|  |  |  |  |
| --- | --- | --- | --- |
| 分類 | 指令名稱 | 指令內容 | 開放  關卡 |
| 動作 | step() | 玩家載具(車、坦克、船)向前移動一格。 | 1 |
| turnRight() | 玩家載具(車、坦克、船)順時針轉 90 度。 | 2 |
| turnLeft() | 玩家載具(車、坦克、船)逆時針轉 90 度。 | 3 |
| printf() | 輸出參數值、數字、字元、字串。範例如下：  輸出固定字串：printf("123");  輸出單一字元：printf("%c",'A');  輸出字串：printf("%s",字串陣列名稱);  輸出變數值：printf(變數名稱);  輸出十進位整數：printf("%d",變數名稱);  輸出浮點數：printf("%f",變數名稱); | 5 |
| scanf() | 輸入整數、浮點數、字元等變數值，範例如下：  輸入：scanf("輸入格式",&變數名稱);  輸入字元：scanf("%c",&字元變數名稱);  輸入整數：scanf("%d",&整數變數名稱);  輸入浮點數：scanf("%f",&浮點數變數名稱);  輸入字串：scanf("%s",字元陣列名稱); | 6 |
| fire() | 向前方發射子彈，攻擊距離為 2 格，攻擊力可於裝備內提升。 | 15 |
| 變數  宣告 | int 變數名稱 | 宣告一個變數，型態為整數值。 | 6 |
| char 變數名稱 | 宣告一個變數，型態為字元。 | 6 |
| 函式 | main() | main是主程式，是一個程式開始的地方與主體。 | 1 |
| for(迴圈次數設定) | 用來創出一個迴圈，使用一個變數設定進入迴圈的條件，若未達成則跳出迴圈。  使用方法如下：  for(參數初始值;進入迴圈條件;每跑一次迴圈後參數變化){  欲執行指令  }  範例如下：  int i=0;  for(i=5;i>0;i++){  ...  } | 10 |
| function 函式名稱(){} | 在Blockly模式中用來創出一個自訂函式。 | 17 |
| void 函式名稱(){…} | 建立一個自訂函式也稱作副程式，能夠自訂函式型態、函式名稱、函式內容。  自訂函式後，想呼叫自訂函式時，只需要直接打出函式名稱即可。  使用方法：  傳回值型態 函數名稱(參數一的型態 參數一的名稱, 參數二的型態 參數二的名稱, ....)  {  變數宣告  程式碼  return 傳回值;  }  傳回值型態若為void則代表不會回傳任何數值。  範例1：  void ABC(){  step();  step();  }  void main(){  turnRight();  turnRight();  ABC();  }  宣告一個自訂函式ABC，當呼叫ABC時會執行2次step。  而整個程式執行後會是右轉兩次後前進兩格。  範例2：  int C(int a){  int b=5;  a=a+b;  return a;  }  void main(){  int sum=2;  sum=C(sum);  }  宣告一個自訂函式C，當呼叫C時，需給C一個型態為int的變數，而C會將這個變數加上b的數值後傳回來。  整個程式執行後，sum的值將變為7。 |  |
| call function 函式名稱() | 在Blockly模式中用來呼叫已創建的自訂函式。 | 17 |
| becameCar() | 使載具變形為汽車。 | 27 |
| becameTank() | 使載具變形為坦克。 | 27 |
| becameShip() | 使載具變形為船。 | 27 |
| getKeyArray(整數陣列名稱,陣列大小) | 將密碼陣列存入給予的整數陣列中。  用法：  int array[6];  int size=6;  getKeyArray(array,size); | 31 |
| getDistance(整數陣列名稱) | 將距離陣列存入給予的整數陣列中，獲得的陣列內容代表的是每一段直線要走的距離。  用法：  int dist[10];  getDistance(dist); | 33 |
| getDirection(字元陣列名稱) | 將方向陣列存入給予的字元陣列中，獲得的陣列內容代表的是每一次要做的動作。  用法：  char dire[5];  getDistance(dire);  字元對應動作：  'L'：左轉  'R'：右轉  'F'：開火 | 34 |
| getKey(整數變數位址x,整數變數位址y) | 將密碼存入給予的2個整數變數中，給予兩個整數變數的位址，系統將自動給予數值。  用法：  int x,y;  getKey(&x,&y); | 37 |
| getBox(字元陣列名稱) | 將寶箱裡所藏的字串存入給予字元陣列中。  用法：  char str[256];  getBox(str); | 38 |
| getString(字元陣列名稱) | 將關卡給予的字串存入給予的字元陣列中，字串依照關卡不同所要做的動作也不同。  用法：  char str[256];  getString(str); | 39 |
| getMap(二維整數陣列名稱,地圖的長度) | 將代表關卡的數字地圖存入給予的二維陣列中，陣列裡的每個元素所代表的就是對應到的地圖資訊。  用法：  int map[9][9];  int size=9;  getMap(map,size);    陣列數值對應地型：  0：障礙物  1：草原  2：河流  3：沙漠  4：終點 | 39 |
| 判斷式 | if(條件){…} | 用來判斷是否達成條件，若達成條件則執行大括號內的指令。 | 7 |
| if(條件){…}else{…} | 用來判斷是否達成條件，若達成條件則執行大括號內的指令，沒達成的話則執行else內的指令。 | 8 |
| switch(條件參數){…} | 用來判斷多重選擇，以一個參數的值做為選擇的key，再用case來選擇執行指令。  使用方法如下：  switch(關鍵參數名稱){  case 關鍵參數值：  欲執行指令...  break;  case 關鍵參數值：  欲執行指令...  break;  ...  }  範例如下  switch(hint){  case 'R'：  turnRight();  break;  case 'L'：  turnLeft();  break;  ...  } | 9 |