

## DOCUMENTACIÓN DE LOS JUEGOS:

### Juego 1:

- **Título:** La Oca
- **Estudio/Diseñadores:** David Blanco, Sergi Pifarré y Daniel Romero
- **Plataforma:** entorno de programación como, por ejemplo, Eclipse.
- **Versión:**1.0
- **Licencia:** basado en el tradicional juego de la Oca.
- **Mecánica:** el tradicional juego de la oca, en el que dos jugadores tienen que ir tirando el dado por turnos hasta llegar al final del tablero. Algunas casillas tienen unas características propias que favorecen o perjudican cada turno de los jugadores. Estas están especificadas en el apartado de información del módulo de juegos.
- **Modos de juego:** se puede jugar contra otro jugador o contra la IA.
- **Tecnología:** se juega por consola y ha sido programado en Java.
- **Público:** dirigido a todo tipo de público, pero sobre todo a niños y jóvenes. A la gente a la que le guste divertirse con juegos tradicionales.

## Juego 2:

- **Título:** El pistolero
- **Estudio/Diseñadores:** David Blanco, Sergi Pifarré y Daniel Romero
- **Plataforma:** entorno de programación como, por ejemplo, Eclipse.
- **Versión:** 1.0
- **Licencia:** basado en el tradicional juego de manos del pistolero.
- **Mecánica:** los jugadores deben escoger entre tres opciones que se les proporciona, que son: Introduciendo un 0 en la consola recargan su arma, introduciendo un 1 disparan y, por último, introduciendo un 2 se cubren del ataque enemigo. En total tienen 5 vidas y para lograr ganar se tienen que restar todas las del contrincante.
- **Modos de juego:** se puede jugar contra otro jugador o contra la IA.
- **Tecnología:** se juega por consola y ha sido programado en Java.
- **Público:** a personas de entre 15 y 30 años, aunque lo puede jugar quien quiera.

### Juego 3:

- **Título:** Lucha
- **Estudio/Diseñadores:** David Blanco, Sergi Pifarré y Daniel Romero
- **Plataforma:** entorno de programación como, por ejemplo, Eclipse.
- **Versión:**1.0
- **Licencia:** basado en juegos de lucha.
- **Mecánica:** El juego de lucha es un juego en el que se enfrentan dos combatientes, que según las habilidades que tienen (ataque, escudo, velocidad y salud) se desenvuelven durante el juego de una u otra forma. Por supuesto esas habilidades no se muestran más a lo largo de la partida, así que los jugadores deberán elegir su estrategia según como se va desarrollando la partida. Pueden elegir entre atacar o defender y es aquí donde entran en juego las habilidades, si un jugador elige atacar y el otro defender, el defensor se verá “recompensado” usando su escudo que evitará que le reste el daño directamente a su vida. En el caso de que los dos jugadores indiquen que quieren atacar, el atacante será el jugador que tenga más velocidad y el enemigo se defenderá únicamente con su vida. Por último, en el caso que los dos jugadores decidan defenderse, ninguno recibirá ni provocará ningún daño. En el caso que un combatiente muera, el combatiente vencedor se proclamará ganador.
- **Modos de juego:** se puede jugar contra otro jugador o contra la IA.
- **Tecnología:** se juega por consola y ha sido programado en Java.
- **Público:** dirigido a todo tipo de público, pero sobre todo a niños y jóvenes.

