

CLASE 17:



UNIDAD 2- PROGRAMACIÓN

ALTERNATIVA CONDICIONAL



Universidad
Nacional
de Quilmes



Definición

La alternativa condicional permite elegir en base a una condición si un bloque de código debe ejecutarse o no.



Reemplazamos con
la primitiva
correspondiente

si (CONDICIÓN) entonces{

BLOQUE SI LA CONDICION SE CUMPLE

} sino{

BLOQUE SI LA CONDICION NO SE CUMPLE

}



- ✗ **estaPintadoDeNegro?** Denota Verdadero si la celda está pintada de color negro, falso en caso contrario.
- ✗ **estaPintadoDeRojo?** Denota Verdadero si la celda está pintada de color rojo, falso en caso contrario.
- ✗ **estaPintadoDeVerde?** Denota Verdadero si la celda está pintada de color verde, falso en caso contrario.
- ✗ **estaVacía?** Denota Verdadero si la celda está vacía, falso en caso contrario.



También se utilizan algunos operadores vistos en lógica

Negación (\neg)

p	$\neg p$
V	F
F	V

Conjunción (\wedge)

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Disyunción (\vee)

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F



Por ejemplo:

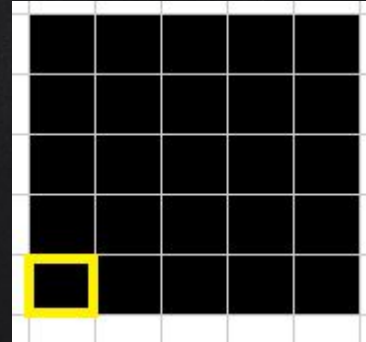
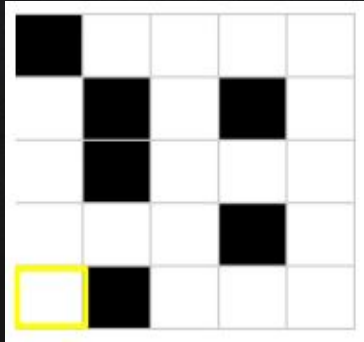
```
si (estaPintadaDeNegro? v estaPintadaDeRojo?) entonces{  
    PintarVerde  
}
```




Al igual que la repetición simple,
en alternativa condicional no se
puede anidar. Lo solucionamos
con procedimientos



Ahora tenemos el siguiente problema. Tenemos una cuadrícula con algunas celdas pintadas de negro y otras vacías. Pero no sabemos dónde están ubicadas. Por lo que voy a tener que preguntar en cada celda su color.




```

Procedimiento DibujarCuadrado(){
    /*      */
    repetir 4 veces{
        DibujarColumna()
        MoverDerecha
    }
    DibujarColumna()
    VolverInicio()
}

```

```

procedimiento DibujarColumna(){
    /*      */
    repetir 4 veces{
        PintarCeldasNegro()
        MoverArriba
    }
    PintarCeldaNegro()
    IrProximaFila()
}

```

```

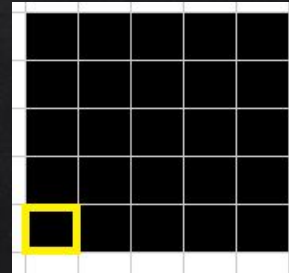
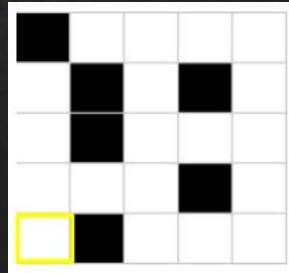
procedimiento PintarCeldasNegro(){
    /*      */
    si (estaVacía?) entonces{
        PintarNegro
    }
}

```

```

procedimiento IrProximaFila(){
    /*      */
    repetir 4 veces{
        MoverAbajo
    }
}

