

Práctica 3.6 - Ejercicios Integradores para Parcial

Ejercicio N° 1: Tito el Robotito

Tito el robotito, es un robot programable que sirve para prender luces.

Se incorporaron al set de instrucciones de QDraw las siguientes instrucciones y condiciones:

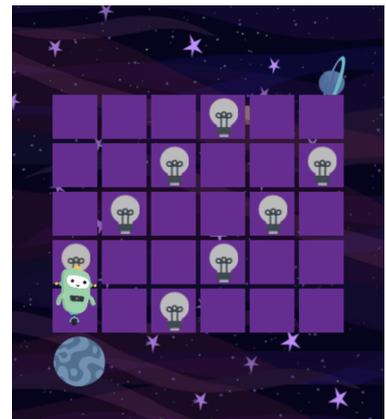
- **PrenderLuz**, que prende la luz del casillero actual. Debe haber una lámpara apagada en la celda actual.
- **ApagarLuz**, que apaga la luz del casillero actual. Debe haber una lámpara encendida en la celda actual.
- **hayUnaLamparitaAcá?**. Denota Verdadero si en la celda actual hay una lámpara. Falso en caso contrario
- **laLamparitaEstáPrendida?** Denota Verdadero si la lámpara de la celda actual está prendida. Falso en caso contrario

Se pide que resuelva los siguientes ejercicios:

Tito prende las luces

Programá a Tito para que encienda todas las luces de la habitación. Las luces siempre están en la misma ubicación, y todas están apagadas. Tito arranca siempre en el mismo lugar para prender las luces.

Comienza en la esquina inferior izquierda del tablero.



Tito y el gran pasillo

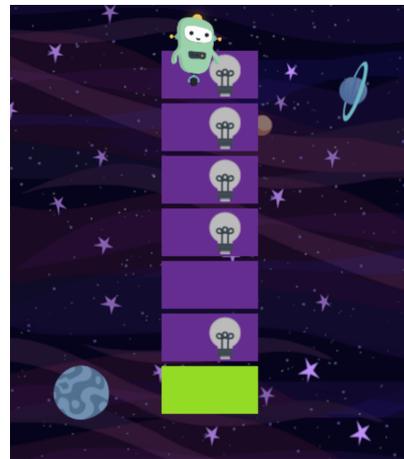
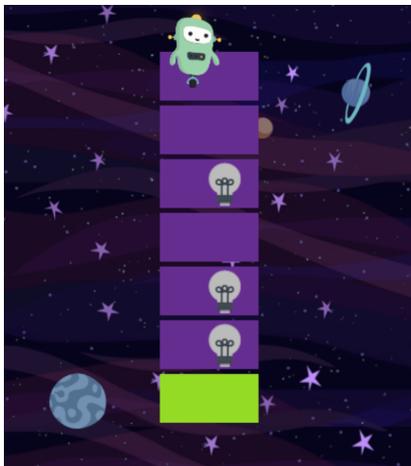
Ahora tito debe realizar la misma tarea, pero el pasillo es más largo. ¿Cómo ajustamos la estrategia para solucionar el problema?



Práctica 3.6 - Ejercicios Integradores para Parcial

Tito Recargado

Ahora Tito debe prender todas las luces de la columna, pero la gracia es que nunca sabemos donde van a estar las luces inicialmente. Si sabemos que la columna siempre tiene 7 celdas, que Tito arranca arriba del todo, y que las lamparitas pueden estar en las primeras 6 (las violetas). La última celda, la verde, es donde debe quedar parado Tito tras prender todas las luces.



Tito al cuadrado

Ahora Tito debe prender todas las luces del cuadrado. De la misma forma que antes, nunca sabemos en qué lugar del cuadrado van a aparecer las luces. Tito siempre parte de la esquina superior izquierda del cuadrado.

Tener en cuenta que las lamparitas comienzan todas apagadas.

Tito al cubo



Ahora Tito debe resolver un problema similar, pero ligeramente distinto. Ya no quiere prender todas las luces, sino solo aquellas que están inicialmente apagadas. En el cuadrado ahora, hay luces que ya arrancan encendidas.

Ejercicio N° 2

Práctica 3.6 - Ejercicios Integradores para Parcial

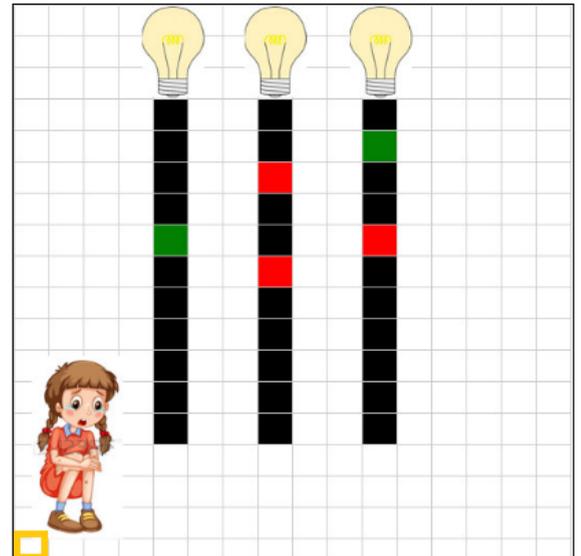
Ana está asustada porque se apagaron todas las luces de su habitación. Para resolver el problema debe encender las 3 lámparas que se encuentran distribuidas por su cuarto.

