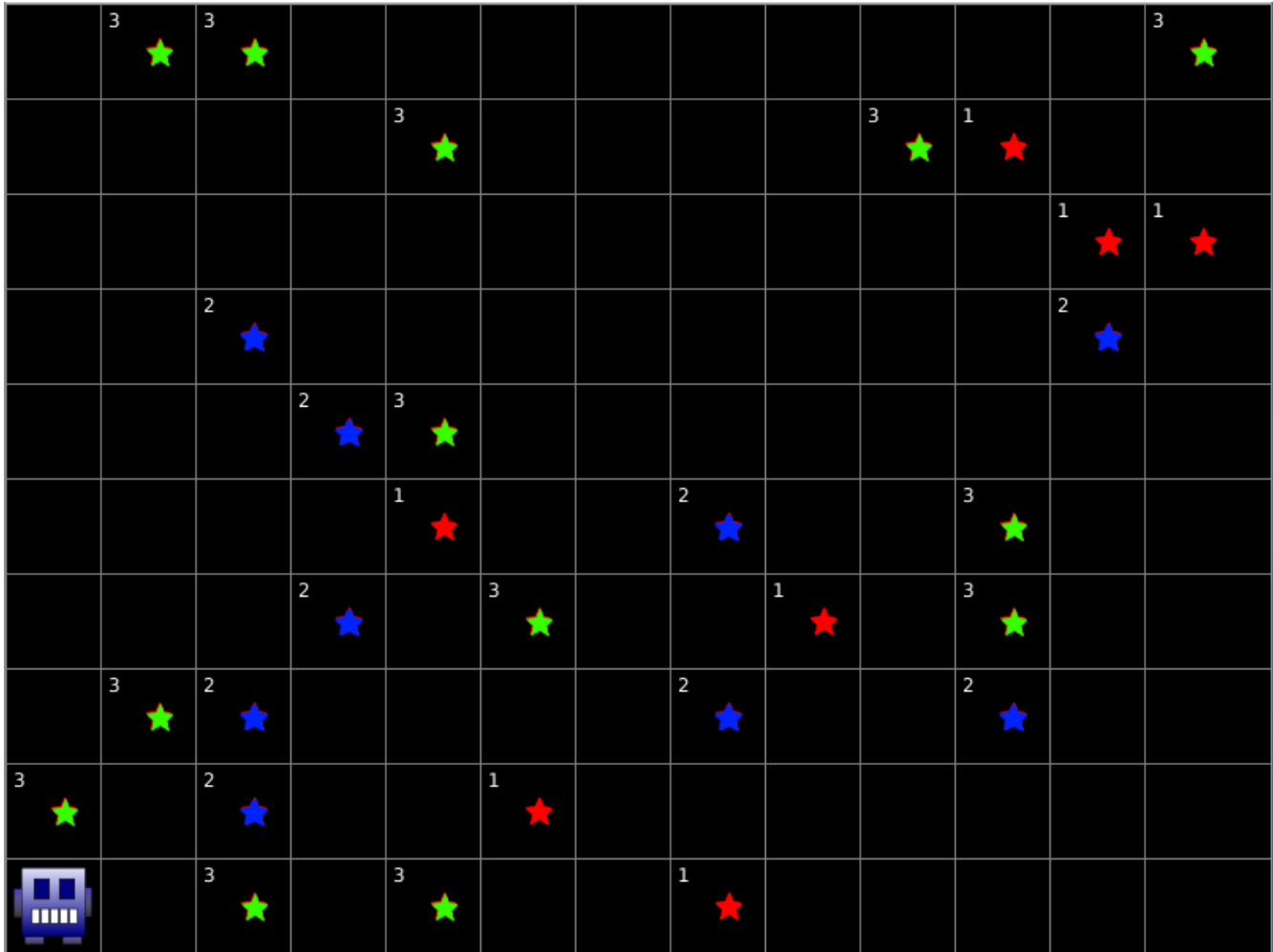


Lenguajes de Programación – Proyecto Semestral

El proyecto de este semestre consiste en implementar un intérprete para un lenguaje llamado BOT. BOT es un lenguaje de programación que permite controlar un robot que se encuentra en la pantalla, moverlo en diferentes direcciones y recoger o botar objetos. El ambiente visual de BOT es como el de la siguiente figura:



