

シューティングゲームに特化したプログラミング言語

Connect STG の設計と評価

岸本 有生†

兼宗 進‡

大阪電気通信大学高等学校†

大阪電気通信大学‡

1. はじめに

シューティングゲーム(以下 STG)を通してプログラミングの基礎が学習できる環境 Connect STG を提案する。次期学習指導要領[1]の「情報 I」では、プログラミングが含まれており、授業では、興味・関心を惹きつけ、個々の能力に合った課題が必要となってくる。STG は、学習者になじみの深いコンピュータの一例であり[2]、学習者にとって興味が持ち易い題材である。さらに、理解している者には弾幕 STG といった難しい課題も取り組むことができる利点がある。しかし、ゲーム制作は、従来の言語を使用するとアルゴリズムが難しく理解が困難であった。Connect STG では、その問題点をシステムがサポートすべき部分として整理することで解決している。STG を題材にすることで、ゲーム画面の座標系、プログラムの変数、制御構造、変数、オブジェクト指向などのプログラミング要素を理解でき、多様な作品に発展させることが可能である。

2. Connect STG の言語的な工夫

従来の言語を用いてゲームを制作した場合、表 1 のような問題点が挙げられる。これらは、プログラムの記述ミスを生じ動作確認をするだけで多くの時間を取られてしまう。Connect STG では、プログラムの計算過程が直接結果に表れるものだけを記述する形を採用しており、先程のような問題点は、システムが内部的に自動的に処理してくれる事により解決している。

表 1 従来の問題点と解決方法

番号	問題	解決方法
1	ゲームループ	ゲームループの自動化
2	ソースコードが長い	動作のみを記述
3	キャラクターに配列	クラスを用意
4	出現に線形検索	専用関数を用意
5	当たり判定の記述	当たり判定の自動化
6	関数の引数が多い	引数の省略

Design and Evaluation for Connect STG: Programming language specialized in Shooting Game

†Osaka Electro-Communication University High School

‡Osaka Electro-Communication University

3. Connect STG の設計

(1) プログラムの動き

プログラムは、1 秒間に 60 フレーム (60 回) で動作する。STG は自機、敵機、自機弾、敵機弾が存在し、それぞれのキャラクターにプログラムを記述する。1 フレームのプログラムの流れを図 1 に示す。

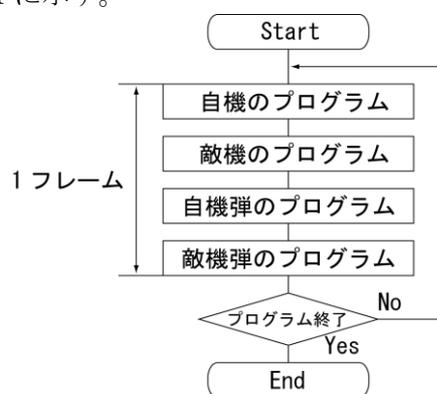


図 1 1 フレームのプログラムの流れ

(2) ゲーム画面

ゲーム画面は、図 2 のように x 軸は 640pixel、y 軸が 480pixel の大きさで構成されている。座標軸は、x 軸が右方向に行くほど数値が大きくなるが、y 軸は下方向に行くほど数値が大きくなることに注意する。



図 2 ゲーム画面

(3) 提案する授業モデル

表 2 には、提案する授業モデルを示している。STG 完成までの時限は、6 時限で構成されており、最初から順番に完成させなければいけない。課題 1~4 は、指導者が示したサンプルを写すような課題ではなく、図や矢印で書かれた行動パターンを学習者に示し、自分で考えさせてコーディングできるような課題を用意する。学習者は、課題の仕様を満たすためにキャラクターを動かしながら考えることで論理的な思考を深められる。自由課題に関しては、学習者の能力に合わせたアルゴリズムを教えることも可能である。

表 2 提案する授業モデル

時限	概要	学習内容
1	自機の移動	変数、条件分岐、座標
2	自機弾の発射 課題 1	タイマー変数、移動量 発射方向変更
3	自機弾の複数発射 課題 2	繰り返し文 弾数の変更
4	敵機の出現 課題 3	フレーム変数 出現位置、方向の変更
5	自機弾の発射 課題 4	オブジェクト指向 旋回位置、方向の変更
6	自機弾の発射 自由課題	狙い撃ち弾 自主作品の制作

4. Connect STG の授業実践

(1) 実践した授業

実際に高等学校にて授業を行った。対象生徒は、普通科 3 年生 29 人と電子工業科 3 年生 22 人である。授業内容は、本来ならば 3 時限かかる内容を、課題 1, 2 を実施しない形として圧縮し、1 時限(50 分間)で行った。理由は、時間が多く取れなかった事と、短い時間でも多くの機能を生徒に体験してもらいたかったからである。次に、自機の移動の学習方法について説明する。

表 3 実践した授業

時限	概要	学習内容
1	自機の移動 自機弾の発射 自機弾の複数発射	変数、条件分岐、座標 タイマー変数、移動量 繰り返し文

(2) 自機の移動

図 3 は、上キーを押した時、自機が上に移動するプログラムである。指導者が、図 3 のプログラムを説明し動作を確認する。そして、学習者に図 4 のような、下の場合は DOWN()、左の場合は LEFT()、右の場合は RIGHT() を使用したプログラムを追加させる。学習者は、上に移動する時は、dy を減少させなければいけないが、下に移動する時は、dy を増加させなければいけないことを知りゲームの座標系を学習できる。

```
01:if(UP()){
02:    y=y-dy;
03:}
```

図 3 自機が上に移動するプログラム

```
01:if(DOWN()){
02:    y=y+dy;
03:}
04:if(LEFT()){
05:    x=x-dx;
06:}
07:if(RIGHT()){
08:    x=x+dx;
09:}
```

図 4 自機が下左右に移動するプログラム

5. 評価

授業後、生徒に対してアンケートを実施した。それぞれ、5 段階に分けて評価してもらい、表 4, 5 には、平均値と評価 4 以上の割合を示している。ここでは、評価 4 以上の生徒は、質問に対して肯定的な意見を持っているとする。普通科では、キーボード操作に慣れずに「難しい」と答える生徒が多かった。そして、全体を通して言えることが、「楽しい」と感じている生徒の割合が多いことである。実際に、「自機弾を斜めに発射させる」、「タイマーを減らしてレーザーに変更する」、「for 文の回数を多くして自機弾を多く発射させる」など自ら能動的な学習を行った生徒も現れた。

表 4 授業アンケート (普通科)

	平均 M	評価 4 以上の割合
楽しさ	3.72	68.97%
使いやすさ	3.03	27.59%
興味	3.03	31.03%
難しさ	3.72	65.51%

表 5 授業アンケート (電子工業科)

	平均 M	評価 4 以上の割合
楽しさ	4.46	90.91%
使いやすさ	3.64	45.45%
興味	3.86	63.64%
難しさ	3.14	22.73%

参考文献

[1] 文部科学省：高等学校学習指導要領, 入手先 (http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm) (2018).
 [2] 長瀧 寛之：コンピュータゲームを通して情報科学を概観する一般情報教育の授業手法の提案と評価, 情報処理学会論文誌, Vol. 54, No. 1, pp. 2-13 (2013).