Proyecto *El juego de la oca*

Fecha de entrega de la versión 1: 6 de noviembre de 2022 a las 23:00

(Imagen extraída de *https://www.juegodelaoca.com/Reglamento/reglamento.htm*)

El objetivo del proyecto es desarrollar un programa que permita jugar a una versión del popular Juego de la oca. Iremos desarrollando diversas versiones que permitirán poner en práctica los distintos conocimientos de la asignatura.

1. El juego en detalle

El Juego de la oca es un juego de mesa para dos o más jugadores. Se dispone de un tablero donde hay un camino con 63 casillas numeradas del 1 al 63 que contienen dibujos. Al inicio del juego cada jugador coloca su ficha en la casilla de partida, que es la 1. Cada jugador, en su turno, tira un dado y avanza su ficha, desde la posición en la que se encuentre, tantas casillas como indique el número obtenido en el dado. El juego lo gana el primer jugador que consigue llegar a la última casilla (casilla 63).

En el tablero hay casillas que no tienen ningún efecto añadido –en las que el jugador permanece hasta su siguiente turno– pero también hay casillas especiales que pueden conllevar avances o retrocesos, así como penalizaciones –turnos sin tirar– o premios –tiradas extra.

Estos son los tipos de casillas especiales que vamos a considerar y su efecto:

- **Oca.-** Se trata de una casilla que conlleva avance a la siguiente oca del tablero y proporciona un premio (una tirada extra). Si un jugador cae en una casilla con una oca avanza hasta la siguiente casilla donde haya una oca y, antes de que el turno pase al siguiente jugador, vuelve a tirar el dado una vez más (*de oca a oca y tiro porque me toca*…). La última casilla del tablero (casilla 63) es una oca.
- **Puente.-** Se trata de una casilla que conlleva avance o retroceso a la otra casilla de puente y proporciona un premio (una tirada extra). En el tablero hay dos casillas con puente, y si un jugador cae en una de ellas se va a la otra, ya sea avanzando o

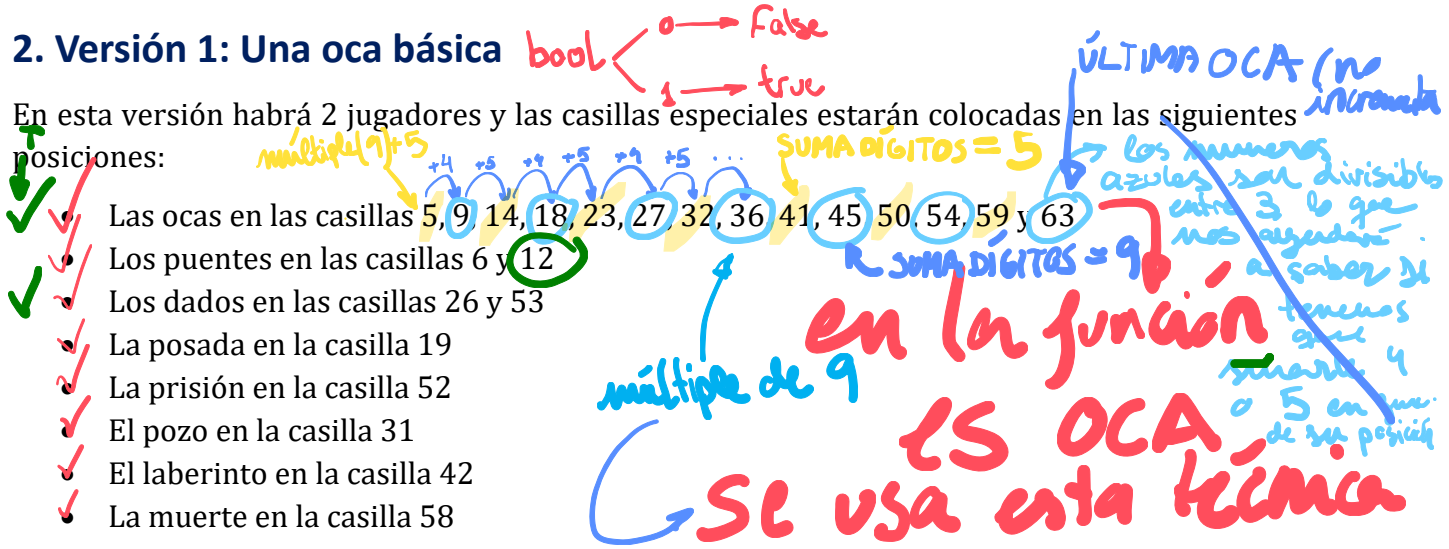
retrocediendo según haya caído en la que se encuentra primero o en la segunda, respectivamente, y, antes de que el turno pase al siguiente jugador, vuelve a tirar el dado una vez más (*de puente a puente y tiro porque me lleva la corriente..*).

