

対戦型ゲームによるプログラミング導入教育システム 「アルゴロボ」

小西 誉†, 野口 翔平†, 紫合 治†,
東京電機大学 情報環境学部

1. はじめに

近年、急激に発達し続ける情報技術に合わせ、システムを扱える多くのITエンジニアが求められている。また、それに合わせ、小学校でのプログラム授業の必修化が決定した。それらに先駆け、ブロック型言語を使ったプログラミング学習^{[1][2]}の機会が増えてきている。そこでそれらのブロック型言語にゲーム性を追加することで、初学者にも「分かりやすく」、そしてゲーム性を採用した「楽しい」プログラミング導入教育を目的としたツールの開発・評価を実施した。

2. システム構成

ブロック型言語を使ったプログラムで、画面上のロボットを動かして敵を倒すための試行錯誤を繰り返すことで、アルゴリズムへの理解と興味を喚起させることが本システムの目的となる。

システム構成を図1にて図示する。当システムは主に簡単なコマンドを組み合わせるプログラミングを行うプログラミングツールと、実際にロボットを動かして戦わせるゲーム実行画面の2つで構成されている。プログラミングツールでは、コマンド合計10種類を組み合わせるプログラムをドラッグ&ドロップで作成できる。ゲーム実行画面では、前述のプログラミングツールを使って作成したプログラムをロボットごとに割り当て、開発者側で用意した課題を解き、ユーザー同士で作成したプログラムを競わせることが可能である。

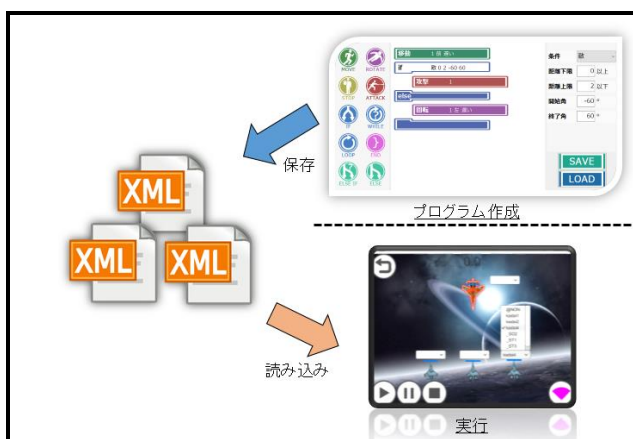


図1 プログラミングツール画面

"AlgoRobo": A programming education system by battle game creation

†Homare Konishi, †Shohei Noguchi and †Osamu Shigo,
School of Information Environment, Tokyo Denki University

3. システム概要

3.1 ブロック型プログラミングツール

図2にて図示されるブロック型プログラミングツールでは、実行画面にてロボットを動かすためのプログラミングを作成する。画面左のアイコンをドラッグ&ドロップにて中央エリアに移動させると、プログラムが記述される。一度記述されたプログラムは、自由に移動や複製、削除することが可能である。

本ツールはビジュアル型プログラミングツールであり、プログラムは「コマンド」と呼ばれる実際のプログラムの記述をひとまとめにして簡略化させたようなものを最小単位として使用する。コマンドは「行動系」と「条件分岐系」の2つに分類される。「行動系」のコマンドには、ロボットを指定方向に移動させる「移動」、その場で自分を軸に回転する「旋回」、指定時間中プログラムを停止させる「停止」、前方向に向かった弾丸を連続で発射する『攻撃』の4つがある。『条件分岐系』のコマンドには、指定条件を満たすかどうかで分岐する「if」、指定回数だけ内部のコマンド群を繰り返す「Loop」、条件を満たす間ループし続ける「while」、「if」が否定された時の分岐先に使用できる『else』、それに条件文を追加した「elseif」、そして条件ブロックの終了を示す『』の6つがある。

なおプログラムは最後のコマンドの処理が終了するとプログラムに先頭にジャンプし、ループし続ける。実行時にはひとつのコマンドのみ実行可能で、同時に2つ以上のコマンドが実行されることはない。プログラムに追加したコマンドをクリックすると画面右に引数を受け渡すエリアが出現する。ここではロボットをどの程度移動させるかの調整や、if の分岐条件を指定することが可能。移動、旋回、停止の時間は frame 数にて指定する。Ifなどの条件文として使えるのは「ある扇形範囲上に、オブジェクト(敵、味方、壁、弾のいずれか)が存在するかどうか」である。扇形範囲の作成にはロボット前方を 0 度として、開始角と終了角を指定する。

