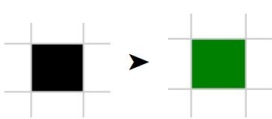


Práctica 3.4 - Alternativa Condicional

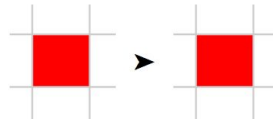
Nota general para todos los ejercicios de esta guía: Cualquier programa o procedimiento que implemente debe estar adecuadamente documentado mediante comentarios. Esto incluye pero no se limita a propósitos y precondiciones. Debe incluir cualquier comentario que considere adecuado para aumentar la calidad expresiva de su código.

Ejercicio N° 1

Implementar utilizando QDraw, un procedimiento que pinte de color Verde la celda actual solo si está pintada de color Negro.



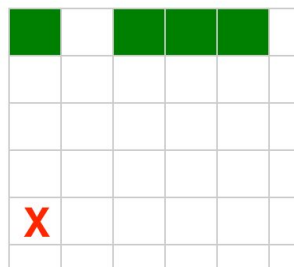
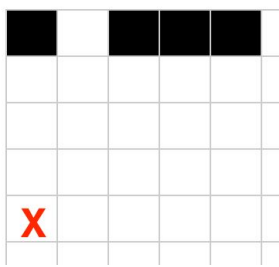
Ejemplo Caso 1



Ejemplo Caso 2

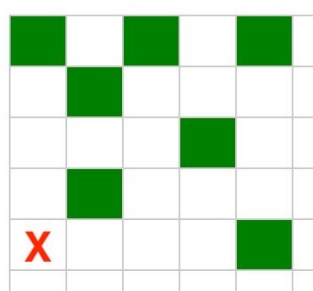
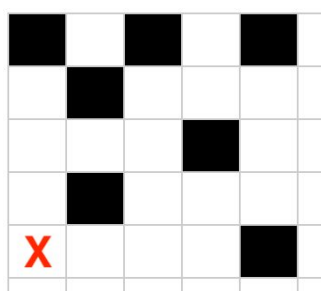
Ejercicio N° 2

Implementar utilizando QDraw, un procedimiento que pinte de color Verde cada celda de la última fila superior de un tablero de 5x5, solo si dicha celda está pintada de Negro. El cabezal comienza en la esquina inferior izquierda del tablero y termina en el mismo lugar. ¿Podría reutilizar todo o parte del procedimiento del ejercicio 1?



Ejercicio N° 3

Implementar utilizando QDraw, un procedimiento que pinte de color Verde todas las celdas de un tablero de 5x5, solo si cada celda está pintada de Negro. El cabezal comienza en la esquina inferior izquierda del Tablero y termina en el mismo lugar. ¿Podría reutilizar todo o parte del procedimiento del ejercicio 2?

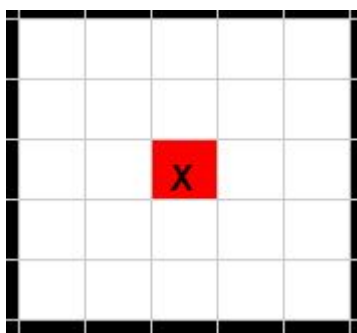


Práctica 3.4 - Alternativa Condicional

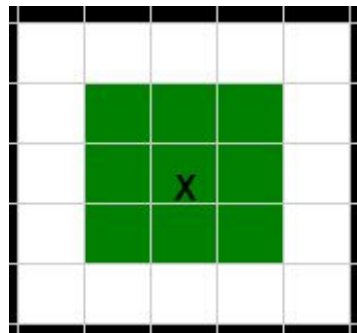
Ejercicio N° 4

Implementar utilizando QDraw, un procedimiento que dibuje un cuadrado o una cruz dependiendo del color de la celda actual. En caso de que dicha celda esté coloreada de Rojo debe dibujar un Cuadrado de 3x3 Verde centrado en la celda inicial, caso contrario debe dibujar una Cruz de 3x3 Verde también centrada en la celda inicial. Se cuenta con un tablero de 5x5 con el cabezal ubicado en la celda central (fila 2 y columna 2).