```



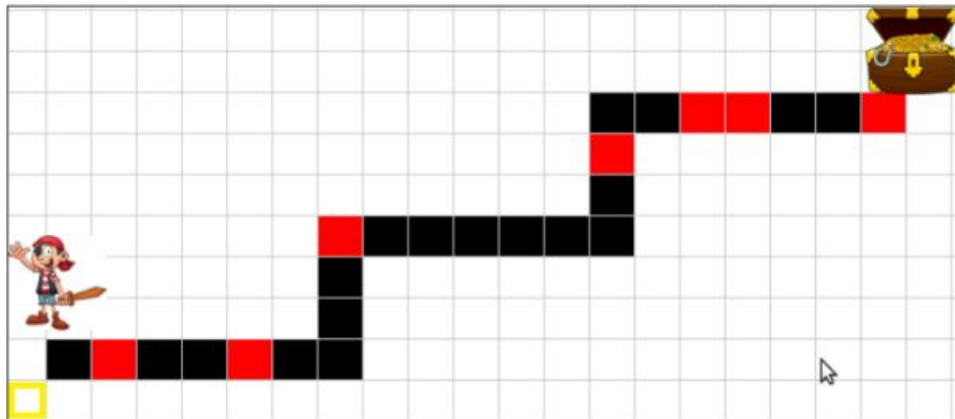
TENEMOS EJERCICIOS DE QDRAW+CONTEXTO.

AQUÍ SABEMOS EL CAMINO

Ejercicio 3

El pirata debe recorrer el camino marcado hasta llegar al tesoro. Puede suceder, que a medida avanza, se tope con una muralla (representada por una celda roja). En tal caso, deberá derribarla, pintando dicha celda de color Negro, y continuar con su camino. Utilizando QDraw, definir el procedimiento **EncontrarTesoro()** que resuelva la situación mencionada. Sabemos que el camino tiene una forma fija. El pirata se encuentra ubicado un paso previo al inicio del camino.

A continuación se muestra la forma que posee el camino, y un ejemplo de distribución de las murallas (no necesariamente están ubicadas en esas posiciones, es sólo un ejemplo)



TENEMOS EJERCICIOS DE CONTEXTO

Ejercicio 7

REX el temerario combatiente de Fornite quien "lleva la competencia hacia la extinción", necesita llegar hasta la otra punta de un túnel para 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 deberá desactivar para continuar avanzado. 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** define el procedimiento **RecuperarArma()**, que ayude al combatiente REX a recuperar su arma. Una vez que la recupere sacar a Rex del túnel:

1. **IrAlComienzo**: coloca a REX al comienzo del camino (ya se encuentra en la primer celda a evaluar)
2. **MoverRex**: mueve a REX un paso hacia adelante. Debe haber camino
3. **GirarRex**: gira a REX 90° en sentido de las agujas del reloj.
4. **TomarArma**: encuentra el arma y la toma.
5. **DesactivarBomba**: desactiva la bomba marcando la celda como correcta. Debe haber bomba.
6. **DesactivarTrampa**: desactiva la trampa marcando la celda como correcta. Debe haber trampa.
7. **hayBomba?**: denota VERDADERO cuando hay Bomba, FALSO caso contrario
8. **hayTrampa?**: denota VERDADERO cuando hay Trampa, FALSO caso contrario



Antes de ejecutarse RecuperarArma()



Después de ejecutarse RecuperarArma()

