初めてのプログラミング

理数科2年　　黒土　照久　　末松　尚史
高瀬　浩行　　田中　千裕

1．主題設定の理由

私たちが「プログラミング」について研究するきっかけとなったのは、数学の授業中の余談だ。授業中に少し触れた「プログラミング」に興味を持ち、詳しく調べたいと思ったからだ。身近にあるコンピュータゲームも「プログラミング」を基盤としていることを知り、自分たちでも簡単なゲームのプログラムを作れるのではないか？そもそもゲームの基盤とは何か？このような理由から私たちはこの課題研究の主題を「プログラミング」に設定した。

2．目的

コンピュータゲームの基礎である「プログラミング」について研究し、ゲームを形作っているものを突きつめること。また、実際に簡単なコンピュータゲームを作成し、研究した内容を実践すること。

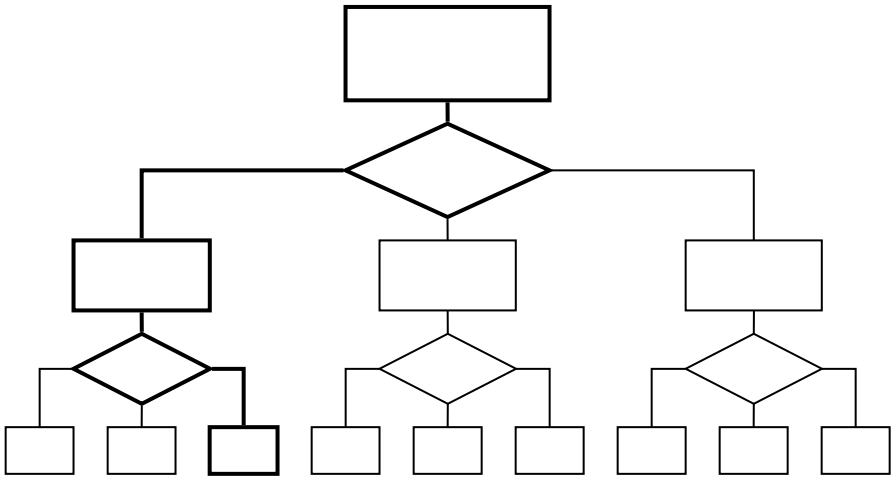
3．仮説

私たちは、コンピュータゲームを形作っているものだから「プログラミング」とは複雑で、記号もたくさんあると推測した。

4．プログラミング

私たちが使用したのは「HSP」というプログラミング言語である。これはインターネット上の公式ホームページで無償ダウンロードが可能であり、2005 年 12 月現在ではバージョン 3.0 が配布されている。しかし、今回私たちが使用したのはそれよりも古いバージョンであるため、一部のプログラムが Ver.3.0 では実行できない。「BASIC」に似ているところもあり、それを扱ったことのある人には、下に示すプログラム紹介はいくか分かりやすいかもしれない。

私たちは当初、「コウ」という主人公を操るアドベンチャーゲームを作ろうとした。



上にそのゲームのプログラムの流れを模式的に示した。
このように一度条件分岐をしてストーリーをいくつかに分けた後、そのそれぞれのストーリーでまた条件分岐をする。この流れを繰り返し最終的に様々なエンディングの内の一つにたどり着くというものである。
このプログラムの組み方には次に示すような利点がある。

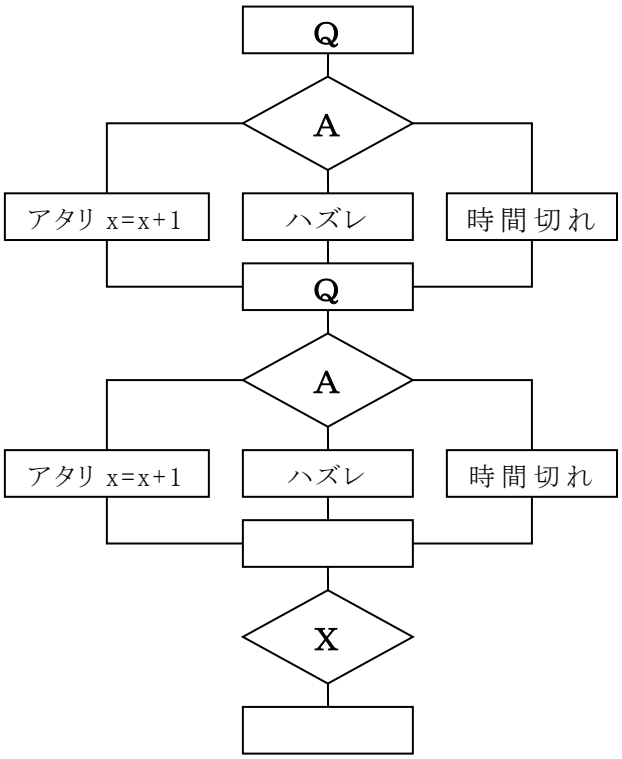
- 繰り返しが基本なので、一パターン作れば後はその繰り返しで容易に流れを作ることができる。
- 変数、「if～:」を用いずに条件を分岐させることができる。