A continuación un ejemplo



Tablero Inicial

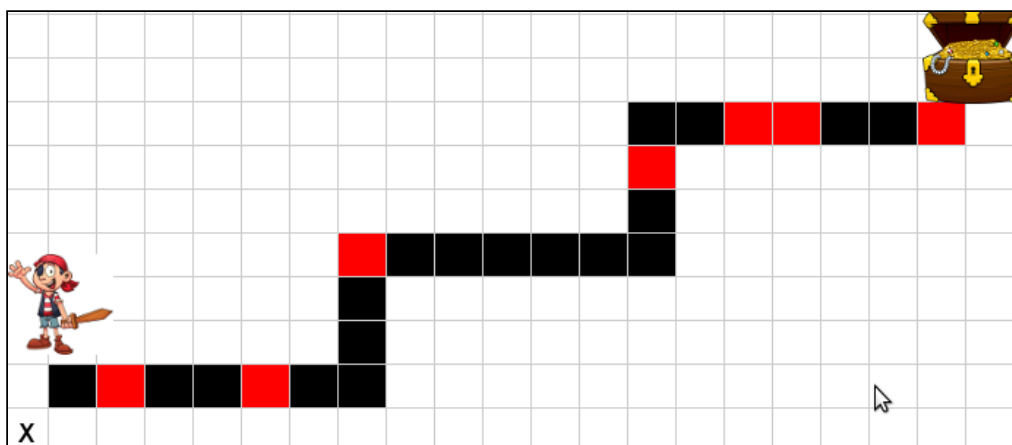


Tablero final

Ejercicio N° 5

El pirata debe recorrer el camino marcado, teniendo en cuenta que cada celda roja es una muralla que le imposibilita avanzar. Dado que su misión es llegar hasta al cofre del tesoro debe derribar el obstáculo, ésto es pintar la celda que tiene la muralla de negro.

Implementar utilizando QDraw, el procedimiento **EncontrarTesoro()**, sabiendo que contamos con un tablero de 10 x 21 y el cabezal comienza en la esquina inferior izquierda del Tablero.



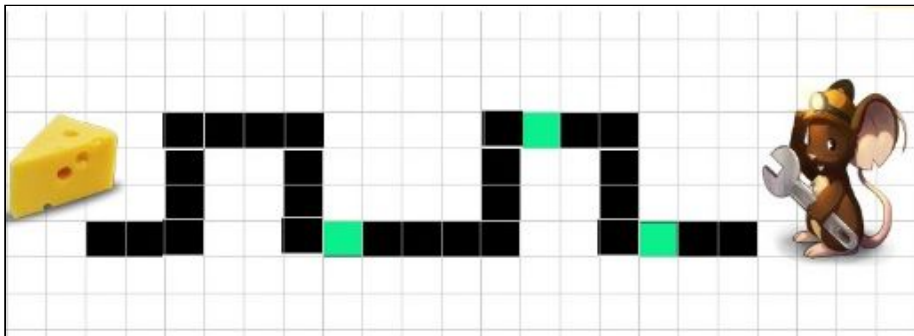
Práctica 3.4 - Alternativa Condicional

Ejercicio N° 6

El pequeño ratoncito Perez, desea comer un delicioso queso que se encuentra al final del camino, por lo que deberá seguir la ruta marcada en la siguiente figura. Lamentablemente, el ratoncito descubre que hay trampas, las cuales debe desarmar.

Teniendo en cuenta que cada celda Verde indica una trampa que lo puede atrapar, debe desactivarlas, ésto es pintar la celda que tiene la trampa de negro.

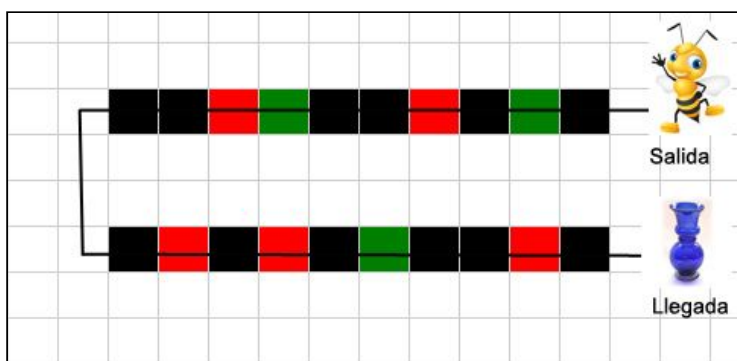
Implementar utilizando QDraw, el procedimiento **DesactivarTrampas()**, sabiendo que contamos con un tablero de 9 x 23 y el cabezal comienza en la esquina inferior derecha del Tablero. Tenga en cuenta que el camino mostrado es un ejemplo, y que la distribución de trampas dentro del camino podría ser distinta, mientras que la forma del camino siempre es la misma.



Ejercicio N° 7

La abeja Susi poliniza flores de su jardín. Para ello siempre sigue estas reglas:

1. Rojo, indica que la flor está lista para ser polinizada.
2. Verde, indica que la flor aún no está madura pero aun así, está lista para ser polinizada.
3. Negra, indica que no hay flores.