- **Dados.-** Se trata de una casilla que conlleva avance o retroceso a la otra casilla de dados y proporciona un premio (una tirada extra). En el tablero hay dos casillas con dados, y cuando el jugador cae en una de ellas se va a la otra, ya sea avanzando o retrocediendo según haya caído en la primera o en la segunda, respectivamente, y, antes de que el turno pase al siguiente jugador, vuelve a tirar el dado una vez más (*de dados a dados y tiro porque me ha tocado…*).
- **Posada.**- Se trata de una casilla que conlleva una penalización. En el tablero hay una única casilla para la posada y su efecto es que el jugador que cae en ella se queda un turno sin jugar (es decir, en su siguiente turno, el jugador no tirará los dados).
- **Prisión.**- Se trata de una casilla que conlleva una penalización. Hay una única casilla para la prisión en el tablero y el jugador que cae en ella se queda dos turnos sin jugar.
- **Pozo.**- Se trata de una casilla que conlleva una penalización. Solo hay una casilla para el pozo en el tablero. El jugador que cae en el pozo se queda tres turnos sin jugar.
- **Laberinto.**- Se trata de una casilla que conlleva un retroceso. Hay una única casilla para el laberinto en el tablero y su efecto es que el jugador retrocede 12 casillas.
- **Muerte.**- Se trata de una casilla que conlleva un retroceso. Solo hay una casilla de muerte en el tablero, y el jugador que cae en ella vuelve a la casilla de partida (casilla 1).

2. Versión 1: Una oca básica

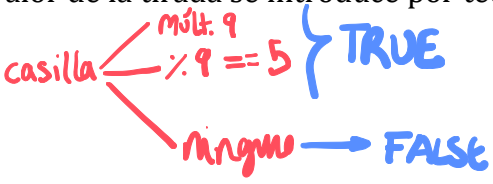
En esta versión habrá 2 jugadores y las casillas especiales estarán colocadas en las siguientes posiciones:

- Las ocas en las casillas 5, 9, 14, 18, 23, 27, 32, 36, 41, 45, 50, 54, 59 y 63
- Los puentes en las casillas 6 y 12
- Los dados en las casillas 26 y 53
- La posada en la casilla 19
- La prisión en la casilla 52
- El pozo en la casilla 31
- El laberinto en la casilla 42
- La muerte en la casilla 58



El juego lo gana el primer jugador que consigue llegar a la última casilla del camino (casilla 63), bien sea obteniendo en el dado un número de puntos igual o superior a los que le falten para alcanzar dicha casilla, bien sea cayendo en la oca inmediatamente anterior (casilla 59).

El programa debe permitir jugar una partida en la que se pueda establecer si el lanzamiento del dado es aleatorio (modo normal) o bien si el valor de la tirada se introduce por teclado (modo depuración).



2.1. Detalles de implementación

Definiremos constantes que permitan:

- Representar casillas especiales. Serán constantes de tipo `int`. Por ejemplo: `NUM_CASILLAS = 63` para representar la última casilla, `CASILLA_PUENTE_1 = 6` y `CASILLA_PUENTE_2 = 12` para representar las casillas de los puentes; `CASILLA_DADOS_1 = 26` y `CASILLA_DADOS_2 = 53` para representar las casillas de los dados; etc.
- Representar el número de turnos de penalización (sin tirar). Serán constantes de tipo `int`. Por ejemplo: `TURNOS_POSADA = 1` representa la penalización por caer en la posada; etc.