## **Documentación de las funciones:**

### Funciones comunes para los tres juegos:

```
//*****
/** Nombre de la función: main
/** Explicación de lo que hace la función: Muestra la pantalla principal del
módulo de juegos.
/** Parámetros de entrada: String[] args.
/** Parámetros de salida: -
//*****

//*****
/** Nombre de la función: informacion
/** Explicación de lo que hace la función: Permite elegir de qué juego se quieren
leer las instrucciones.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****

//*****
/** Nombre de la función: libro
/** Explicación de lo que hace la función: Muestra un libro en la sección de
instrucciones.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****

//*****
/** Nombre de la función: instruccionesOca
/** Explicación de lo que hace la función: Explica el funcionamiento del juego de
la Oca.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****

//*****
/** Nombre de la función: instruccionesPistolero
/** Explicación de lo que hace la función: Explica el funcionamiento del juego
del pistolero.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****

//*****
/** Nombre de la función: instruccionesLucha
/** Explicación de lo que hace la función: Explica el funcionamiento del juego de
lucha.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****
```

```
//*****
/** Nombre de la función: jugar
/** Explicación de lo que hace la función: Menú de selección de juego.
/** Parámetros de entrada: -
/** Parámetros de salida: Variables de juego y vs.
//*****

//*****
/** Nombre de la función: seleccionRival
/** Explicación de lo que hace la función: Menú de selección de rival.
/** Parámetros de entrada: Variables juego y vs.
/** Parámetros de salida: Variables juego y vs.
//*****

//*****
/** Nombre de la función: dibujoOca
/** Explicación de lo que hace la función: Muestra el dibujo de la Oca en la
pantalla de selección de juego.
/** Parámetros de entrada: -
/** Parámetros de salida: String oca.
//*****

//*****
/** Nombre de la función: dibujoPistolero
/** Explicación de lo que hace la función: Muestra el dibujo del Pistolero en la
pantalla de selección de juego.
/** Parámetros de entrada: -
/** Parámetros de salida: String pistolero.
//*****

//*****
/** Nombre de la función: dibujoLucha
/** Explicación de lo que hace la función: Muestra el dibujo de la Lucha en la
pantalla de selección de juego.
/** Parámetros de entrada: -
/** Parámetros de salida: String lucha.
//*****

//*****
/** Nombre de la función: autores
/** Explicación de lo que hace la función: Muestra los autores del módulo de
minijuegos y permite volver a la pantalla anterior.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****

//*****
/** Nombre de la función: mano
/** Explicación de lo que hace la función: Muestra una mano en la sección de
autores.
/** Parámetros de entrada: -
/** Parámetros de salida: -
//*****
```

```
/**
*****
/** Nombre de la función: version
/** Explicación de lo que hace la función: Muestra la versión del módulo de
minijuegos y permite volver a la pantalla anterior.
/** Parámetros de entrada: -
/** Parámetros de salida: -
/**
*****

/**
*****
/** Nombre de la función: versionDibujo
/** Explicación de lo que hace la función: Muestra un joystick en la sección de
versión.
/** Parámetros de entrada: -
/** Parámetros de salida: -
/**
*****

/**
*****
/** Nombre de la función: salir
/** Explicación de lo que hace la función: Finaliza el programa.
/** Parámetros de entrada: -
/** Parámetros de salida: -
/**
*****
```

### Funciones juego 1:

```
/**
*****
/** Nombre de la función: oca
/** Explicación de lo que hace la función: Genera los tableros y todas las
variables del juego de la Oca.
/** Parámetros de entrada: Variables vs.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/**
*****

/**
*****
/** Nombre de la función: tiradaOcaJ1
/** Explicación de lo que hace la función: Hace los procesos necesarios para la
tirada del jugador 1.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/**
*****

/**
*****
/** Nombre de la función: comprobacionesTiradaOcaJ1
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1
tiene que rebotar al llegar al final del tablero.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/**
*****
```

```

/*****
/** Nombre de la función: tiradaOcaJ2
/** Explicación de lo que hace la función: Hace los procesos necesarios para la
tirada del jugador 2.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesTiradaOcaJ2
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 2
tiene que rebotar al llegar al final del tablero.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesOcaJ1
/** Explicación de lo que hace la función: Comprueba si en la casilla que ha
caído el jugador 1 hay una oca o si es el final del tablero.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ1Puente
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1 ha
caído en un puente.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ1Dado
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1 ha
caído en un dado.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

```

```

/*****
/** Nombre de la función: comprobacionesJ1Carcel
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1 ha
caído en la cárcel o si el jugador 2 está en ella.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ1Pozo
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1 ha
caído en el pozo o si el jugador 2 se encuentra en ella.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ1Posada
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1 ha
caído en la posada o si el jugador 2 se encuentra en ella.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ1Finales
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 1 ha
caído en el laberinto, en la calavera o en una casilla vacía.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesOcaJ2
/** Explicación de lo que hace la función: Comprueba si en la casilla que ha
caído el jugador 2 o la IA hay una oca o si es el final del tablero.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

```



```

/*****
/** Nombre de la función: comprobacionesJ2Puente
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 2 o
la IA ha caído en un puente.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ2Dado
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 2 o
la IA ha caído en un dado.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ2Carcel
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 2 o
la IA ha caído en la cárcel o si el jugador 1 está en ella.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ2Pozo
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 2 o
la IA ha caído en el pozo o si el jugador 1 se encuentra en ella.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

/*****
/** Nombre de la función: comprobacionesJ2Posada
/** Explicación de lo que hace la función: Comprueba si la ficha del jugador 2 o
la IA ha caído en la posada o si el jugador 1 se encuentra en ella.
/** Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
/** Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1,
tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
*****/

```

```
/**
 * Nombre de la función: comprobacionesJ2Finales
 * Explicación de lo que hace la función: Comprueba si la ficha del jugador 2 o la IA ha caído en el laberinto, en la calavera o en una casilla vacía.
 * Parámetros de entrada: Variables vs, Arrays de char tablero, tableroJ1, tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
 * Parámetros de salida: Variables vs, Arrays de char tablero, tableroJ1, tableroJ2 y variables J1, J2, pTJ1, pTJ2, tirada1, tirada2.
 */
```