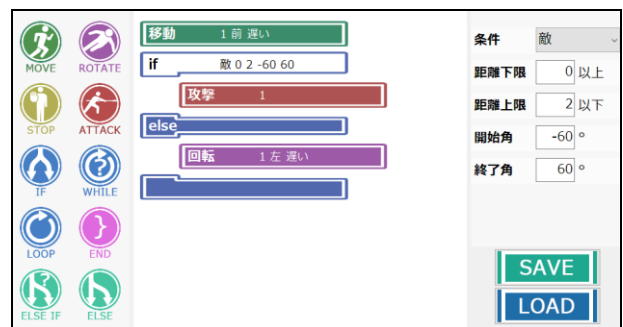


図2 プログラミングツール画面

3.2 ゲーム実行画面

図3にて図示されるゲーム実行画面では、プログラミングツールを使って作成したプログラムをゲーム開始前にプルダウンメニューにてロボットごとに割り当て、開発者側で用意した課題で敵を倒すことが目的となる。機体にはそれぞれに耐久力(HP)が設定されており、『攻撃』コマンドで放った弾に被弾すると減ってゆき、0になると行動不能となる。敵機体すべてのロボットの耐久値をゼロにすることで課題達成となる。課題は、コマンド3つ程度でクリアできる簡単なものから、しっかりとプログラムの構造理解をした上で条件文を多用しなくてはクリアできないものまで、6種類のものを用意した。実行中はいつでも一時停止ボタンで戦闘を止めることができる。また、右下のボタンを押すことで条件文に指定された扇形範囲を任意で表示することも可能なので、プログラムの問題点をいち早く発見できる。課題を解くだけでなく、ユーザー同士で作成したプログラム同士を競わせることも可能である。

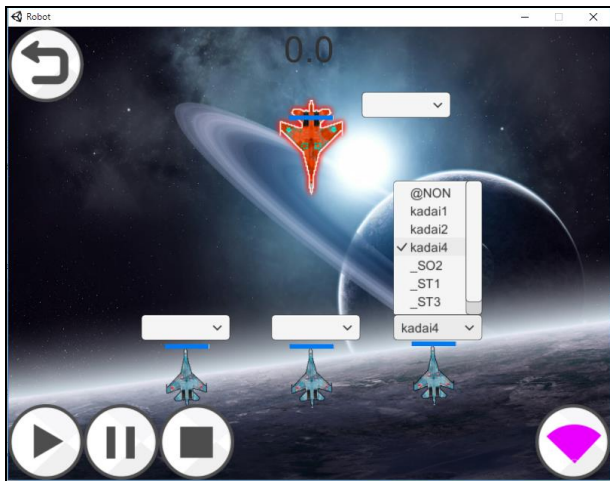


図3 実行画面

4. 評価

4.1 評価方法

本学生 10 名を対象に本システムを一時間ほど実際に使ってもらい、全9問のアンケート調査を実施した。以下の質問に5段階で評価してもらった。質問内容は以下である。

- Q1.プログラミング学習への興味が出たか
- Q2.ツールは使いやすかったか
- Q3.プログラムの構造(分岐, 繰り返し)についてこれを使うことで理解できたか
- Q4.ゲーム内容は理解しやすかったですか
- Q5.課題の難易度は適切だったか
- Q6.Java や C 等の導入教育として適切か
- Q7.既存のプログラミング授業と比べて、楽しく学習できたか

Q8, Q9 ではシステムを触ってみた感想と、使いづらい点などを自由記述してもらった。

4.2 評価結果

表1のアンケートの結果を示すように、概ね評価の平均値が中央の3を超える結果となった。Q4「ゲーム内容理解」とQ7「楽しさ」には相関があり、元々ゲームを遊ぶ学生は内容理解が早く、楽しく学習できたが、その逆にあまりゲームをやらない学生は「内容理解」「楽しさ」の項目評価も低い傾向にあった。

Q3「条件分岐の理解」についての項目は評価平均が高く、また分散も小さいことから本ツールがアルゴリズム教育ソフトとして一定の効果が認められると予想される。

また、唯一平均値が3を下回っているQ2『使いやすさ』について、自由記述欄にて問題が指摘された。現在、コマンドのオプションにはそのコマンドをどの程度の時間実行するかFrameを指定しているが、これに関して「時間(Frame)ではなく、移動量や角度で指定したい」というコメントが多く、Q2 の評価値が低い原因と思われる。

質問	内容	S1	S2	S3	S4	S5	S6	S7	S8	平均	分散
Q1	興味	4	3	4	3	3	5	4	5	3.9	0.70
Q2	使いやすさ	4	2	2	3	3	3	3	3	2.9	0.41
Q3	構造理解	4	5	4	3	2	4	4	4	3.9	0.79
Q4	ゲーム理解	4	2	5	3	2	5	2	5	3.5	2.00
Q5	課題難易度	3	3	5	3	1	5	4	3	3.4	1.70
Q6	導入教育	3	4	2	3	2	4	3	3	3	0.57
Q7	楽しさ	4	2	4	5	3	5	5	5	4.1	1.27

表1 評価項目

4. おわりに

ブロック型プログラミングツールで仮想上のロボットを動かすことで、「楽しく」、そして「わかりやすい」教育支援ソフトウェアの開発、そして本学生を対象にしたシステムのアンケート評価を行った。

ツールの評価の結果を分析し、プログラムの構造理解をする上での有効性と、開発動機のひとつである「楽しい」学習ツールとしてユーザーが利用できることを確認した。

しかしそれと同時に、コマンドオプションの指定方法と条件文の扇形範囲指定について、ユーザーの不満に繋がる課題点が浮き彫りになった。

参考文献

- [1] MIT メディアラボ:Scratch
<https://scratch.mit.edu/>
- [2] ARTDINK:Carnage Heart EXA
<http://www.artdink.co.jp/japanese/title/che/>