Como esta todo oscuro decide utilizar su pequeña linterna para seguir el mapa, tal cual muestra la siguiente figura.

En el camino se encuentra con casilleros marcados con colores y una nota donde explica lo siguiente:

1. Casillero rojo: Hay un juguete en el suelo entonces mueve el juguete a la izquierda, para liberar el camino y seguir hacia la luz. Es decir debe pintar la celda actual de Negro y la celda lindante a la izquierda de Rojo

2. Casillero verde: Hay lápices en el suelo, entonces mueve los lápices hacia la derecha. Es decir debe pintar la celda actual de Negro y la celda lindante a la derecha de Verde



Implementar utilizando QDraw, todos los procedimientos necesarios para encender las 3 luces de la habitación de Ana.

Para ayudarte en la resolución te dejamos usar las siguientes nuevas instrucciones;

- esLampara?: Denota Verdadero si en el casillero actual hay una lámpara
- Encender : Enciende la luz que se encuentra en dicho casillero. Debe haber una lámpara en la celda actual

Sabemos que la habitación es un tablero de 16 x 18 y el cabezal comienza en la esquina inferior izquierda del Tablero.

Ejercicio N° 3

Juan es contratado por el municipio de Berazategui para limpiar la plaza "San Martín", que luego de un recital quedó llena de papeles. Juan debe levantarlos y depositarlos en el contenedor de residuos.

Nuestra tarea como desarrolladores es programar un robot para que realice el trabajo de Juan. El robot arrastra los papeles hacia el cesto de basura.

Práctica 3.6 - Ejercicios Integradores para Parcial

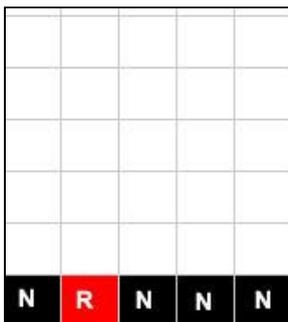
A continuación vemos un mapa de 5 x 6 de la plaza, con papeles en el suelo:



Donde :

- Los casilleros rojos representan el contenedor de residuos, ubicado a los largo de toda la plaza. El mismo se encuentra vacío.
- Los casilleros negros representan los papeles

Luego de limpiar la plaza el mapa es el siguiente:



Implementar el procedimiento **LimpiarPlaza()** teniendo en cuenta que:

- Los papeles se encuentran dispersos en cualquier sección de la plaza
- Siempre debemos comenzar limpiar la plaza desde la esquina superior izquierda

Ejercicio N° 4

Se quiere generar el código de programación necesario para realizar la afinación de un piano. Para esto, el afinador posee un dispositivo que escucha la nota de cada tecla, la compara con una nota esperada, e indica si es correcta o no. La nota escuchada en el piano será correcta si la celda que la representa tiene el mismo color que la celda que representa la nota esperada. Hay dos tipos de teclas, blancas y negras, por lo que hay dos formas de representar la nota, con una celda blanca (vacía) o negra. En el caso contrario, el dispositivo indicará que la nota del piano debe afinarse y esto se representará marcando la nota mediante una celda de color Rojo.

La siguiente imagen muestra un **ejemplo** antes y después de la verificación donde:

- Cada columna representa una tecla o nota del piano.
- Solo se representan las primeras 12 teclas del piano.

Práctica 3.6 - Ejercicios Integradores para Parcial

Ejercicio N° 6

El jardín donde habita El Sapo Pepe tiene una superficie de 11x 5 metros. En dicho lugar podemos encontrar varias mariposas. Se pide definir al procedimiento

PepeAtrapaTodasLasMariposas(). Pepe se encuentra en la esquina inferior izquierda del jardín. La imagen es solo un ejemplo.

Para poder realizar esta tarea Pepe dividió su jardín en 5 filas y 11 columnas, determinando así pequeñas parcelas de 1 metro cada una, que pueden contener Tierra, Pasto o una pequeña tarima para visualizar si hay mariposas sobrevolando.



Además, Pepe decide recorrer todas las columnas de una por vez. Para avanzar en la columna y poder llegar a ver si hay mariposas tendrá que hacerlo bajo ciertas condiciones:

- Si en la primer fila, es decir, en la primer parcela, encuentra pasto o tierra, entonces puede avanzar a la siguiente parcela
- Si en la segunda fila, o segunda parcela, encuentra pasto o tierra, entonces puede avanzar a la siguiente parcela
- Y si ,llega a la última o tercera parcela, y encuentra una tarima, entonces podrá subir a la tarima y visualizar si hay mariposas para poder atraparlas.