## Funciones juego 2:

```
/**
 * Nombre de la función: pistolero
 * Explicación de lo que hace la función: Esta función inicializa las variables que necesitare en el programa
 * Parámetros de entrada:-
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 */
```

```
/**
 * Nombre de la función: turnoJ1
 * Explicación de lo que hace la función: Esta funcion le pide al jugador 1 que escoja lo que quiere hacer
 * y controla que los valores no sean erróneos.
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 * Parámetros de Salida: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 */
```

```
/**
 * Nombre de la función: IA
 * Explicación de lo que hace la función: Esta funcion le otorga a la variable ia un
 * valor random entre el 0 i el 2 mientras pasan los turnos
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 */
```

```
/**
 * Nombre de la función: turnoJ2
 * Explicación de lo que hace la función: Esta funcion le pide al jugador 2 que escoja lo que quiere hacer
 * y controla que los valores no sean erróneos. Depende del vs
 * que se escoja se hará random o se hará manualmente por el jugador 2.
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2, ia, vs
 */
```

```

/*****
/** Nombre de la función: comparacion
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando los dos
/** jugadores recargan
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: comparaciona
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con tres de munición o más y el jugador 2 recarga.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: comparacionb
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con tres de munición o más y el jugador 2 dispara con
/** menos de tres de munición.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: comparacionc
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con tres de munición o más y el jugador 2 se cubre
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

```

```
/**
*****
/** Nombre de la función: comparaciond
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con tres de munición o más y el jugador 2 dispara
/** también con tres o más de munición.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****

```

```
/**
*****
/** Nombre de la función: comparacion1
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con menos de tres de munición y el jugador 2 /** recarga.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****

```

```
/**
*****
/** Nombre de la función: comparacion2
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con menos de tres de munición y el jugador 2 dispara
/** también con menos de tres de munición.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****

```

```
/**
*****
/** Nombre de la función: comparacion3
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** dispara con menos de tres de munición y el jugador 2
/** se cubre.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****

```

```
/**
 * Nombre de la función: comparacionx
 * Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
 * cuando el jugador 2
 * dispara con tres de munición o más y el jugador 1
 * recarga.
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
 * el2, ia, vs
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
 * ia, vs
 */
```

```
/**
 * Nombre de la función: comparaciony
 * Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
 * cuando el jugador 2
 * dispara con tres de munición o más y el jugador 1
 * se cubre.
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
 * el2, ia, vs
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
 * ia, vs
 */
```

```
/**
 * Nombre de la función: comparacion4
 * Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
 * cuando el jugador 2
 * dispara con menos de tres de munición y el jugador 1 recarga.
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
 * el2, ia, vs
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
 * ia, vs
 */
```

```
/**
 * Nombre de la función: comparacion5
 * Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
 * cuando el jugador 2
 * dispara con tres de munición o más y el jugador 1 dispara con
 * menos de tres de munición.
 * Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
 * el2, ia, vs
 * Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
 * ia, vs
 */
```

```

/*****
/** Nombre de la función: comparacion6
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 2
/** dispara con menos de tres de munición y el jugador 1 se cubre.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: comparacion7
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** se cubre y el jugador 2 también se cubre.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: comparacion8
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 1
/** se cubre y el jugador 2 recarga.
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: comparacion9
/** Explicación de lo que hace la función: Esta funcion comprueba lo que pasa
cuando el jugador 2
/** se cubre y el jugador 1 recarga
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

/*****
/** Nombre de la función: ganador
/** Explicación de lo que hace la función: Esta funcion comprueba quien ha ganado
según las vidas que tienen
/** Parámetros de entrada: Variables municion1, municion2, vida1, vida2, el1,
el2, ia, vs
/** Parámetros de salida: Variables municion1, municion2, vida1, vida2, el1, el2,
ia, vs
*****/

