

robots

lickray/ray

## 1. プログラムの仕様

### 1.1 プログラムの機能

実行すると、robots というゲームで遊ぶことが出来る。

### 1.2 robots のルール

#### 1.2.1 概要

プレイヤーは悪いロボットと戦っている。図 1 では、'@'がプレイヤー、'+'がロボットを、'・'が空白を表す。また、図の左下にある Level は現在のステージのレベルであり、Score は現在の獲得スコアを表す。

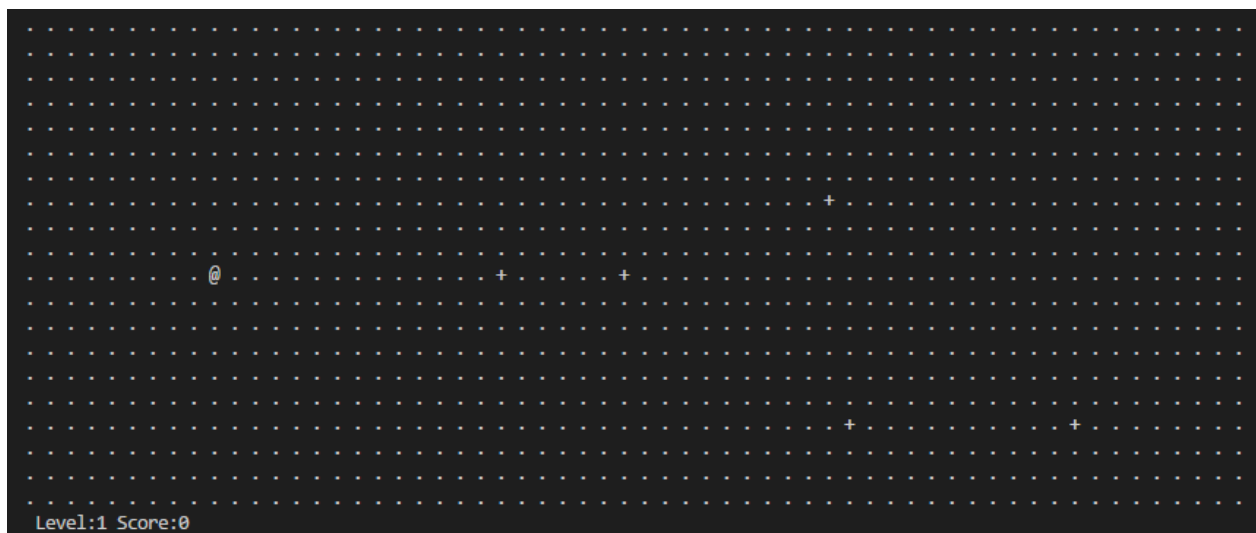


図 1 robots のイメージ

プレイヤーは 1 回の動作で 8 方向（上下左右と斜め）のいずれかに 1 マス移動することができる。または、まったく動かないこと、どこかにテレポートする（行き先はランダムに決まる）ことも選択可能である。

プレイヤーが動くと、次にロボットが移動する。ロボットも 8 方向のいずれかに 1 マス移動できる。ロボットは、常にプレイヤーに最も接近する方向に移動する。ロボット同士が衝突すると壊れてスクラップになる。また、ロボットがスクラップに衝突した場合も、そのロボットは壊れる。ロボットが壊れるとスコアが加算される。スクラップは'\*'で表される。また、プレイヤーがロボットに捕まるとゲームオーバーとなり'☆'が表示される。これは確保されたことを表す。

フィールド上のすべてのロボットが壊れると、次のステージが現れる。その際、ステージのクリアボーナスのスコアが加算される。

これをゲームオーバーになるまで繰り返し、ハイスコアを狙うゲームである。

### 1.2.2 動作の制約

1.2.1 で述べたように、プレイヤーは 8 方向に移動出来る。ただし、空白のある場所以外に移動することはできない。

また、プレイヤーのテレポートについても同様である。

### 1.2.3 ゲームの設定

#### 1.2.3.1 ステージのレベル

ゲーム開始時には、ステージのレベルは 1 である。フィールド内のロボットを全滅させることで、レベルが 1 上がる。ステージのレベルに上限はない。

#### 1.2.3.2 ロボットの数

ゲーム開始時にはロボットの数は 5 であるが、これは以下の式によって決定される。

$$\text{ロボットの数 } N = \text{ステージのレベル} \times 5 \quad (N \leq 40)$$

つまり、ステージのレベルが 8 以上では常にロボットの数は 40 ということになる。

#### 1.2.3.3 スコア

ロボットを 1 体破壊するごとに 1 ポイント加算され、ロボットを全滅させるごとにステージのレベル  $\times 10$  ポイントのボーナスが加算される。

例えば、最初のステージをクリアした際には

$$\text{スコア(ポイント)} = 5 \left( \text{破壊したロボットの数} \right) + 1 \left( \text{ステージのレベル} \right) \times 10$$

が得られる。

## 1.3 使い方

図 1 (p2 参照) のようにプログラムはステージの状態、ステージのレベル/獲得スコアを表示し、待機する。

その際、ユーザーは表 1 (p4 参照) のような入力を行うことでプレイヤーをコントロールすることが出来る。

表1 プレイヤーの動作

入力するキー	対応した動作
0	テレポート
1	左下に移動
2	下に移動
3	右下に移動
4	左に移動
5	待機
6	右に移動
7	左上に移動
8	上に移動
9	右上に移動

## 2. 動作環境

以下の環境での動作を確認している。Windows 用のモジュールを使用しているため、原則 Windows 上のみで動作するはずである。

OS	Windows10 Pro (最新バージョン:2018/06/27現在)
プロセッサ	Intel® core™ i7-8700K CPU @ 3.70GHz 3.70GHz
RAM	16.0GB