Se incorporaron al set de instrucciones de QDraw las siguientes condiciones e instrucciones:

- **hayTierra?**, denota Verdadero si en la celda actual hay Tierra, Falso en caso contrario.
- **hayPasto?**, denota Verdadero si en la celda actual hay Pasto, Falso en caso contrario.
- **hayTarima?**, denota Verdadero si en la celda actual hay una Tarima para poder visualizar mariposas, Falso en caso contrario.
- **hayMariposa?**, denota Verdadero si ve una mariposa desde la celda actual, Falso en caso contrario.
- Debe haber una tarima en la celda actual
- **SubirALaTarima**, sube a Pepe a una superficie elevada para poder visualizar mariposas. Debe haber una tarima en la celda actual.
- **SaltarHastaLaMariposa**, Pepe salta hasta la celda donde se encuentra la mariposa.
- **AtraparMariposa**, Atrapa la mariposa de la celda actual. Debe haber una mariposa.

Práctica 3.6 - Ejercicios Integradores para Parcial

Ejercicio N°7: Sembrando el campo

Martín es dueño de un campo en Ranchos y decidió contratar los servicios de TechAdvise Soluciones para optimizar el tiempo de siembra.

Los desarrolladores de Tech dividieron el campo de 1000 x 1000 metros en parcelas de 10 x 10 m cada una, convirtiéndolo en un tablero. Además se incorporaron al set de instrucciones de QDraw las siguientes condiciones e instrucciones, que se encuentran disponibles para utilizar, sin necesidad de implementarlas:

1. **hayTrigoSembrado?**, denota Verdadero si la parcela actual cuenta con trigo previamente sembrado, Falso en caso contrario
2. **estaTierraPreparadaParaSembrarse?**, denota Verdadero si la parcela actual ha sido fertilizada y se encuentra lista para sembrar trigo.
3. **estaRegada?** Denota Verdadero si la parcela actual se encuentra correctamente regada, Falso en caso contrario
4. **PrepararTierra**, que fertiliza la parcela actual, dejándola lista para sembrar.
5. **PlantarTrigo**, que planta las semillas de trigo en la parcela actual
6. **PlantarBanderinEnCelda**, que coloca un banderín en la parcela actual indicando que la misma posee problemas

Se pide implementar los siguientes procedimientos:

1. **PrepararParcela()**, que fertiliza la parcela actual solo si no se encuentra fertilizada.
2. **PrepararSeccionArriba()**, que fertiliza todas las parcelas de la columna actual.
3. **SembrarParcela()**, que planta las semillas de trigo solo si no hay trigo sembrado previamente.
4. **SembarSeccionArriba()**, que siembra todas las parcelas de la columna actual, solo si se encuentran preparadas.
5. **SembrarCampo()**, que siembra el campo completo.
6. **MarcarParcelaSinRiego()**, que marque la parcela actual si tiene trigo sembrado pero no se encuentra correctamente regada

Nota. Puede hacerse un dibujo del campo con sus secciones para visualizar la solución del problema.

Práctica 3.6 - Ejercicios Integradores para Parcial

Ejercicio N° 8: El marciano en el desierto

El marciano en el desierto es un juego que se trata de comer todas las manzanas que encuentra en su camino. Para trabajar convertimos el desierto en un tablero de 4 x 4.

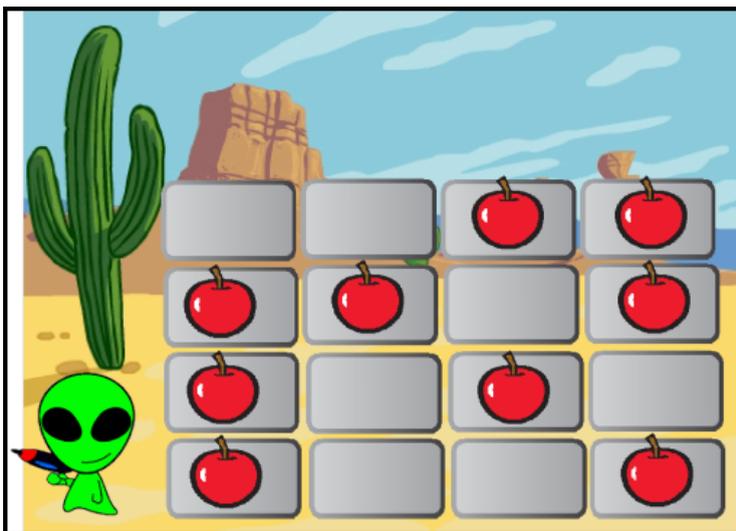
Se incorporaron al set de instrucciones de QDraw las siguientes instrucciones:

- **Comienzo**, que ubica el marciano al comienzo del recorrido.
- **ComerManzana**, que toma la manzana que encuentra en la celda actual. Debe haber una manzana en ese lugar
- **hayManzana?**, que denota Verdadero si en la celda actual hay una manzana, Falso en caso contrario.

Las nuevas instrucciones se encuentran disponibles para utilizarlas sin necesidad de implementarlas.

Se pide implementar los siguientes procedimientos:

1. **TomarYcomerManzana()**, que come la manzana de la celda actual solo si hay una en la misma.
2. **RecorrerDesierto()**, que recorre todo el desierto, comiendo las manzanas que encuentra para no perder energía.



Recuerde que es solo un tablero de ejemplo.