```

### Funciones juego 3:

```
//*****
/** Nombre de la función: pelea
/** Explicación de lo que hace la función: muestra por consola los dibujos de los
personajes.
/** Parámetros de entrada: Variable vs.
/** Parámetros de salida: Variable vs.
//*****

//*****
/** Nombre de la función: jugador1
/** Explicación de lo que hace la función: da a elegir al jugador 1 el
combatiente que desea.
/** Parámetros de entrada: Variable vs.
/** Parámetros de salida: Variables combatiente1, combatiente2, vs.
//*****

//*****
/** Nombre de la función: jugador2
/** Explicación de lo que hace la función: da a elegir al jugador 2 el
combatiente que desea, en caso de ser contra la IA se hace aleatorio.
/** Parámetros de entrada: Variables comb1, comb2, vs.
/** Parámetros de salida: Variables vs, comb1, comb2.
//*****

//*****
/** Nombre de la función: creacionCaracteristicas
/** Explicación de lo que hace la función: crea los valores de las habilidades
que tiene el personaje escogido (en algunos casos es aleatorio)
/** Parámetros de entrada: Variables vs, combatiente1, combatiente2.
/** Parámetros de salida: Variables ataque1, escudo1, salud1, velocidad1,
ataque2, escudo2, salud2, velocidad2, vida1, vida2, i, vs.
//*****

//*****
/** Nombre de la función: caracteristicas
/** Explicación de lo que hace la función: muestra por consola los valores de las
habilidades que tiene el personaje escogido (en algunos casos es aleatorio)
/** Parámetros de entrada: Variables vs, combatiente1, combatiente2, ataque1,
escudo1, salud1, velocidad1, ataque2, escudo2, salud2, velocidad2, vida1, vida2,
i.
/** Parámetros de salida: Variables ataque1, escudo1, salud1, velocidad1,
ataque2, escudo2, salud2, velocidad2, vida1, vida2, i, vs.
//*****

//*****
/** Nombre de la función: turnos
/** Explicación de lo que hace la función: muestra por consola el inicio de la
ronda
/** Parámetros de entrada: Variables a1, e1, s1, v1, a2, e2, s2, v2, vida1,
vida2, i, vs.
/** Parámetros de salida: Variables eleccion1, eleccion2, a1, e1, s1, v1, a2, e2,
s2, v2, ataque, vida1, vida2, i, vs.
//*****
```

```

/*****
/** Nombre de la función: turno1
/** Explicación de lo que hace la función: muestra por consola lo que debe hacer
el jugador 1 para atacar o defender, la elección.
/** Parámetros de entrada: Variable eleccion1.
/** Parámetros de salida: -
*****/

/*****
/** Nombre de la función: turno2
/** Explicación de lo que hace la función: muestra por consola lo que debe hacer
el jugador 2 para atacar o defender, la elección (en el caso de la IA es
aleatorio)
/** Parámetros de entrada: Variable eleccion2, vs.
/** Parámetros de salida: -
*****/

/*****
/** Nombre de la función: decisiones
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables eleccion1, eleccion2, a1, e1, s1, v1, a2,
e2, s2, v2, ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables eleccion1, eleccion2, a1, e1, s1, v1, a2, e2,
s2, v2, ataque, vida1, vida2, i, vs.
*****/

/*****
/** Nombre de la función: jugador1
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
*****/

/*****
/** Nombre de la función: jugador1_1
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
*****/

```



```

/*****
/** Nombre de la función: jugador1_2
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
*****/

/*****
/** Nombre de la función: jugador1_3
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs
*****/

/*****
/** Nombre de la función: jugador2
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs
*****/

/*****
/** Nombre de la función: jugador2_1
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs
*****/

/*****
/** Nombre de la función: jugador2_2
/** Explicación de lo que hace la función: muestra por consola lo que sucede en
el campo de batalla según la elección de los 2 jugadores.
/** Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs.
/** Parámetros de salida: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
ataque, vida1, vida2, i, vs
*****/

```

```
/**
 * Nombre de la función: jugador2_3
 * Explicación de lo que hace la función: muestra por consola lo que sucede en
 * el campo de batalla según la elección de los 2 jugadores.
 * Parámetros de entrada: Variables el1, el2, a1, e1, s1, v1, a2, e2, s2, v2,
 * ataque, vida1, vida2, i, vs.
 * Parámetros de salida: Variables a1, e1, s1, v1, a2, e2, s2, v2, vida1, vida2,
 * i, vs.
 */
```

```
/**
 * Nombre de la función: ganador
 * Explicación de lo que hace la función: comprueba si se ha terminado el juego,
 * en caso contrario, vuelve a la función turnos del inicio
 * Parámetros de entrada: Variables a1, e1, s1, v1, a2, e2, s2, v2, vida1,
 * vida2, i, vs.
 * Parámetros de salida: Variables a1, e1, s1, v1, a2, e2, s2, v2, vida1, vida2,
 * i, vs.
 */
```

```
/**
 * Nombre de la función: personajeManolo/Jacobo/George/Toni/Tere
 * Explicación de lo que hace la función: Muestran los dibujos en el juego de
 * lucha.
 * Parámetros de entrada: -
 * Parámetros de salida: -
 */
```

### Funciones final de juego:

```
/**
 * Nombre de la función: victoria
 * Explicación de lo que hace la función: Muestra una Copa cuando hay ganador en
 * cualquiera de los juegos.
 * Parámetros de entrada: Variable vs.
 * Parámetros de salida: Variable vs.
 */
```

```
/**
 * Nombre de la función: finJuego
 * Explicación de lo que hace la función: Pregunta si se quiere volver a jugar
 * y, en caso afirmativo, lleva a la pantalla de selección de juego.
 * Parámetros de entrada: Variable vs.
 * Parámetros de salida: -
 */
```