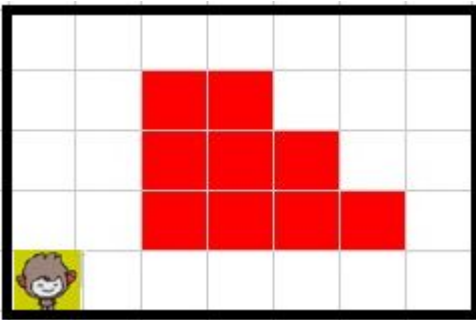
Susi debe polinizar todas las flores posibles del jardín. Para polinizar una flor, debe despintar la celda donde la encuentre.

Ayuda a Susi a polinizar todas las flores, teniendo en cuenta que el camino a recorrer es el de la figura, pero no sabe donde hay flores, es decir la distribución de flores puede diferir del ejemplo

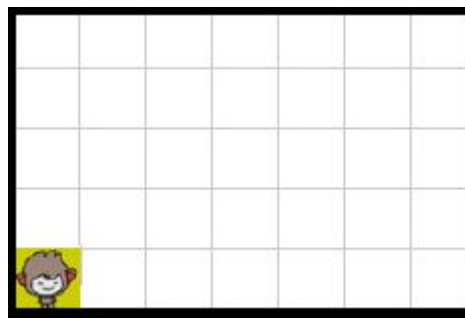
Práctica 3.4 - Alternativa Condicional

Ejercicio N° 8

La casa donde vive el mono Jorgito es de 5 por 7 celdas. La madre trabaja todo el día y le dejó las suficientes bananas para que pueda alimentarse. Sabemos de antemano que las bananas están representadas por celdas rojas y **pueden estar distribuidas por toda la casa** (no necesariamente como en el ejemplo). Jorgito necesita comer todas las bananas del tablero para quedar satisfecho. Para ello se requiere implementar un procedimiento **MonoJorgitoComeBananas()**. El cabezal comienza en la esquina inferior izquierda del Tablero.



Tablero Inicial

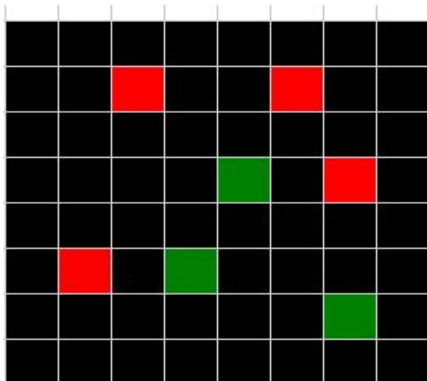


Tablero Final

Ejercicio N° 9

Una empresa dedicada a realizar videojuegos, comienza a desarrollar la batalla naval, para ello define un tablero de 8 x 8 con reglas específicas:

1. Cada celda Negra representa agua del océano.
2. Cada celda Verde representa un Barco Chico
3. Cada celda Roja representa un Barco Grande.



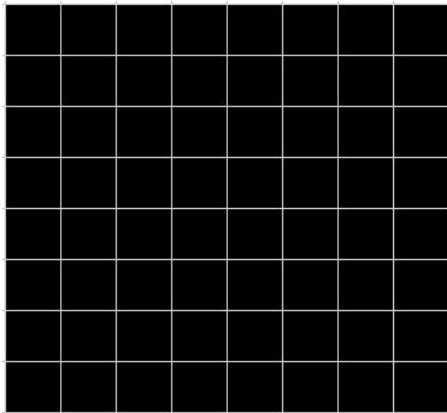
La siguiente figura es un ejemplo, donde vemos una porción de océano de 8 x 8 con 4 barcos grandes y 3 barcos chicos.

Tablero A

Práctica 3.4 - Alternativa Condicional

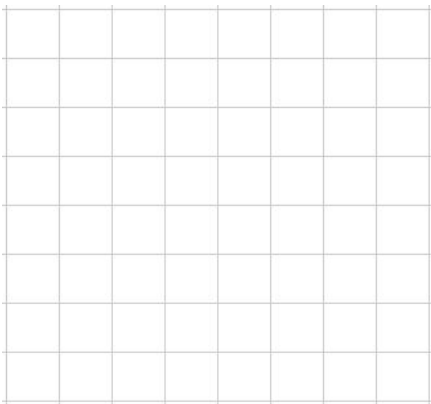
Se pide implementar en QDraw los siguientes procedimientos:

1. **EliminarBarcosDelOceano()**, que retira del juego tanto barcos chicos como grandes, colocando en su lugar agua.