Implementaremos, entre otras, las funciones que se indican a continuación. PRIMERAS FUNCIONES

En primer lugar, funciones que verifican si una casilla es de un tipo especial:

- `bool esOca(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde o no con una oca.
- `bool esPuente(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con un puente.
- `bool esDados(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con unos dados.
- `bool esLaberinto(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con el laberinto.
- `bool esMuerte(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con la muerte.
- `bool esPosada(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con la posada.
- `bool esPrision(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con la prisión.
- `bool esPozo(int casilla)`: devuelve un valor que indica si la casilla `casilla` se corresponde –`true`– o no –`false`– con el pozo.
- `bool esMeta(int casilla)`: devuelve `true` si la casilla `casilla` es igual o superior a la última (casilla 63) y `false` en caso contrario.

En segundo lugar, funciones que calculan la casilla de destino, habiendo caído en una casilla especial de oca, puente, dados, laberinto o muerte: SEGUNDAS FUNCIONES

- `int siguienteOca(int casilla)`: dada la casilla `casilla` correspondiente a una oca devuelve la posición de la siguiente oca de la secuencia.
- `int siguientePuente(int casilla)`: dada la casilla `casilla` correspondiente a un puente devuelve la posición del otro puente.
- `int siguienteDado(int casilla)`: dada la casilla `casilla` correspondiente a unos dados devuelve la posición de los otros dados.
- `int siguienteLaberinto()`: devuelve una posición resultado de retroceder 12 casillas desde el laberinto.
- `int siguienteMuerte()`: devuelve la posición resultado de caer en la muerte, es decir, devuelve la casilla de partida (la casilla 1).

En tercer lugar, tendremos funciones que regulan el juego: TERCERAS FUNCIONES

- `int tirarDado()`: devuelve un número en el intervalo [1,6] determinado de manera aleatoria, y que representa la tirada del dado.
- `int quienEmpieza()`: devuelve un número en el intervalo [1,2] determinado de manera aleatoria, indicando el jugador que comienza la partida.
- `int efectoPosicion(int casillaActual)`: dada la casilla en la que está el jugador `casillaActual` determina si es una casilla especial que conlleva un avance o un retroceso y devuelve la casilla a la que avanza o retrocede (por ejemplo, si `casillaActual` es una oca diferente a la de la última casilla devuelve la casilla de la siguiente oca); si no es una casilla especial, o bien es una casilla especial cuyo efecto es una penalización, es decir, que no conlleva ni avance ni retroceso –posada, prisión, pozo–, se devuelve la misma casilla que se ha recibido. Además, mostrará por pantalla la casilla en la que se está (`casillaActual`), y, si procede, la casilla a la que se avanza o se retrocede.
- `int efectoTiradas(int casillaActual, int numeroDeTiradas)`: recibe la casilla en la que está el jugador (`casillaActual`) y el número de tiradas de que dispone en su turno (`numeroDeTiradas`) y devuelve el número de tiradas actualizado teniendo en cuenta el efecto de `casillaActual` sobre las tiradas, si es que tiene alguno. La actualización consiste en restar a `numeroDeTiradas` tantas tiradas como marque la penalización (1 si es la posada, 2 si es la prisión y 3 si es el pozo) o sumar tantas tiradas como indique el premio (gana 1 tirada tanto si es oca como si es puente o dados); el resto de casillas no modifican el número de tiradas, devolviéndose en ese caso el mismo valor que se ha recibido.

2.1.1. Generación de números aleatorios

Ten en cuenta que para generar los números aleatorios usaremos las funciones `srand(semilla)` y `rand()` de la biblioteca `cstdlib`. Una secuencia de números aleatorios comienza en un primer número entero que se denomina semilla.

