una identificación (J1, J2) o su id personalizada.
2. El juego finaliza cuando un jugador llega exactamente a la casilla 63.
3. El juego debe tener una o dos matrices para representar su estado.
4. El juego debe indicar quien debe tirar el dado y debe mostrar su valor en cada lanzamiento.
5. El juego debe tener opciones para terminar o empezar de nuevo.

VII. POLÍTICAS DE EVALUACIÓN

Buenas prácticas de programación: Documentación del código, simplicidad de la solución implementada (mantenimiento), convención de nombres, modularidad e indentación.		10%
Defensa del proyecto: lo debe presentar ante el profesor o grupo de clase y probar su funcionalidad o deficiencias.		10%
Funcionalidad	<p>Generar la interfaz de la matriz de juego. (10 pts)</p> <p>Solicita los nombres de los jugadores, opciones para terminar y reiniciar el juego. (10 pts)</p> <p>Avance del juego: solicita lanzar los dados a cada jugador y refresca el estado del juego. (10 pts)</p> <p>Cierre del juego: pérdida o gane. (10 pts)</p> <p>Aplica correctamente las reglas del juego. (20 pts)</p> <p>Aplica el patrón de diseño MVC. (10 pts)</p>	70%
Documentación (digital)	<p>Portada. Descripción del proyecto. Alcances y limitaciones.</p> <p>Diagrama de clases implementado. Conclusiones de la experiencia.</p>	10%