

# レトロな 打ち込み ゲーム



クラシックなゲーム、Robotsをプレイしよう。全てのロボットはただひたすらプレイヤーに向かってくる。ロボット同士をぶつけるとスクラップになる。ロボットをスクラップにぶつけることで壊すこともできる。

**ロボット同士をぶつかるスクラップになる。スコアとマスにマスを増やす**

```
(defun robots ()
  (loop named main
    with directions = '( (q . -65) (w . -64) (e . -63) (a . -1)
                        (d . 1) (z . 63) (x . 64) (c . 65))
      for pos = 544
      then (progn (format t "~%qwe/asd/zxc to move, (t)eleport, (l)eave:")
                 (force-output)
                 (let* ((c (read))
                       (d (assoc c directions)))
                    ← assocで入力からオフセットを得る
                    (cond (d (+ pos (cdr d)))
                          ((eq 't c) (random 1024))
                          ((eq 'l c) (return-from main 'bye))
                          (t pos))))
                  ← プレーヤーがゲーム終了を選択
                  → ロボットを増やすにはこの数値を変える
                for monsters = (loop repeat 10
                                     collect (random 1024))
                then (loop for mpos in monsters
                           collect (if (> (count mpos monsters) 1)
                                         mpos
                                         (cdar (sort (loop for (k . d) in directions
                                                             for new-mpos = (+ mpos d)
                                                             collect (cons (+ (abs (-
                                                                 (mod new-mpos 64)
                                                                 (mod pos 64)))
                                                                 (abs (- (ash new-mpos -6)
                                                                 (ash pos -6))))
                                                            new-mpos))
                                         (key #'car))))))
          when (loop for mpos in monsters
                     always (> (count mpos monsters) 1))
            return 'player-wins
          do (format t
                    "~%|~{<~|~>,65;::~A~>}|")
              (loop for p
                    below 1024
                    collect (cond ((member p monsters)
                                   (cond ((= P pos) (return-from main 'player-losses))
                                           (> (count p monsters) 1) #\#)
                                  (t #\A)))
                            (= p pos)
                            #\@)
                    (t
                     #\ ))))))
```

**ループにmainと名づけておけば、いつでも"return-from"を使ってこのループへ抜けることができる**

**64文字幅のゲーム盤において、それぞれの方向に移動するたに加算するオフセット**

**ゲーム盤の大きさは 64x16=1024文字分。(random 1024) でランダムな位置を選択**

**ここでプレーヤーまでの「マンハッタン距離」を計算**

**すべてのモンスターがスクラップになったかどうか調べる**

**プレーヤーまでの距離でソートして、一番近くなる移動を採用**

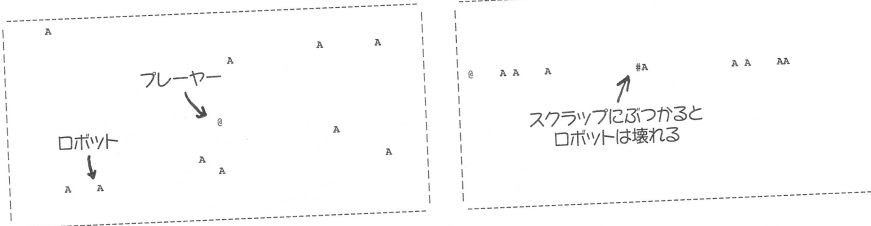
**1箇所に複数のロボットがあるということはスクラップになっているということ**

**ゲーム盤の各マスについてループ**

**この無茶苦茶なformatコマンドの意味はまだ気にしなくていい**

**プレーヤーがロボットと同じマスにいたら残念な知らせとなる**

**複数のロボットが積み重なってたらスクラップを描画**



コードに基本的  
見返す必要がある  
より完全な説明を  
ゲームに勝つに  
にする。移動に  
いて、配置の通り  
楽しんでくれ

## 11.9 本章小结

この章だけで  
良い紹介にはな

- format
- format  
指定で  
備えて
- format  
る。
- Lisp の
- 数値を
- format  
ことな

ことな