Para establecer la semilla el programa deberá invocar a la función `srand()` con el argumento deseado. Lo que hace que la secuencia se comporte de forma aleatoria es precisamente la semilla. Una semilla habitual es el valor de la hora del sistema que se obtiene con una invocación a `time(NULL)` (biblioteca `ctime`), ya que así es siempre distinta para cada ejecución. Así pues, el programa deberá ejecutar `srand(time(NULL))` (una sola vez, al principio del programa principal).

Una vez establecida la semilla, la función `rand()` genera, de forma pseudoaleatoria, un entero positivo a partir del anterior. Por ejemplo, si quieres que los números aleatorios generados estén en un determinado intervalo, deberás utilizar el operador %. Así, para obtener un entero aleatorio en el intervalo [limiteInferior, limiteSuperior] hay que usar la expresión `limiteInferior + rand() % (limiteSuperior+1-limiteInferior)`.

2.1.2. El modo depuración

Para facilitar las pruebas y depuración del programa, tendremos dos modos de juego, el normal y el de depuración. El modo depuración será exactamente igual que el modo de juego normal con la salvedad de que, en lugar de generar aleatoriamente la tirada del dado, se permitirá al jugador introducir desde teclado el valor de la misma (pudiendo ser incluso un valor fuera del rango entre 1 y 6 a fin de facilitar la realización de las pruebas).

Para implementar el modo depuración usaremos una constante `MODO_DEBUG` de tipo `bool` que pondremos a `true` si cuando ejecutemos el programa queremos jugar una partida en modo depuración y a `false` para jugar en modo normal. Comprobando el valor de esta constante generaremos el valor de la tirada de dados aleatoriamente (mediante el uso de la función `tirarDado`) o permitiremos su introducción por teclado. Para su introducción por teclado implementaremos la función `int tirarDadoManual()` que lee de teclado un valor entero y lo devuelve.

2.1.3. Ejemplos de ejecución

En la siguiente captura de pantalla podemos ver un extracto de una ejecución del programa en modo normal. Aleatoriamente le ha correspondido al jugador 1 iniciar la partida. Puede verse el efecto de caer en casillas que conllevan avance/retroceso y premio (oca y puente).

```
****  EMPIEZA EL JUGADOR: 1****
CASILLA ACTUAL: 1
INTRODUCE EL VALOR DEL DADO: 4
VALOR DEL DADO: 4
PASAS A LA CASILLA 5
SALTAS A LA SIGUIENTE OCA EN LA CASILLA: 9
VUELVES A TIRAR
CASILLA ACTUAL: 9
INTRODUCE EL VALOR DEL DADO: 1
VALOR DEL DADO: 1
PASAS A LA CASILLA 10

TURNO PARA EL JUGADOR 2
CASILLA ACTUAL: 1
INTRODUCE EL VALOR DEL DADO: 3
VALOR DEL DADO: 3
PASAS A LA CASILLA 4

TURNO PARA EL JUGADOR 1
CASILLA ACTUAL: 10
INTRODUCE EL VALOR DEL DADO: 2
VALOR DEL DADO: 2
PASAS A LA CASILLA 12
DE PUENTE A PUENTE Y TIRO PORQUE ME LLEVA LA CORRIENTE
SALTAS AL PUENTE DE LA CASILLA 6
VUELVES A TIRAR
CASILLA ACTUAL: 6
INTRODUCE EL VALOR DEL DADO: 2
VALOR DEL DADO: 2
PASAS A LA CASILLA 8

TURNO PARA EL JUGADOR 2
CASILLA ACTUAL: 4
INTRODUCE EL VALOR DEL DADO:
```

Turno del jugador 1: el dado saca un 4, cayendo en una oca (casilla 5). Esto provoca: un avance a la siguiente oca (casilla 9) y ganar una tirada. En la segunda tirada de su turno el dado saca un 1 y el jugador alcanza la casilla 10 (casilla sin efecto añadido) en la que permanece hasta su siguiente turno.

