

## Prácticas Introducción a Lua

### Práctica 1

Usando el esqueleto de la práctica. En el esqueleto sin ninguna modificación aparece una criatura en pantalla.

El primer paso es hacer igual que en la práctica de operaciones y hacer que se mueva por la pantalla hacia la derecha.

Después, el objetivo de esta práctica es que al pulsar el botón izquierdo del ratón la criatura pase a moverse en la dirección contraria, es decir, si se movía hacia la derecha pasará a moverse hacia la izquierda y si se movía hacia la izquierda pasará a moverse hacia la derecha.

Para ello contamos con la posición del ratón actualizada en las variables globales `mousePositionX` y `mousePositionY`, y un fragmento de código que se ejecutará siempre que se pulse el botón izquierdo del ratón. Dentro de este código existen variables que almacenan la posición de la criatura: `creaturePositionX` y `creaturePositionY`, y variables que almacenan el tamaño de la criatura: `creatureSizeX`, `creatureSizeY`. La criatura la consideramos que es un rectángulo de esas dimensiones, no vamos a preocuparnos de la forma real de la imagen. Para resolver la práctica tienes que escribir código en el fragmento que se encuentra entre los comentarios -- Escribe tu código aquí para botón izquierdo ratón y -- Termina tu código.

### Práctica 2

El objetivo de esta práctica es hacer lo mismo que la práctica anterior pero haciendo que pase por todas las direcciones cardinales: de derecha a izquierda, de izquierda a arriba, de arriba a abajo y de abajo a la derecha.

### Práctica 3

El objetivo de esta práctica es utilizar el teclado para mostrar cartas de una baraja en la pantalla.

Si ejecutáis el esqueleto de la práctica veréis que se pinta un as de corazones. Con la práctica vamos a cambiar esa carta pulsando teclas en el teclado.

Contamos con las siguientes cartas: 2, 3, 4, 5, 6, 7, 8, 9, J, Q y K para cuatro palos distintos de la baraja de póker: corazones, tréboles, diamantes y picas.

Las combinaciones de teclas que vamos a usar para seleccionar qué carta se quiere ver en pantalla es una letra c, d, t o p para seleccionar el palo (corazones, diamantes, tréboles y picas respectivamente) y sin soltar esta tecla pulsar un número del 2 al 9 o las teclas j, q o k para seleccionar que carta queremos.

Dentro de la práctica hay una carpeta llamada `cards` que contiene una imagen para cada carta con una nomenclatura: `2_C.png` para el 2 de corazones, `A_P.png` para el As de picas y así sucesivamente.

Dentro del esqueleto hay un fragmento de código donde hace falta añadir cosas para que la práctica funcione. Está marcado entre los comentarios habituales.

Para completar la práctica se tienen disponibles unas variables que pueden ser de ayuda:

- `key_pressed`: contiene un valor de tipo cadena con el texto de la tecla que se ha pulsado. Por ejemplo si se ha pulsado la tecla 1, la variable contendrá el valor "1".
- `image_file`: tiene que contener el nombre del fichero de la imagen que se quiere mostrar por pantalla para la carta. Por ejemplo si queremos mostrar el rey de picas, `image_file` debería contener `cards\K_P.png`, ya que las imágenes están en un directorio llamado `cards`. Si no rellenáis `image_file` con ningún valor no se cambia la carta que se esté mostrando en ese momento.
- `down`: indica si lo que ha ocurrido es que se ha pulsado la tecla (`down` será igual a `true`) o se ha soltado la tecla (`down` será igual a `false`). Esto significa que el código entre comentarios se ejecuta una vez cuando pulsamos la tecla y otra vez cuando la soltamos.

Ahora extiéndelo para que no importe si está la tecla de mayúsculas pulsada.

## Práctica 4

El objetivo de esta práctica es implementar la base de un pequeño juego muy sencillo. El juego consiste en que el jugador con el botón izquierdo coloca criaturas en la pantalla en la posición donde ha hecho click. Estas criaturas nacen con 25 puntos de vida. Con el botón derecho si hace click encima de una criatura se le quitan 5 puntos de vida. Cuando una criatura llega a 0 de vida se elimina de la pantalla.

Para ello usarás el esqueleto proporcionado. Puedes crear todas las variables que consideres oportuno para gestionar la información.

El código que tienes que escribir estará entre los comentarios apropiados para el botón izquierdo y para el botón derecho.

Como siempre tienes las variables `mousePositionX` y `mousePositionY` que indican la posición del puntero del ratón en todo momento.

Además dentro del fragmento de código para el botón derecho tienes disponible el tamaño de la criatura en las variables: `creatureSizeX` y `creatureSizeY`.

Para añadir una criatura a la pantalla se tiene la función:

```
addCreature(nombre, posicionX, posicionY)
```

donde `nombre` es el nombre de la criatura, que en este caso será "griphon", `posicionX` y `posicionY` es un número que indica la posición donde se pintará la criatura.