TENEMOS EJERCICIOS DONDE NO SABEMOS EL CAMINO Y TENEMOS QUE PREGUNTAR

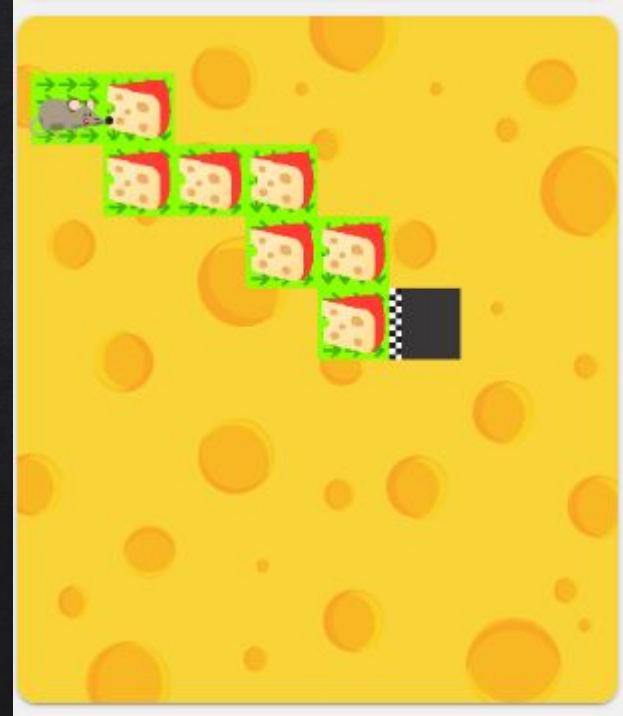
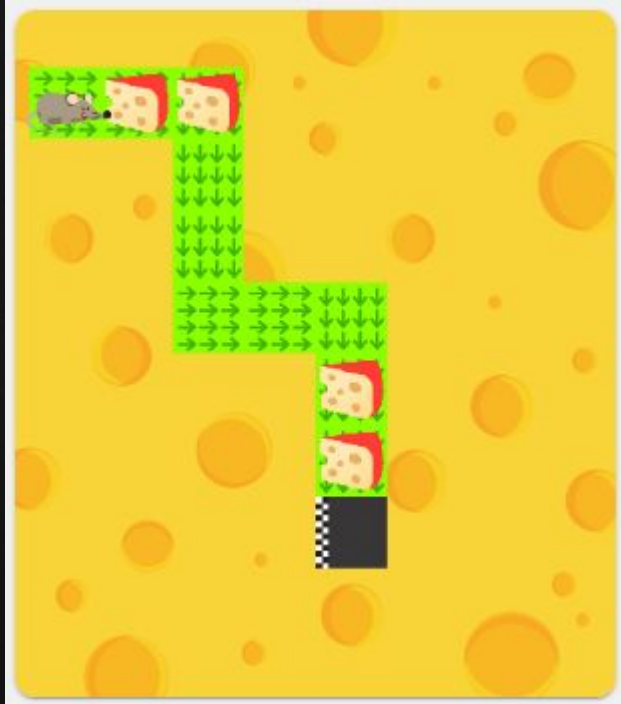
Ejercicio 7.2

Suponga ahora que no conoce la forma del túnel, pero que por su disposición solo podemos movernos hacia abajo y/o hacia la derecha, y que solamente hay que dar exactamente 10 pasos hasta llegar a encontrar su arma. Teniendo en cuenta parte de la solución que planteo en el punto 7.1, vea que podría reutilizar o modificar para que funcione, y defina la **procedimiento RecuperarArmaEn10Pasos()**

Se incorpora las siguientes instrucciones:

-
- **PosicionarHaciaCamino:** ubica a Rex justo en la posición correcta para poder avanzar
 - **hayCaminoHaciaAbajo?:** denota Verdadero si hay una porción del camino hacia abajo en la cual pueda avanzar.
 - **Hay caminoHaciaLaDerecha?**denota Verdadero si hay una porción del camino hacia la derecha en la cual pueda avanzar.

ACÁ VARÍA LA POSICIÓN DE QUESO Y EL CAMINO, PERO SABEMOS
SU LONGITUD



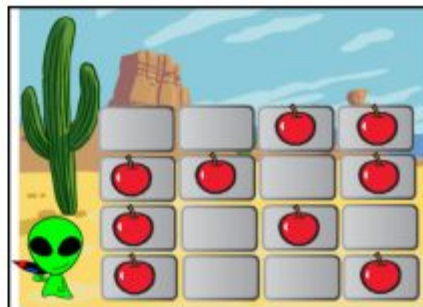
Ejercicio N° 8: El marciano en el desierto

El marciano en el desierto es un juego que consiste en comer todas las manzanas que el marciano encuentra en su paso por el camino.

Como primera versión del juego contamos con un desierto pequeño de 4 x 4 mts.

Para resolver el juego, se incorporaron al set de primitivas de QDraw las siguientes instrucciones y condiciones:

- **Comenzar:** ubica el marciano al comienzo del recorrido
- **ComerManzana:** toma la manzana que encuentra en el paso actual. Debe haber una manzana en dicho lugar.
- **hayManzana?:** denota VERDADERO si en el paso actual hay una manzana. FALSO en caso contrario.



Se requiere definir los siguientes procedimientos:

1. **ComerManzanasPorFila():** va recorriendo la fila y toma y come la manzana del paso actual, sólo en caso que haya una manzana en dicho paso.
2. **RecorrerDesierto():** recorre todo el desierto comiendo las manzanas que encuentra en el camino para no perder energía.

Recordar que la figura sólo es un ejemplo representativo del juego.