Turno del jugador 2: el dado saca un 3, cayendo en la casilla 4 (casilla sin efecto añadido) en la que permanece hasta su siguiente turno.

Nuevo turno del jugador 1: el dado saca un 2, cayendo así en la segunda casilla de puente (casilla 12). Esto provoca: un retroceso al anterior puente (casilla 6) y ganar una tirada. En la segunda tirada de este turno el dado saca un 2 y el jugador alcanza la casilla 8 (casilla sin efecto añadido) en la que permanece hasta su siguiente turno.

Nuevo turno del jugador 2…

En esta segunda captura puede verse una ejecución en modo depuración, donde de nuevo aleatoriamente le ha correspondido al jugador 1 iniciar la partida. Puede observarse el efecto de caer en casillas que conllevan penalización (posada) y avance/retroceso junto con premio (dados). También puede observarse como un jugador gana el juego.

```
****  EMPIEZA EL JUGADOR: 1****
CASILLA ACTUAL: 1
INTRODUCE EL VALOR DEL DADO: 18
VALOR DEL DADO: 18
PASAS A LA CASILLA 19
HAS CAIDO EN LA POSADA.
PIERDES 1 TURNOS

TURNO PARA EL JUGADOR 2
CASILLA ACTUAL: 1
INTRODUCE EL VALOR DEL DADO: 25
VALOR DEL DADO: 25
PASAS A LA CASILLA 26
DE DADO A DADO Y TIRO PORQUE ME HA TOCADO
SALTAS AL SIGUIENTE DADO EN LA CASILLA: 53
VUELVES A TIRAR
CASILLA ACTUAL: 53
INTRODUCE EL VALOR DEL DADO: 2
VALOR DEL DADO: 2
PASAS A LA CASILLA 55
CASILLA ACTUAL: 55
INTRODUCE EL VALOR DEL DADO: 5
VALOR DEL DADO: 5
PASAS A LA CASILLA 60

TURNO PARA EL JUGADOR 1
CASILLA ACTUAL: 19
INTRODUCE EL VALOR DEL DADO: 3
VALOR DEL DADO: 3
PASAS A LA CASILLA 22

TURNO PARA EL JUGADOR 2
CASILLA ACTUAL: 60
INTRODUCE EL VALOR DEL DADO: 5
VALOR DEL DADO: 5
PASAS A LA CASILLA 65

------ GANA EL JUGADOR 2---------
```

Turno del jugador 1: se introduce un 18 como tirada y cae en la posada (casilla 19), lo que conlleva penalización de 1 turno si tirar.

Turno del jugador 2: se introduce un 25 como tirada y cae en los primeros dados (casilla 26), lo que conlleva un avance a los segundos dados (casilla 53) y ganar una tirada. En la tirada extra que acaba de ganar se introduce un 2, yendo a parar a la casilla 55 (sin efecto añadido). En estos momentos sería el turno del jugador 1, pero como tiene penalización de 1 turno, en la práctica significa que el jugador 2 dispone de otra tirada, en la que se introduce un 5 y alcanza la casilla 60 (sin efecto añadido).

Nuevo turno del jugador 1: se introduce un 3 como tirada y va de la casilla 19 a la 22. La nueva posición no tiene efecto añadido.

Nuevo turno del jugador 2: se introduce un 5 como tirada y va de la casilla 60 a la 65, es decir, supera la casilla final. El jugador 2 gana el juego y finaliza la ejecución.

- constantes ✓
- primeras funciones ✓ FUNCIÓN DEBUG ✓
- testear primeras funciones ✓
- siguientes funciones ✓
- testear siguientes funciones ✓
- terceras funciones 3/4
- testear terceras funciones
- juntar todo — ESTRUCTURA
- testear todo
- arreglar errores
- CHEQUEAR REQUISITOS!!!
- TESTEAR!!!
- Documentar

Entregar!

to-do

- última parte efectoposicion();
- int efectoTiradas();
- Testear terceras variables ✓

FALTA:

- implementar → 2 jugadores
  - turnos ⇕ penalizaciones