変数、「if～:」やその他のプログラムは後ほど説明をする。
しかし、上に「作ろうとした」とあるように、私たちはこのゲーム作成を断念した。それは次に示す欠点からである。

- 一つの分岐から最低二つのストーリーができるため、プログラムが進むにつれて多くのストーリーを作る必要がある。
- 発表時間内にはプログラムのすべてを見ることができない。

これは大変きついものだった。まずプログラムを書く手間が2倍、4倍、8倍（もしくはそれ以上の増加率で）と増えていくのである。そしてそれに伴ってストーリーも同じ数考えなければならないのだった。当然、それほどのストーリーの構成力が私たちにあるはずも無かった。また例えば太線のストーリーを見ていった場合、他の細線の部分にはまったく触れないことになる。これではせっかく作ったプログラムのすべてを見せることができないので、製作者としても何か悲しいものがあった。
これらの欠点から、私たちは一度プログラムを作り変えることになった。

そして作ったのが次のようなものである。



私たちは「コウ」の欠点を踏まえて「QUIZ」というゲームのプログラミングに取り掛かった。
これはその名の通りクイズゲームである。

はじめに、Qのところで出題する。そのときに使用するのが文字表示のプログラムである。次にその例を示した。
まず、①～①´の部分で文字表示の性質を決める。この例では文字のフォントが「HG丸ゴシックM－PRO」、サイズが「25」、太字、アンチエイリアス(文字の形を滑らかにする)有となっており、その指定した形の文字を「100」という速度で一文字ずつ表示することになっている。
その次の②～②´の部分に表示させたい文字を書く。ここでは「　」(空白)も一文字として数えるため、一行の後に空白を三つ書き、一行表示して次の一行を表示するまでの時間を調節した。
この①～①´、②～②´のプログラムは、ともに「コウ」にも使用した。

```
①       font"HG丸ゴシックM－PRO",25,17
        text 100
        repeat 1
                mes
①´      loop

②       mes"Q1      "
        mes"新撰組の三番隊組長はだ～れだ？      "
        mes"A. 斉藤一      "
        mes"B. 井上源三郎      "
        mes"C. 藤堂平助      "
②´      mes"D. 原田左之助      "
```

その次のAの部分では、Qで表示した答えA～Dの内の一つを選択する。そのプログラムは下に示す例のようになっている。

```
③       button"A",*t1
        button"B",*f1
        button"C",*f1
③´      button"D",*f1
④       snd 0
⑤       wait 500
        sndoff : picload "times up.bmp"
        button"Q2へ",*q2 : stop
*f1
        snd 1 : picload "×.bmp"
        button"Q2へ",*q2 : stop
*t1
        snd 2
        x=x+1
        picload "正解.bmp"
        button"Q2へ",*q2 : stop
```

③～③´の部分では答えを選択するためにクリックするボタンを表示する。「”　”」内

にはボタン上に表示させたい文字を書くのだが、三文字程度しか書くことが出来なかった。その後の「*…」は、そのボタンをクリックするとどこのプログラムに飛ぶかを表しており、上の模式図において「t」は「アタリ」、「f」は「ハズレ」である。
④は音楽を鳴らすプログラムである。
⑤はプログラムの進行を一時止めるプログラムである。これにより、一定時間次の行のプログラムに移らないので、③～③´のボタンを表示させたままに出来る。この時間内にボタンをクリックしなければ次の行のプログラムである、模式図内の「時間切れ」に進む。

「アタリ」「ハズレ」「時間切れ」の三つはどれも、基本は「picload」というプログラムを用いて画像を表示するのだが、「アタリ」のみはそれに加えて「x=x+1」と書き込んである。これが「コウ」の話で書いた「変数」である。一見するとおかしな式であるが、これは「x を x+1 と置き換える」と考えてもらいたい。
③～③´は「コウ」でも使用した。