La función devuelve un valor que almacena la criatura, este es el valor que hay que usar posteriormente para eliminar la criatura de la pantalla.

Para eliminar una criatura de la pantalla se tiene la función:

```
removeCreature(criatura)
```

a la que hay que pasarle el valor de la criatura que nos devolvió la función addCreature cuando la añadimos.

Date cuenta que sólo con añadir que las criaturas se muevan por la pantalla o que aparezcan y desaparezcan tenemos un juego muy sencillo pero perfectamente funcional:



## Opcional

### Práctica 5

El objetivo de esta práctica es hacer lo mismo que en la práctica 2, pero en lugar de hacer click encima de la criatura, tiene que cambiar de dirección cada vez que el ratón pasa a estar dentro de la criatura.

Si tenemos el ratón que no está encima de la criatura, ésta se está moviendo hacia una dirección. Cuando paso a poner el ratón encima de la criatura, cambia de dirección. Sigue en esa dirección mientras tenga el ratón encima de la criatura. Si saco el ratón de encima de la criatura, ésta sigue en la misma dirección. Si vuelvo a poner el ratón encima de la criatura entonces ésta cambia de dirección.

### Práctica 6

Partiendo de lo que hiciste en la práctica 2, la idea es convertirlo en un juego muy sencillo. Para ello tienes que usar el esqueleto de esta práctica que tiene algunos componentes adicionales.

Para ello vamos a incrementar la velocidad a la que se mueve la criatura a 350 píxeles por segundo.

Vamos a cambiar el cambio de direcciones, que ahora es predecible, a que sea aleatorio. Para obtener un número aleatorio entre dos valores low y high podemos escribir esta línea:  
resultado = math.random(low, high), low y high pueden ser variables o valores.

Aún nos falta un objetivo en el juego, el objetivo es que la criatura no salga de la pantalla. Para controlarlo hay que añadir código que ponga la variable gameOver a true cuando el cuadrado que engloba a la criatura (definido por los valores de las variables creatureSizeX y creatureSizeY) salga de la pantalla. La pantalla está definida entre 0 y 1024 en coordenadas X, y entre 0 y 768 en coordenadas Y.

## Práctica 7

En esta práctica vas a hacer un sencillo juego de cartas. El juego presenta unas cuantas cartas en la pantalla en una cuadrícula y elige un palo de la baraja. El jugador debe pulsar encima de todas las cartas que correspondan a ese palo o pulsar fuera de la zona de las cartas si no hay ninguna de ese palo. Al pulsar encima de todas las cartas del palo elegido el jugador gana el juego.

Para ello hay que usar el esqueleto de la práctica.

En él se definen algunas variables que configuran el juego:

- `number_cards_X`: Número de cartas a pintar en la dimensión horizontal
- `number_cards_Y`: Número de cartas a pintar en la dimensión vertical. Por ejemplo si `number_cards_X` vale 4 y `number_cards_Y` vale 4 hay que pintar 16 cartas.
- `card_sizeX`, `card_sizeY`: Es el tamaño de una carta en la pantalla en
- `card_init_X`, `card_init_Y`: Coordenada del pixel a partir del cual se pintan las cartas.
- `mousePositionX`, `mousePositionY`: Como siempre guardan las coordenadas del ratón.

Se tienen también las siguientes funciones:

- `addImage(fichero_imagen, posicionX, positionY)`: Para pintar las cartas pasándole el fichero de imagen de la carta y la posición donde hay que pintarla. Devuelve un valor que se puede usar posteriormente para borrar la imagen
- `removeImage(imagen)`: Para borrar una imagen de la pantalla. Hay que pasarle un valor que haya sido devuelto por `addImage`
- `lost()`: Llamar cuando se haya perdido el juego
- `won()`: Llamar cuando se haya ganado el juego
- `math.floor(numero)`: devuelve la parte entera del número decimal pasado como parámetro
- `math.ceil(numero)`: devuelve el entero redondeado hacia arriba del número decimal pasado como parámetro
- `math.random(low, high)`: devuelve un número entero comprendido entre low y high, ambos inclusive

La mecánica es simple.

Pulsando el botón izquierdo se resetea el juego. Se eligen aleatoriamente las cartas a mostrar en la pantalla y se muestran en forma de matriz una carta junto a la otra, sin dejar huecos entre ellas. Se elige un palo aleatoriamente y se muestra el as de ese palo en la esquina superior derecha (teniendo en cuenta que la pantalla mide 1024x768).

Pulsando el botón derecho se elige la carta sobre la que se ha pulsado. Si el palo de la carta no coincide con el palo elegido o se ha pulsado fuera de las cartas, pero hay cartas del palo elegido en la pantalla se ha perdido el juego (hay que llamar a la función `lost`). Cuando se hayan elegido todas las cartas pertenecientes al palo seleccionado se ha ganado el juego (hay que llamar a la función `won`).