

## SHARIKISTONES / TAREA #08

El Shariki (las esferas, en ruso) es un videojuego de tipo puzzle desarrollado originalmente para DOS por Eugene Almezhin. El objetivo del juego era intercambiar una esfera por otra lindante (Pueden establecerse otros movimientos especiales, además de este), de forma tal de obtener líneas horizontales o verticales que contuviera tres o más esferas del mismo color. Al lograrlo, esas esferas explotaban, otorgando puntos al jugador, y apareciendo en su lugar nuevas esferas de colores aleatorios. El juego se terminaba cuando no había más movimientos válidos para realizar. Shariki se volvió muy popular y su mecánica vió múltiples reencarnaciones bajo distintos nombres, como Bejeweled, Jewel Quest, Candy Crush Saga y Pokemon Shuffle.

Vamos a trabajar con una adaptación propia del juego, implementada en Gobstones, a la que llamaremos Shariikistones. Primero explicaremos en detalle las reglas del juego, y luego su representación en Gobstones.

El juego se desarrolla en una grilla rectangular de tamaño variable, en donde en cada ubicación puede haber, o bien una ficha, o bien un obstáculo. Las fichas pueden ser de 4 tipos distintos: estrellas, círculos, rombos o triángulos. El juego consiste en intercambiar una ficha por otra, mediante un movimiento válido. Un movimiento es válido si:

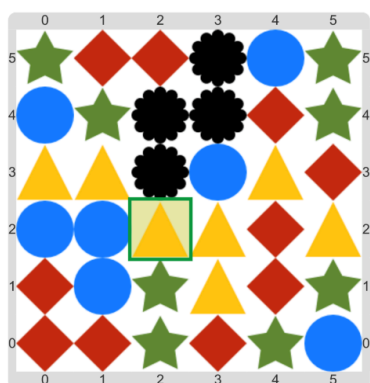
- Las fichas que se intercambian son colindantes.
- Al intercambiar las fichas se forma una línea de 3 o más fichas iguales ya sea de forma horizontal o vertical, en cualquier fila o columna de la grilla.

Al realizar un movimiento válido, se “queman” las fichas que forman parte de la línea de 3 o más fichas iguales. Estas fichas son reemplazadas por nuevas fichas aleatorias que aparecen en la misma posición.

Los obstáculos consisten en elementos que se ubican en los mismos lugares que podrían ocupar fichas, pero no pueden moverse, limitando la cantidad de movimientos que se pueden realizar, y aumentando el nivel de dificultad del juego.

Representaremos la grilla del juego en el tablero de gobstones, donde cada celda se corresponde con la ubicación en donde puede haber o bien una ficha o bien un obstáculo. Las fichas se representan con una bolita de color Negro, y tantas bolitas de color Azul como su forma (2 para círculo, 3 para triángulo, 4 para rombo y 5 para estrella). Los obstáculos por su parte se representan con dos bolitas de color Negro.

A continuación hay un tablero de ejemplo que muestra un posible tablero inicial, con vestimentas:



En este ejemplo se muestra el cabezal sobre 2,2, una ficha “triángulo”. Sin embargo no se puede mover ese triángulo al Norte para intercambiar, porque la mancha negra al norte es un obstáculo, y no puede moverse.

Hay dos movimientos válidos posibles, o bien moviendo el triángulo en (4,3) hacia el Sur, o bien moviendo el rombo en (5, 3) hacia el Oeste. En ambos casos se formarán líneas.

El equipo de desarrollo planteó los siguientes tipos para poder trabajar con el juego.

**type TipoDeElemento is variant**

PROPÓSITO. Modela el tipo de un elemento posible en una ubicación.

case Libre {}

case Obstáculo {}

case Ficha {}

**type FormaDeFicha is variant**

// PROPÓSITO. Modela el tipo de una ficha

case Círculo {}

case Triángulo {}

case Rombo {}

case Estrella {}

Además se plantearon las siguientes primitivas.

**elementoAcá**

PROPÓSITO. Describe el elemento en la ubicación actual.

TIPO. TipoDeElemento.

PRECONDICIONES. Ninguna.

**fichaAcá**

PROPÓSITO. Describe la forma de la ficha en la ubicación actual.

TIPO. FormaDeFicha.

PRECONDICIONES. El elemento en la ubicación actual debe ser una Ficha.

**SacarObstáculo**

PROPÓSITO. Saca el obstáculo de la ubicación actual.

PRECONDICIONES. Hay un obstáculo en la ubicación actual.

**QuemarFicha**

PROPÓSITO. Quema la ficha en la ubicación actual.

PRECONDICIONES. Hay una ficha en la ubicación actual.

**PonerFicha\_**

PROPÓSITO. Pone la ficha con la forma dada en la ubicación actual.

PARÁMETROS. \*formaAPoner\*. FormaDeFicha. La forma de ficha a poner.

PRECONDICIONES. La ubicación actual está libre.

**Ejercicio #01)** Realice la función **fichaAleatoria** que describe la forma de ficha con la cual reponer un espacio vacío según un proceso pseudo-aleatorio, basado en las fichas alrededor (en direcciones ortogonales, es decir, línea recta) con las reglas que siguen:

- Si la ubicación a reponer tiene al menos 2 fichas estrella alrededor y está sobre un borde, se rellena con un círculo.
- Si la ubicación a reponer tiene 3 fichas iguales alrededor, se rellena con una ficha rombo.
- Si la ubicación a reponer tiene un obstáculo alrededor, se rellena con una estrella.
- Cualquier otro caso se rellena con un triángulo.

**Ejercicio #02)** Realice la función **cantidadDe\_EnGrilla** que dado tipo de elemento describe la cantidad total de elementos del tipo dado en la grilla.

**Ejercicio #03)** Si se queman una hilera de 6 o más fichas de la misma forma, el juego aplica un bonus especial, quemando la totalidad de los obstáculos del tablero. Realice el procedimiento **QuitarLosObstáculosDeLaGrilla** que quema la totalidad de obstáculos en el tablero. Notar que no se espera compruebe si se quemaron 6 o más fichas, sino que simplemente realice el efecto posterior, dado por el bonus

: