アルゴリズム学習のための流れ図ゲーム

太　田　幸　一

要旨

本学の授業である「アルゴリズム論」でアルゴリズム学習のために作成したゲームを紹介する。ゲームは，プログラムを使ってロボットを動作させ，目的を達成させるパズルゲームである。

プログラムは，私が独自で設計・作成した流れ図による言語のプログラムで，マウスドラッグだけで流れ図が作成・実行できる。そしてゲームの目的は，目標達成となる流れ図を作成することである。

さらに，複数のロボットに対してそれぞれの流れ図を作成し，それらを一斉に動作させることによってプログラムを競い合うゲームを紹介する。これらのゲームで，プログラムの基本機能の動作・繰り返し・判断・関数（サブルーチン）を順に学習させる。

キーワード：アルゴリズム，ゲーム，パズル，流れ図，プログラミング，情報処理教育

１．は　じ　め　に

本学の講義「アルゴリズム論」では，プログラミングに慣れ親みながら学習の効果を高めるために，独自に制作したアプリケーションソフトを活用している。そして「アルゴリズム論」の初期段階の講義で，構造化プログラミングのプログラムの基本動作である連接・繰り返し・判断・関数（サブルーチン）を順に学習させる目的で，そのアプリケーションソフトを応用したゲームを提示している。

提示しているゲームは，まず流れ図のプログラムで動作するロボットのパズルゲームである。

２．ロボットのパズルゲーム

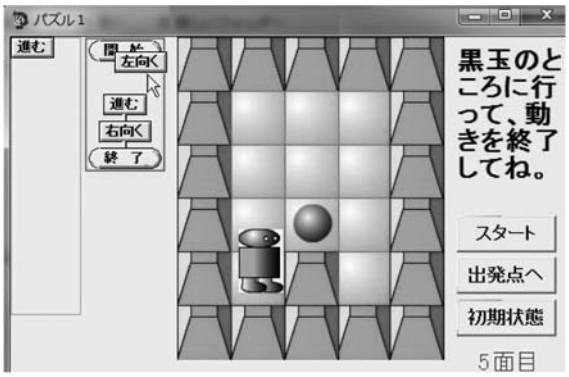
2.1　連接の流れ図のパズル

ゲーム面は，図１の中央部の「障害物の山」で囲まれた部分で，ロボットのいる場所を斜め上から見た面である。

ロボットの向きは，画面の上下左右の４通りだけで，ロボットの動きは，流れ図記号の１回の動作で，90度左と右に向きを変えるのと，向いている方向に１マス移動する３パターンだけである。このゲームは，ロボットを黒玉の位置に移動させ，そこで動きを終了するのが目的である。

ロボットの動きは，図１の左側の流れ図で行う。

図１　パズルゲーム連接



ロボットが目的を達成できるように動かすため，図１の左側に配置された，限られた数の流れ図記号をマウスでドラッグし，開始と終了の間に入れて流れ図を組み立てて行く必要がある。

なお。流れ図記号は JIS に従った形を使用している。

ロボットの動きを開始するためには，右側のスタートのボタンをクリックする。

図１は，流れ図記号である処理記号の「進む」・「左向く」・「右向く」の３パターンの動作を組み合わせている様子である。

連接は，ロボットの動作を上から下に順番に並べ，その順番に従ってロボットを動かすものである。

流れ図の作成は，まず左端に並んでいる流れ図記号の中の一つをマウスで左クリックする。すると，クリックされた左端の流れ図記号が消え，図１の左側の様にマウスカーソルの上にその流れ図記号が現れる。そして，マウスの動きによって，マウスカーソルと共にその流れ図記号が移動する。図１では「左向く」の処理記号がそれにあたる。そして，流れ図の中の追加挿入したい位置の縦棒である「流れ線」記号の上にマウスカーソルを合わせると，図１のように流れ図のその位置の流れ線が下に伸びて，流れ図全体が下の方に広がり，追加挿入を受け入れる形となる。その状態で左クリックして，流れ図記号を流れ図のその位置に追加挿入する。

流れ図内の流れ図記号を左端に戻すことも可能である。流れ図記号内の流れ図記号を左クリックし，左端に移動して左クリックするのである。

限られた数の流れ図記号を正しく組み合わせて流れ図を完成させ。右端の「スタート」のボタンをクリックして流れ図を実行すれば，そのパズル面はクリアである。

図１の場合，右上の「黒玉のところに行って，動きを終了してね。」の指令を達成し，この５面目をクリアすれば，次の面に進んでパズル画面は６面目となる。指令を失敗すれば「残念！！また遊んでね。」の文字が現れ，５面目の再度の挑戦となる。

図１の５面目のパズル面では，流れ図を上から順に「左向く」・「進む」・「右向く」・「進む」の４つの処理記号の連接とし，「スタート」のボタンをクリックすれば指令達成とな

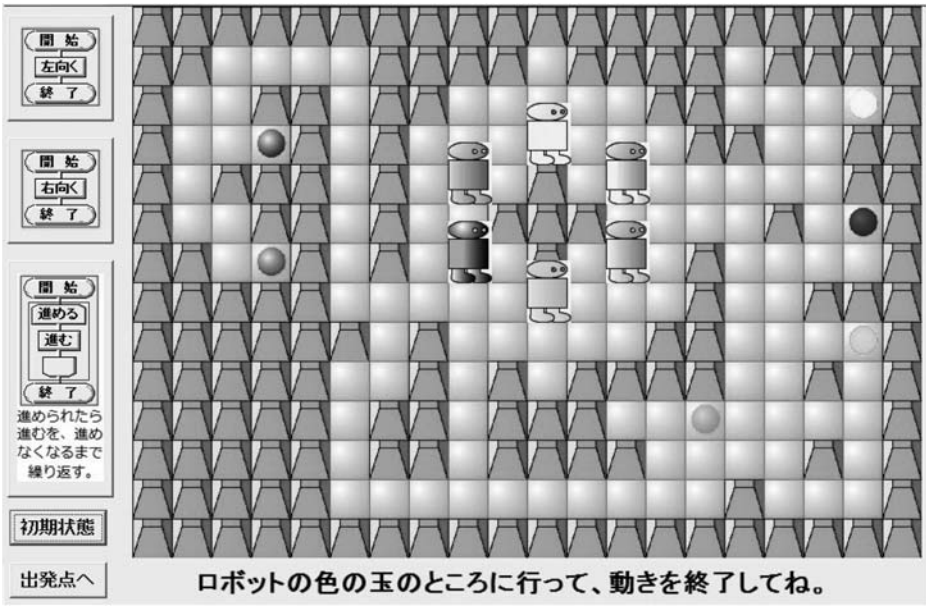
る。

右側の「出発点へ」のボタンをクリックすると，作成中の流れ図は作成中の状態で，ロボットを出発点の位置に戻し，「初期状態」のボタンをクリックすると，このパズル面の最初の形に戻す。パズル面の，再度の挑戦時にはそれらのボタンを利用する。

2.2 繰り返しの流れ図のパズル

図２の繰り返しの流れ図のパズルは，図１の連接の流れ図のパズルとはルールの異なるパズルである。

図２　パズルゲーム繰り返し



繰り返しの流れ図のパズルは流れ図を作成するのではなく，あらかじめ用意された３種類の流れ図ボタンを，ロボットを動かす順に押していき，複数個のロボットを同時に操作して，各ロボットのそれぞれの目的位置に移動させて停止させるパズルである。

図２のパズルの流れ図ボタンは，「１回左向く」，「１回右向く」，「向きの方向に進めるだけ進む」の３種類である。

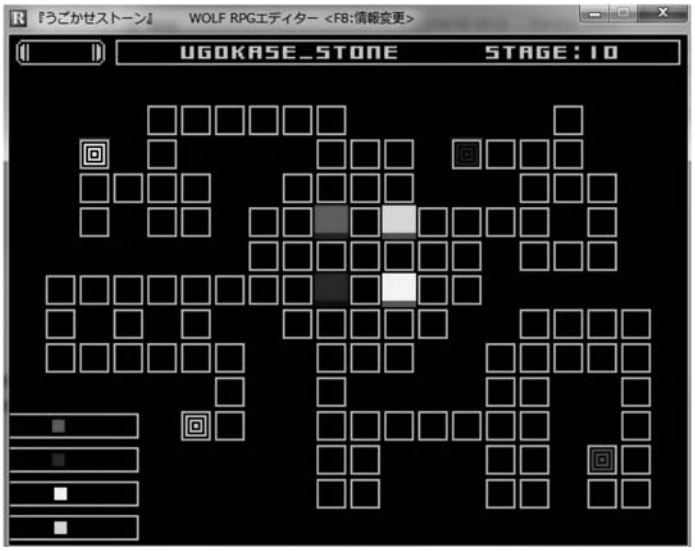
この種のパズルは，氷でできた滑る床や，地面全体が１方向に傾いて転がる岩石，といった設定のミニゲームとして，ロールプレイングゲームやアドベンチャーゲームの中にしばしば登場する。

ルールが単純な割に，動作がダイナミックであり，簡単なものから複雑なものまで作れるので，アクションパズルとしてもしばしば現れる。

それらの既存のゲームから，パスル面を移植して作成したのが図２である。

ロボットが１体だけの面は，慣れるとそんなに難しくなくなる。

図３　うごかせストーン



そこで，色んなこの種類のゲームの中でも難易度の高いものとして，ネット上に公開されている「うごかせストーン」という，図３のようなパズルに注目した。

図３で動かすのは，図２のロボットではなくストーンである。図３の画面は，赤・青・黄・スカイブルーの４種類のストーンとしてそれぞれ同時に移動し，左上の黄・左下のスカイブルー・右上の青・右下の赤の位置の上に，４つのストーンを同時に停止させるステージ10の面である。各ストーンの移動は，キーボードの上下左右キーを押して行う。

当然，図２のロボットの移動手段である３種類の流れ図ボタンをクリックして操作するよりも，図３の上下左右キーを押して操作する方が簡便である。しかし図２では流れ図，特に繰り返しの流れ図を学習するためにこのような作りとした。

図２の６体のロボットを動かすパズル面は，「うごかせストーン」の最も難しい面で，難易度が非常に高く，YouTube にこの「うごかせストーン」の図２に示されているパズル面の解き方が紹介されている。

2.3　判断の流れ図のパズル

図４は，面上にある赤玉と青玉の上をロボットが通過しながら，青玉だけを緑玉に変えて，ロボットの動きを終了させる指令を達成するパズルである。

判断の流れ図パズルは，連接の流れ図パズルと同じルールのパズルで，いくつかの流れ図記号を組み合わせて１つの流れ図を作り，その流れ図を実行させて指令を達成させるパズルである。

図４では，二重の前判断繰り返し記号「進める」の流れ図記号を使って，流れ図を作成している途中の様子を表している。作成中の流れ図に，「左向く」の処理記号を正しい位置に挿入すれば完成である。

図４　パズルゲーム判断

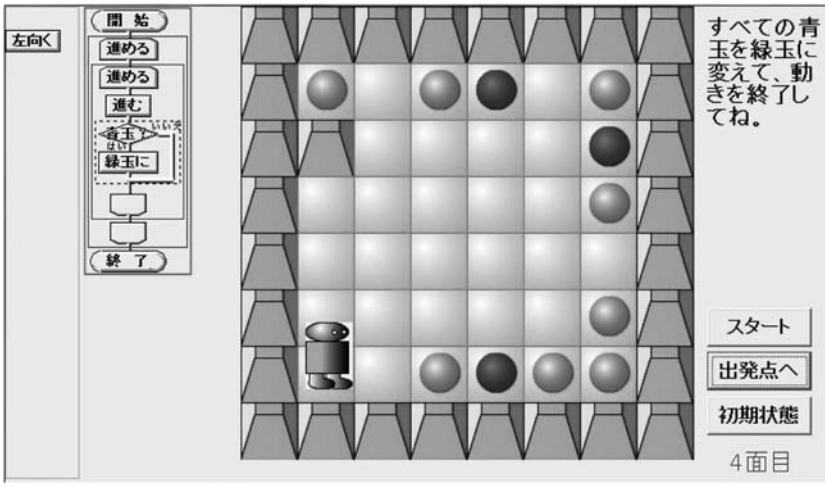


図５　パズル面のデザイン

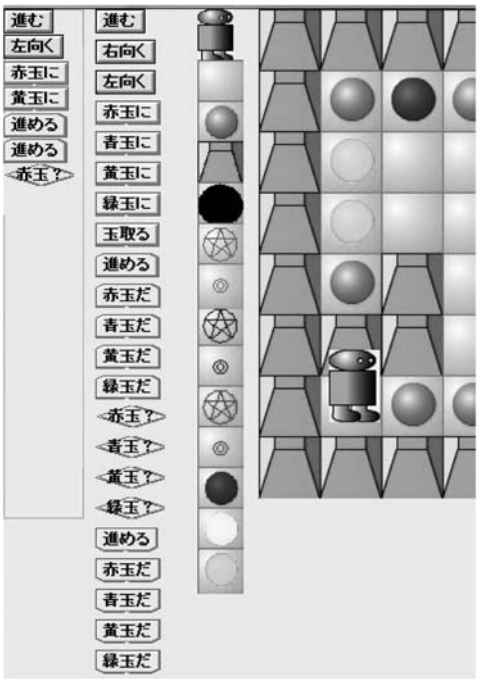


図４の流れ図の中の流れ図記号は，以下のパズル面のデザインで紹介する。

2.4　パズル面のデザイン

図４のような連接・繰り返し・判断の３種類の流れ図記号からなるそれぞれのパズル面は，csv ファイルで保存する。そしてそれらの面が簡単に作れるように，任意のパズル面

の csv ファイルがデザインできるアプリケーションソフトを用意した。

パズル面デザインソフトを起動すると，図５のような画面が表示される。

この画面の各列については以下で順に解説する。

それぞれのパズル面で提供する流れ図記号の指定は，左側の二つの列で行う。

左から２列目が，パズル面で使える流れ図記号の種類である。この列からパズルに必要な流れ図記号を，左端の列に必要な数だけ移動して並べれば完成である。

左から２列目の流れ図記号は，上から下に，連接の処理記号として「進む」・「右向く」・「左向く」・「赤玉に」・「青玉に」・「黄玉に」・「緑玉に」・「玉取る」，繰り返しの前判定繰り返し記号として「進める」・「赤玉だ」・「青玉だ」・「黄玉だ」・「緑玉だ」，判断記号として「赤玉？」・「青玉？」・「黄玉？」・「緑玉？」，さらに繰り返しの後判定繰り返し記号として「進める」・「赤玉だ」・「青玉だ」・「黄玉だ」・「緑玉だ」の計22種類である。

連接の処理記号「赤玉に」から「緑玉に」は，ロボットのいる位置に玉があれば指定の玉の色に変え，「玉取る」はロボットのいる位置に玉があればその玉を取る。

繰り返しの前判定繰り返し記号と後判定繰り返し記号について，それぞれの「赤玉か」から「緑玉は」は，ロボットのいる位置に指定の色の玉がある間繰り返す。

前判定繰り返し記号は，繰り返しの最初に繰り返すかどうかを判定し，後判定繰り返し記号は，繰り返しの最後にさらに繰り返しを続けるかどうかを判定する。

判断記号の「赤玉？」から「緑玉？」は，ロボットのいる位置に指定の色の玉があれば真となって「はい」の方に実行が移り，なければ偽となって「いいえ」の方に実行が移る。

次にパズルの面のデザインである。

左から３列目が面の構成部品で，上から下に，「ロボットの位置」・「何もないマス」・「赤玉のあるマス」・「障害物の山」・「障害物の穴」・「黄ワープ元」・「黄ワープ先」・「青ワープ元」・「青ワープ先」・「赤ワープ元」・「赤ワープ先」・「青玉のあるマス」・「黄玉のあるマス」・「緑玉のあるマス」の14種類である。この中で，「ロボットの位置」・「黄ワープ先」・「青ワープ先」・「赤ワープ先」の４種類は，パズルの１面中に一度のみの配置である。

この列からパズルの面に配置したい部品を，右側のパズルの面に置いていく。

「ロボットの位置」を配置した位置は，パズル面のロボットの初期位置となる。

「何もないマス」・「赤玉のあるマス」・「青玉のあるマス」・「黄玉のあるマス」・「緑玉のあるマス」はそのままの意味である。

「障害物の山」と「障害物の穴」は，ロボットの向きの１マス前にあれば，繰り返し「進める」が偽となり繰り返しが終了する。「障害物の山」と「障害物の穴」は，ここではゲーム上の役割に違いはないが，後程紹介するロボットのゲームでは，違いが出てくる。

「黄ワープ元」・「黄ワープ先」・「青ワープ元」・「青ワープ先」・「赤ワープ元」・「赤ワープ先」では，それぞれの色のワープ元からワープ先に瞬間移動する。

図６が，デザインソフトで作成したパズル面の csv ファイルである。この面は，csv ファイルとしての図４のパズル面であり，流れ図ソフトのデータとして読み込まれて処理されるファイルである。

図６　パズル面の csv ファイル

4	0	4			
5	10	5	51	10	5
-10	10	10	10	10	51
10	10	10	10	10	5
10	10	10	10	10	10
10	10	10	10	10	5
1	10	5	51	5	5
すべての青玉を緑玉に変えて、動きを終了してね。					
5	10	5	53	10	5
-10	10	10	10	10	53
10	10	10	10	10	5
10	10	10	10	10	10
10	10	10	10	10	5
1	10	5	53	5	5
0	1	0	-1	-1	-1
0	3	0	-1	-1	-1
0	7	0	-1	-1	-1
1	0	0	-1	-1	-1
1	0	0	-1	-1	-1
2	2	0	-1	-1	-1
0	0	-1	-1	-1	-1
0	0	-1	-1	-1	-1

１行目が面番号で，ここでは４面目となる。

２～７行目が，提示されるパズル面８×８の内側６×６の面である。パズル面はいずれも周りが「障害物の山」で囲まれているとしているので，内側６×６だけのデータで充分とした。

８行目の「すべての青玉を緑玉に変えて，動きを終了してね。」の文字列は，４面目パズル面をクリアさせるための指令である。パズル面ごとに指令を変えることが可能となるように，csv ファイル内に各パズル面のそれぞれの面達成指令の文字列部分を用意した。

９～14行目が，４面目パズル面の指令達成した結果の面６×６の状態である。流れ図の実行終了時に面の状態をこの値と比較し，同じであれば面クリアとなる。

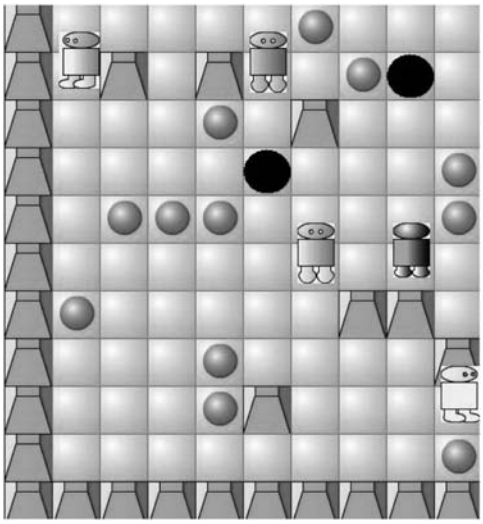
15～23行目の８行が，４面目のパズル面で使用可能となっている流れ図記号の情報である。上から下に，処理記号の「進む」・「左向く」・「緑玉に」，前判定繰り返し記号の「進める」・「進める」，判断記号の「青玉？」で，下２行は空である。

図５のパズル面のデザインソフトを使えば，図６の csv ファイルの２～７行目，９～14行目，15～23行目が作れる。したがって，アルゴリズム学習者にもパズル面を容易に作成することができるので，彼らにパズル面を新たに創造させることで，より深い学習効果が実現できる。実際，パズル面のいくつかは，学生が作成したものである。

３．ロボットのプログラム対戦ゲーム

プログラミングでロボットを動かすゲームで，パズルゲームではないが，同じ面上で複数のプログラムで複数のロボットを同時に動かして，赤玉を取り合って競い合うゲームを紹介する。

図７　ロボット赤玉対戦



このゲームは図７のように，５体のロボットがそれぞれ赤玉を奪い合うゲームで，最初の画面に，各ロボットの向きと位置，赤玉の位置，障害物の山と障害物の穴が乱数によって表示される。

そして，各ロボットをコントロールする流れ図は，図８の左側のようなメニューから，ドラッグにより右側にコピーさせて作成する。ここでは，パズルゲームのような左側の流れ図記号の移動ではなく，流れ図記号のコピーとしていくつでも右側の流れ図に追加できる。

このメニューでは，連接の流れ図記号の処理記号が４種類，判断の流れ図記号の判断記号が８種類用意されている。そしてその他の流れ図記号では，関数（サブルーチン）の働きをする定義済み命令の定義済み記号と，実行を制御するための飛び越し記号が用意されている。

このゲームの流れ図記号は，今までと異なり，構造化プログラミングの繰り返しは使用しない。その替わりに飛び越し記号を使った，関数の単位での繰り返しを実行する。

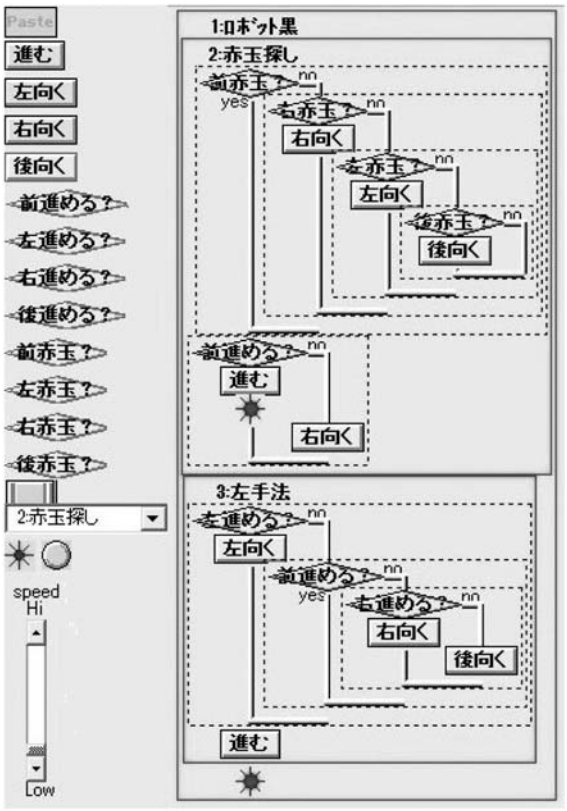
繰り返しの実行は，関数の最初に実行を移動させる飛び越し記号を用いる。

飛び越し記号は，図８の中の左側中の「2：赤球探し」の文字列の下の星印のマークである。星印は水車の形で，この飛び越し記号により実行の流れが関数の最初に飛んで行くことで，実行の流れがクルクル回る様子を表している。

図８の，入れ子になっている「1：ロボット黒」と「2：赤球探し」と「3：左手法」は，それぞれが関数名であり，それらは作成済みの関数として記憶媒体に記憶されている。そして，各ロボットの流れ図は，記憶媒体に読み書きすることができる。また，記憶されている流れ図は，５体のどのロボットにも読み込むことができる。

図８は，黒色のロボットに関数「1：ロボット黒」を記憶媒体から読み込んでいる。関数「1：ロボット黒」の内容は，関数「2：赤球探し」と関数「3：左手法」からなる。

図８　対戦ロボットの流れ図



関数「1：ロボット黒」の実行で，まず関数「2：赤球探し」で，前直線上のどこか・右直線上のどこか・左直線上のどこか・後直線上のどこかに赤玉が見えれば，その方向に進んでいって赤玉を取る。なお，赤玉とロボットの間に「障害物の山」があれば赤玉が見えず，「障害物の穴」だけであれば赤玉が見えるとする。

「障害物の山」と「障害物の穴」があってその方向に進めなくなれば，関数「2：赤球探し」から関数「3：左手法」に実行を移す。

関数「3：左手法」の実行で，順に１マス左，前，右の順に進めるかどうかを調べ，進めればその方向に１マス進む。

４．ROBOTWAR

プログラムで動くロボットの対戦型ゲームの私のルーツは，図９の1981年発売のミューズソフトウェア社制作のアップルⅡ用ゲームソフトの「ROBOTWAR」である。

このソフトは，ミニフロッピーで販売され，当時の月刊ソフトの人気雑誌，徳間書店「テクノポリス」1983年９月号に，一般応募したロボット達の戦いの実況中継と結果報告が載ったほどの人気であった。

ロボットの対戦は最大５体のバトルロワイヤルで，５体のロボットの対戦画面は図10の

図９　ROBOTWAR[1)]

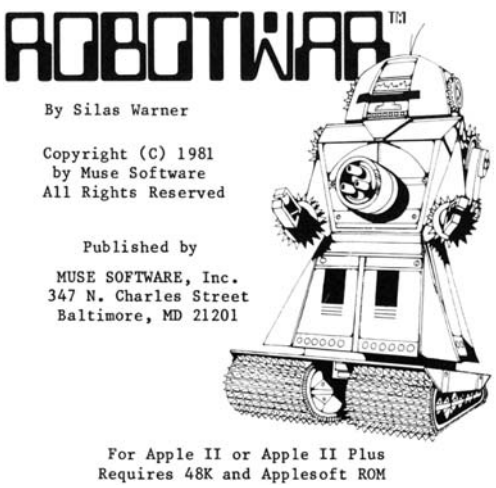
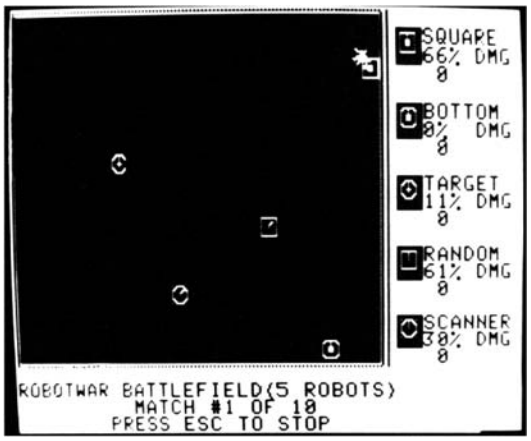


図10　対戦画面



とおりである。

図10は，「SQUARE」・「BOTTOM」・「TARGET」・「RANDOM」・「SCANNER」と名付けられた５体のロボットの対戦の様子である。

各ロボットは，他のロボットとの距離と角度を感知する，360度回転するレーダーを持つ。

そのレーダーで他のロボットを感知すると，取得した距離で爆発する爆弾をその角度で発射し，他のロボットにダメージを与えるのである。

ダメージが100％になると，そのロボットは消滅し画面から消え去る。消滅する順位で

1）図９～11は，ROBOTWAR のマニュアルから抜粋。

図11　ロボットのプログラム

```
];    SAMPLE ROBOT
]
]SCAN
]  AIM + 5 TO AIM           ;MOVE GUN
]  AIM TO RADAR             ;SEND RADAR PULSE
]
]LOOP
]  IF RADAR < 0 GOSUB FIRE  ;TEST RADAR
]  GOTO SCAN
]
]FIRE
]  0 - RADAR TO SHOT        ;FIRE THE GUN
]  ENDSUB
```

ロボットの順位が決まる。

各ロボットの移動は，x 座標と y 座標への-255～255のスピード値の入力で行う。ロボットは，x 座標と y 座標へのスピード値になるまで，加速あるいは減速を続けて移動する。スピード値 0 で停止であるが，うまく停止しないと四方の壁にぶつかり，そこでもダメージを受ける。

レーダーは，壁の場合は壁との距離を，他のロボットの場合はマイナスの値で他のロボットとの距離を与える。

サンプルロボットのプログラムは図11のとおりである。

図11のプログラムでは，まず「SCAN」のラベルから，「AIM ＋ 5 TO AIM」の命令により，砲台の角度である AIM の値を 5 度ずつ時計回りで角度を変える。

さらに「AIM TO RADAR」の命令により，レーダーに砲台と同じ角度を入れて他のロボットの位置を探る。

次に「LOOP」のラベルに移り，「IF RADAR ＜ 0 GOSUB FIRE」の命令により，レーダーを調べてその角度に他のロボットがいれば「FIRE」のラベルに実行を移し，いなければ次の「GOTO SCAN」の命令に実行を移し，再び先頭の「SCAN」のラベルの命令に戻ってプログラムを繰り返す。

「GOSUB FIRE」の命令により「FIRE」のラベルの命令に実行が移ってきたら，「0 - RADAR TO SHOT」の命令で，マイナスである RADAR の値をプラスに反転して SHOT に代入し，AIM の角度と SHOT の距離で爆弾を発射する。

その後「ENDSUB」の命令で，「GOSUB FIRE」の命令の次の「GOTO SCAN」の命令に戻る。

以上のように，各ロボットの動きをプログラミングして戦わせるゲームである。

場面の初期状態が乱数で構成されるので，複数回の対戦による各順位の取得得点の平均値でプログラムの優劣を決定する。

このゲームは，プログラムを作成した後，ただプレイ画面を見るだけで，プレイ中はロボット操作ができなくなっている。そのためプレイ中は，自分のプログラムで動くロボットに対して，あたかもプロスポーツの贔屓チームを見守るがごとく，声援するのみである。

そしてその結果，より一層プログラムを改良することに高いモチベーションが上がる。

5．お　わ　り　に

ROBOTWARの面白さと，プログラミングの楽しさに感銘を受けた私は，当時，人気絶頂の任天堂のゲーム機，ファミリーコンピュータのゲームソフトとならないかと色々と模索した。その結果，PALPS（PAnel Language System）というプログラミングシステムを考案した。

PALPSは，プログラミング言語をパネルという名のアイコンとした視覚言語で，その組み合わせでロボットをプログラム操作する。PALPSは，ROBOTWARに似た種類の複数ロボット対戦型のプログラムゲームのためのプログラミングステムである。

PALPSのゲームは発売されなかったが，その概要を新たにVILLA（Versatile Icon Logic LAnguage）としてまとめ，1993年「VILLAシステム」として嵯峨野書院から出版した。

そして，視覚言語によるプログラムに興味を持ち，公式で有名なロジックの視覚表現である流れ図をプログラミング言語とすることに思い立ち，流れ図がそのままプログラムとなるアプリケーションソフトのLogic Masterを設計・作成した。さらに，そのシステムをアルゴリズムの授業に利用できるように構成した，1995年「構造化プログラミングのアルゴリズム」を同じく嵯峨野書院から出版した。

以後，流れ図ソフトとして現代に至っている。

アルゴリズムの授業を，独自のソフトを使って行ったことへの受講生の反応を確かめたところ，「変数の変化→ロボットの動き　動きの順番→命令を実行する順番ということが考えやすく作られてるので，とてもいいソフトだと思います。……」などの好意的な回答を数多く得ている。

私をROBOTWARが魅了し，加えてプログラミングの面白さで虜にしたようなゲームシステムを目標とし，これからもそのようなシステム作成を目指す。

参考文献

[1]　太田幸一：「アルゴリズム学習用ソフトの開発」，教育システム情報学会研究報告 p. 158-161（2007-3）

[2]　太田幸一：「プログラミングゲーム」，ゲーム学会第5回合同研究会研究報告 p. 3-4（2007-7）

[3]　太田幸一：「アルゴリズム学習の教材」，平成19年度大学教育・情報戦略大会　私立大学情報教育協会 p. 114-115（2007-9）

[4]　太田幸一：「プログラミング学習のためのパズルゲーム」，ゲーム学会「ゲームと教育」研究部会研究会報告 p. 5-7（2013-3）

[5]　太田幸一：「アルゴリズム学習のためのアプリケーションソフト」，大阪経大論集 p. 53-66（2014-9）

[6]　太田幸一：「流れ図ソフトのシステム構造」，大阪経大論集 p. 7-19（2016-1）