

ENTREGA PARSER

GRUP03

Hemos implementado una librería Parser para realizar la separación de las distintas instrucciones y datos que vamos a recibir del servidor en nuestro programa para la Robocup. De esta forma podemos almacenarlas cómodamente. Para ello hemos utilizado tres atributos de tipo vector<int> (inicio, fin1 ,fin2, espacios), un objeto de tipo Datos (valor) y tres funciones dos para la búsqueda de paréntesis y espacios y, la tercera, para realizar la separación de los datos y enviarlos para que sean almacenados.

FUNCIÓN LEERPARENTESIS(STRING CAD):

La función recibe un string. Evaluaremos distintos aspectos de este para saber como debemos actuar. Las posibilidades son las siguientes:

Si la cadena empieza contiene la secuencia de caracteres “see” o “sense” y, además, la longitud de esta es superior a 15 caracteres empezaremos a contar desde la posición 1 de la cadena ya que el primer carácter no nos interesa puesto que sabemos que es un paréntesis que no contiene una instrucción sino el indicador del tipo (see o sense). A partir de aquí el proceso es el siguiente:

- En *inicio* almacenamos la posición de los paréntesis de apertura “(“. En caso de que haya dos seguidos borramos la posición del anterior y la sustituimos por la del último. Si recibimos (see 0 ((flag c) 38.1 -23) ((flag c b) 39.6 28)... : la posición del primer paréntesis de inicio será la del tercero, la segunda la del quinto y así sucesivamente (marcados en rojo), ya que solo nos interesan los paréntesis que delimitan instrucciones y valores.
- En *fin1* almacenamos la posición de los paréntesis de cierre “)”. Esta variable solo se rellena para instrucciones de tipo see ya que es la única que tiene instruccione y valores separados con estos paréntesis. par la misma cadena anterior (see 0 ((flag c) 38.1 -23) ((flag c b) 39.6 28)... fin1=[15,36,...] (marcados en rojo).
- En *fin2* almacenamos la posición de los paréntesis de cierre “)”. Para instrucciones de tipo see: (see 0 ((flag c) 38.1 -23) ((flag c b) 39.6 28)...y para instrucciones de tipo sense: (sense_body 0 (view_mode high normal) (stamina 8000 1)...

Si el tamaño de la cadena es superior a 15 caracteres o no se encuentra alguna de las secuencias antes dichas el punto de unicio de lectura será la posición inicial 0, ya que en estos casos si es necesario leer el primer paréntesis, los tipos de cadenas que recibimos son:

(see 1250)

(hear referee kick_off)

(init I 1 before kick_off)

Como se puede ver en el primer ejemplo tenemos una cadena que es see pero su longitud es inferior a 15 caracteres. El tamaño que hemos elegido es suficientemente grande para no tener problemas.

FUNCIÓN LEERPESPACIOS(String CAD):

Función muy simple que busca los espacios y los guarda las posiciones en el vector espacios

FUNCIÓN SEPARARDATOS(String CAD,DATOS &VALOR):

Hemos separado la función en dos casos, según el string que nos llegue. Si la cadena es de tipo see con un tamaño superior a 15 caracteres y el resto de cadenas. Esto se debe a que, cuando la instrucción es de tipo see necesitamos las posiciones almacenadas en *fin1* y en el resto no.

El funcionamiento es análogo para los dos casos. Copiamos el trozo de cadena que nos interesa en un char (las instrucciones en inst y los datos en data) y posteriormente lo enviamos. Vamos separando la cadena en trozos según vamos avanzando por ella, de tal forma que cada vez solamente atacamos la separación de una instrucción y sus datos.

SEE: Para la copia utilizamos una variable para calcularnos cual es el punto de inicio de copia (b, d) y otra para la longitud del string que queremos recorrer (a, c). El punto de inicio será la posición siguiente al paréntesis de apertura para las instrucciones y la posición siguiente a un espacio para los datos. El tamaño de lectura estará entre las posiciones de *inicio* y *fin1* para las instrucciones y entre *espacios* y *espacios-fin2* para los datos.

RESTO: Siguiendo el caso anterior, el punto de inicio de copia (b) será la posición siguiente a la del paréntesis almacenado en *inicio* y el tamaño de copia entre *inicio* y *fin2*. Ahora separamos la instrucción, que estará entre *inicio* y el *primer espacio* y los datos, que estarán entre *espacios* o *espacios* y *fin2* (en vez de *fin2* utilizamos el tamaño de la cadena que hemos pasado para volver a separar sin el último paréntesis)

Para más información sobre la clase datos véase el documento de la entrega 2 que se encuentra en la sección de descargas del repositorio (ENTREGA2.pdf en <http://ii2-grupo3.googlecode.com/files/ENTREGA2.pdf>)