ここまでの流れを何度か繰り返した後、模式図のXに行き着く。ここでは先ほどの「if～:」を用いる。
「if～:」は、「if」以下の「～」に条件を書き、その条件を満たしていれば「:」以下のプログラムを実行するというものである。上の⑥～⑥´に使っているが、その条件は「x の値」である。前記した「x=x+1」を思い出してもらいたい。これは「x を x+1 と置き換える」と説明したが、ではその置き換えを繰り返した場合 x の値はどうなるのか。少し考えてみればわかると思うが、x の値が1ずつ増えていくのである。これを「アタリ」部分のみに書いた場合、x は正解数を示すのである。そして上のプログラムでその正解数を判断し、それぞれの文章を表示するのである。
なお、「pos ○,○」は文字やボタンなどの表示場所を示す。

```
                （文字性質を決めるプログラム、その他を書く）
⑥      if x=0 : pos 50,300 : mes″   残念0点″
       if x=1 : pos 30,300 : mes″   頑張れ1点″
       if x=2 : pos 10,300 : mes″   まだまだ2点″
       if x=3 : pos 0,300 : mes″    あと一歩だ3点″
⑥´     if x=4 : pos 0,300 : mes″    4点満点！！″
       pos 570,450 : button″FIN″,*fin : stop
```

このプログラムの組み方の場合、次に示す利点がある。

- 「変数」「if～:」を使うため、「button“ ”,*」のみの条件分岐よりもプログラム全体が見やすくなる。
- 一度分岐してもまた一つに戻るので、プログラムの無駄が少なくなる。

この組み方をしてみてわかったのだが、単純そうに見える「ボタン分岐」のみを何度も繰り返すと、実は余計複雑になってしまうのである。また、最後の点数表示のときにはボタン分岐では不可能だと思う。そして「分かれて戻る」を繰り返す「QUIZ」の組み方では、全体に一本の筋があるようで、「どのようにプログラムが進んでいくのか」、その流れをつかみやすく出来るようになった。
欠点としては、

- 少し見ただけではわかりづらい「変数」「if～:」を使用した。

が、一応はあるのだが、上記のようにその利点のほうが極めて大きかったと思う。

5. まとめ

私達はプログラミングについて学んでいく中で、市販のゲームやソフトを一本作ることがどんなに大変か、よく分かった。また1つでも命令が間違っていれば、正常に動かないので、とても根気のいる作業だった。今回製作したプログラムはとても簡単なものではあったがこれを製作する中で、様々なプログラムを組み合わせれば、数限りないプログラムができるので、無限の可能性を秘めていると私達は思い、より一層興味や関心を深めることができた。

黒土:今回初めてプログラミングを組み、ソフトを使って自分達が出す命令でいろいろなことが出来てとても驚きました。今回、自分達の班は発表できませんでしたが、今回の研究を通していろいろなことを学ぶことができました。このことをこれから活かしていけばいいなと思います。

高瀬:私は初めてプログラミングに取り組み、初めは基本的なことを学ぶことだけでもとても難しそうだと思っていました。しかし実際にプログラミングについて理解していくにつれて、規則的なことの連続だということが分かりました。また次第にとても奥の深いものだと思うようになるとともに、もっと詳しく学び今後さまざまな場面で生かしていきたいと思いました。

末松:この報告書では、課題研究の期間中に書いたプログラムの全てを載せることができませんでしたが、それでも伝えたいところをなるべく多く紹介することができたと

思います。選考に落ちた原因である「わかりづらさ」「マニアックさ」を解消することを念頭に置きながら、この報告書作成を行いましたが、できてみるとまだ不安なところが多くあります。それでも読んでもらえれば幸いです。プログラム作成中や直後に感じたことは「数学というよりも英語に近い」というものです。それは上に書いたプログラムを見てもらえばわかりますが、英単語の一部や組み合わせのようなところが多くあるからです。別に英語といわずとも何か新しい言語のようにも感じました。だからまるでパソコンと「会話をする」ようにすれば、もっと上手くプログラミングができたのかも知れません。

田中：初めは「プログラミング」という言葉さえ、知りませんでした。コンピュータゲームの基礎と聞き、どんな複雑なものだろうと思っていましたが、実際は私のイメージよりも簡単なものでした。この課題研究で得たものは、将来必ず役に立つものばかりでした。発表こそできなかったものの、班のみんなと素晴らしいものを作ることができました。