* **Lexemes**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Token | Regular expression | Example | Remarks |
| identifier | ID | [a-zA-Z][\_a-zA-z0-9]{0,20} | Myvar  My\_Var12345 | An identifier can only start with a letter, followed by up to 20 alphanumeric or underscore/upper case/ lower case characters. |
| number | NUM | (0|(([1-9])[0-9]\*) | 0,3,7,9 | Number can either start and only be a 0 or start with 1-9 and be followed by many numbers ranging from 0-9 |
| function | FUN |  | Set(x,y) | ID2 |
| Comment | Com | \# | # | Any string after # is a comment and not seen by compiler as code |
| Operators: arithmetic | ADD | \+ | + | Addition: adds value a to value b |
|  | SUB | \- | - | Subtraction: subtract value a from value a |
|  | MUL | \\* | \* | Multiplication : multiplies value a by value b |
|  | DIV | / | / | Division: divides value a by value b it is a floor function so it wont return a decimal number |
|  | ASSIGN | \= | = | Assignment: assigns value a to value b |
| Operators: comparison | EQU | \$ | $ | Equality: compares if value a equal’s value b |
|  | SMALL | \< | < | smaller then: compares is value a is smaller then value b |
|  | SMALLQUI | \ | \ | smaller or equals: compare if value a is smaller or equals value b |
|  | NOTEQUI | \! | ! | Inequal: compares if value a is inequal to value b  Returns true if they are not the same value |
| Operators: Logic | OR | \I | I | OR: True only when a or b are true |
|  | AND | \& | & | AND: a and b |
|  | NOT | \!! | !! | NOT: true when Not a is True |
|  | CONCA | \~ | ~ | Concatenate value that is already in the array and the value we want to add it to the array |
| Punctuation: | DOT | \. | . | point |
|  | SEMI | \, | , | SEMI |
|  | ENDL | \; | ; | End of line |
|  | OPPARENT | \ ( | ( | Open brace |
|  | CLPARENT | \) | ) | Closing brace |
|  | OPCURLY | \{ | { | Opening curly brace |
|  | CLCURLY | \} | } | Closing curly brace |
|  | OPBRACKET | \[ | [ | Opening bracket |
|  | CLBRACKET | \] | ] | Closing bracket |
| Simple space: | SPACE |  |  | Simple space |
| Reserved Words: | LOOP | loop | loop | Conditional loop |
|  | IF | if | if | Selection statement |
|  | ELSE | else | else | Else Selection statement |
|  | DEFINE | defining | defining | A reserved word to be put before any function definition |
|  | TRUE | TRUE | TRUE | Boolean true |
|  | FALSE | FALSE | FALSE | Boolean false |
|  | BEGIN | BEGIN | BEGIN | Denotates beginning of the main |
|  | END | END | END | Denotates end of the main |
|  | RETURN | RETURN | RETURN | Returns a certain value from a function |

Wumpus World Constants:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WUMPUS | WU | wu | wu | Represents Wumpus |
| AGENT | AG | ag | ag | Represents agent |
| PIT | PT | pt | pt | Represents pit |
| GOLD | GO | go | go | Represents gold |
| BREEZE | BR | br | br | Represent breeze |
| GLITTER | GL | gl | gl | Represents glitter |
| STENCH | ST | st | st | Represents stench |
| EMPTY | EM | em | em | Empty flag |
| SCREAM | SC | sc | sc | When a Wampus is killed it releases a scream |
| BUMP | BU | bu | bu | Represents bump when hitting a wall |
| EAST | EA | ea | ea | Represents east |
| WEST | WE | we | we | Represents west |
| SOUTH | SO | so | so | Represent south |
| NORTH | NO | no | no | Represent north |

WHY PART

Why did we add the new tokens EA,WE,SO,NO it was because of the built in functions that we implemented especially the function move we needed a reserved word for the direction

\n we removed it because we felt there is no need for it

|  |
| --- |
| A |
| P |
| g |
| B |
| GLI |
| S |
| E |
| SCREAM |
| BUMP |
| W |

|  |
| --- |
| WU |
| AG |
| PT |
| GO |
| BR |
| GL |
| ST |
| EM |
| SC |
| BU |

We changed the tokens for

To smaller letters

For aesthetic reasons give it

More meaning