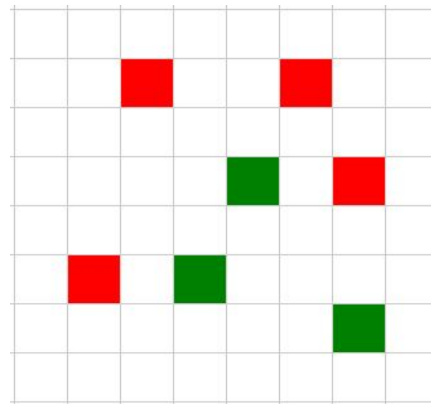


Tablero final B

2. **MarcarOceano()**, que despinta todas las celdas que representan agua.



Tablero final (a partir de tablero final B)



Tablero final (a partir de tablero A)

Ejercicio N° 10.1

REX el temerario combatiente de Fornite quien “lleva la competencia hacia la extinción”, necesita llegar hasta la otra punta de un túnel y poder recobrar un “Pico” (arma) que ha perdido en medio de una batalla.

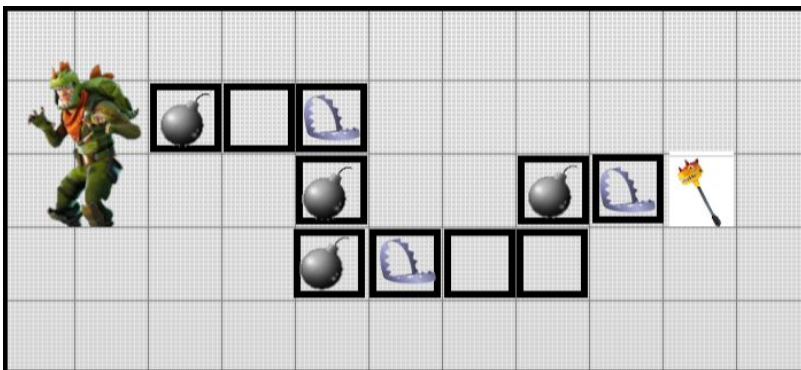
El túnel **tiene la forma que se representa en la imagen** y REX encontrará bombas y trampas durante su recorrido, que debe desactivar, tanto las bombas como las trampas pueden estar distribuidas en el túnel en cualquier posición.

Desactivar tanto bombas como trampas implica marcar la celda como correcta.

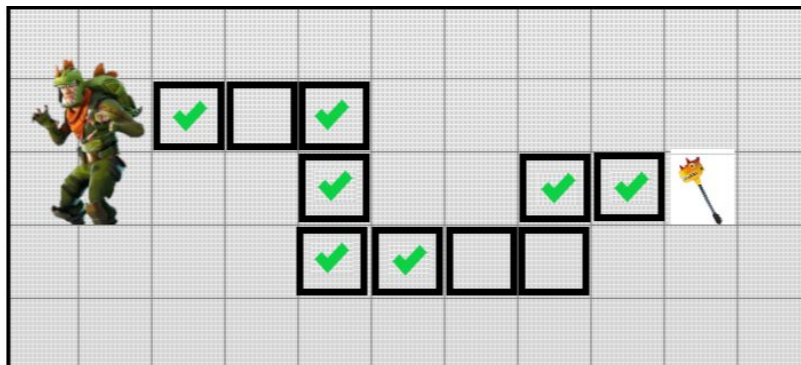
Utilizando las siguientes primitivas implemente el procedimiento **RecuperarArma()**, que ayude a el combatiente a llegar y recuperar su arma :

Práctica 3.4 - Alternativa Condicional

1. **IrAlComienzo**: coloca a REX al comienzo del camino
2. **MoverRex**, Mueve a Rex un paso adelante. Debe haber camino.
3. **GirarRex**, Gira el personaje hacia
4. **hayBomba?**, denota VERDADERO cuando hay Bomba, FALSO caso contrario
5. **hayTrampa?**, denota VERDADERO cuando hay Trampa, FALSO caso contrario
6. **TomarArma**: Encuentra el arma y la toma.
7. **DesactivarBomba**: Desactiva la bomba marcando la celda como correcta. Debe haber bomba.
8. **DesactivarTrampa**: Desactiva la trampa marcando la celda como correcta. Debe haber trampa.



Antes de ejecutar **RecuperarArma()**



Después de ejecutar **RecuperarArma()**

Ejercicio N° 10.2

Suponga ahora que no conoce la forma del túnel, solamente sabe que hay exactamente 10 pasos hasta llegar a encontrar su arma.

Se incorpora las siguientes instrucciones:

hayCaminoEnFrente?, que Denota Verdadero si hay una porción de camino al que pueda avanzar.



Práctica 3.4 - Alternativa Condicional

PosicionarHaciaCamino, que ubica a Rex justo en la posición correcta para poder avanzar.