Para operar el robot, Ud. debe implementar el intérprete del lenguaje BOT, que permita leer un programa desde un archivo y ejecutar sus comandos. Por ejemplo, el siguiente código en BOT, mueve el robot 3 espacios hacia arriba, luego 3 espacios a la derecha, recoge el objeto que allí se encuentra y lo bota 2 espacios más a la izquierda y un espacio hacia abajo.

```
up 3;  
right 3;  
pick;  
left 2;  
down 1;  
drop;
```

Por supuesto, esto es sólo la punta del iceberg. Así como cualquier lenguaje tradicional de programación, BOT puede procesar expresiones aritméticas, lógicas, sentencias de control (condicionales, ciclos), funciones/subrutinas, etc.

El siguiente es un ejemplo que incluye la mayoría de los elementos del lenguaje:

```
function diagonal(let length)
begin
    let counter = 0;
    while (counter < length)
    begin
        up 1;
        right 1;
        counter = counter + 1;
    end;
    writeln "diagonal terminada";
end;
let a;
read a;
if (a == "N" and (2+3*5 >= 6) or (60-6/2 <> 8))
begin
    diagonal(5);
end;
write a;
```

Los elementos del lenguaje son los siguientes:

1. Comandos del robot: up, down, right, left, pick, drop, look
2. Variables: declaración y asignación

```
let miVariable;
miVariable = 5;
let miVariable2 = "hola";
```
3. Constantes y tipos de datos: Números decimales, booleanos y strings

```
100.3
true
false
"hola"
```
4. Condicionales

```
if (condición) begin    end else begin    end;
```
5. Ciclos

```
while (condición) begin    end;
```
6. Impresión y lectura por pantalla:

```
read miVariable; // Solo se pueden leer strings
writeln "hola " + miVariable;
```
7. Expresiones aritméticas: +, -, *, /, inverso aditivo
8. Expresiones lógicas: and, or, not

9. Expresiones de comparación: >, <, <=, >=, ==, <>

10. Declaración de funciones

```
function miFuncion(let param1, let param2, ... , let paramN)
begin end;
```

11. Invocación de funciones

```
miFuncion(10, "hola", ... , true);
```

Para desarrollar el intérprete, Ud. dispondrá de las siguientes herramientas: Una distribución de Eclipse con soporte para ANTLR4 y un proyecto en Eclipse con la implementación básica del ambiente BOT (las librerías que muestran y operan el robot en la pantalla). Todos estos recursos serán publicados por Uvirtual.

Bonificación: Retorno de valores desde funciones

Si desea tener una bonificación, implemente la sentencia `return` en las funciones y la posibilidad de recuperar ese valor desde el lugar donde se llama la función

Esta bonificación consiste en 0.5 puntos, que se sumará a la nota del último parcial. Ésta última no podrá superar el valor 5.0 luego de adicionar la bonificación.

Para acceder a la bonificación, es obligatorio, en la segunda entrega del proyecto, indicar a través de uvirtual que Ud. desea participar en dicha bonificación. De lo contrario el profesor no revisará dicha funcionalidad en su proyecto.

Para desarrollar el intérprete, usted dispondrá de una distribución de Eclipse con soporte para ANTLR4 y un proyecto en Eclipse con los elementos básicos del proyecto. Todos estos recursos serán publicados por Uvirtual.

Entregas

Primera Entrega:

- Intérprete básico de Bot. Sólo comandos para mover el robot: up, down, right, left

Segunda Entrega:

- Intérprete completo de Bot (todos los elementos del lenguaje).