

はじめに	2
この教材の使い方	3
感染症予防対策について	3
ハードウェアの扱い	3
入力の間違いとその対処方法	4
プログラミングを学習する流れ	4
1. IchigoJamでプログラミングしよう！	5
2. アニメーションを作ろう	8
3. 数当てゲームを作ろう	9
4. 乱数でいろいろなものを作ろう	10
5. 九九ゲームを作ろう	11
6. ネコをつかまえる	12
7. スカッシュゲームを作ろう	13
8. 的当てゲームを作ろう	14
9. 記憶力の限界に挑戦！	15
10. オリジナルゲームを作ろう	17

はじめに

このプリントは、プログラミング学習を遠隔地よりオンラインで指導できるよう開発されました。プログラミング環境としてIchigoJam（またはIchigoJam web）を用います。

（IchigoJam webについては、「1. IchigoJamでプログラミングしよう！」の「IchigoJam webを使おう」を参照してください）

- ・ 初学者～中級者向けです。
- ・ 内容は、IchigoJamの基本的な扱い方、プログラミングの基礎、簡単なゲーム作成などです。
- ・ 45分授業用 x 10コマ分（小学校低学年は6コマ分）あります。

	タイトル	主な内容	難しさ
1	IchigoJamでプログラミングしよう！	IchigoJamの接続等、プログラミングの基礎、LED命令、WAIT命令、保存方法	低
2	アニメーションを作ろう	PRINT命令、LOCATE命令、表示する座標、アニメーションの作成方法	低
3	数あてゲームを作ろう	変数の概念、IF、条件分岐の概念、INPUT命令、それらを用いたゲーム作成	中
4	乱数でいろいろなものを作ろう	RND命令、乱数の概念、おみくじの作成方法、乱数を用いるサンプルプログラム	中
5	九九ゲームを作ろう	FORを用いる繰り返しのプログラム方法、九九の表示とゲーム作成、TICK命令	中
6	ネコをつかまえる	カーソルキーでキャラクターを動かすプログラムとそれを用いたゲーム作成	高
7	スカッシュゲームを作ろう（※）	画面上でキャラクターを自動的に動かすプログラムとそれを用いたゲーム作成	高
8	的あてゲームを作ろう（※）	中学理科・数学で習う放物線をシミュレートするプログラムとそれを用いたゲーム作成	高
9	記憶力の限界に挑戦！（※）	配列の概念と使用方法、記憶力ゲームの作成、問題を増やすよう改造する方法	高
10	オリジナルゲームを作ろう（※）	ゲームの仕組みの理解、あたり判定、シンプルさの重要性、ゲームのアイデア	超

（※） … 小学校低学年向けは無し

学年に応じて3バージョンがあります。基本的に同じ内容で、漢字とひらがなの使い分け、フリガナ、説明の方法などが異なります。区別しやすいよう、背景色を変えています。

小学校低学年（1～3年生）向けは、「6. ネコをつかまえる」までです。

対象学年	背景色
小学校低学年（1～3年生）向け	青
小学校高学年（4～6年生）向け	オレンジ
中学生向け	緑

この教材の使い方

この教材は、オンラインでの指導に用いることができるよう、学習者・遠隔地にいるプログラミングに詳しい講師・学習者のそばにいてサポートする（必ずしもプログラミングに詳しい必要のない）大人という状況を想定して開発されています。この教材を用いることで、プログラミングに詳しい講師が対面授業を行える環境にない場合でも、プログラミングの指導が行いやすくなります。

以下のような状況でも用いることができ、役に立つと考えられます。

- ・プログラミングに詳しい講師、または詳しくない大人による、対面授業
- ・学習者の側にサポートする大人がおらず、遠隔地からの指導のみで行う授業
- ・学習者による独習
- ・講師1名、またはそれに加えて数名のアシスタントが行う、最大30人程度までの一斉授業（主に対面授業）

オンラインで用いる場合は、学習者があらかじめプリントを印刷するか、iPadなどに表示して、IchigoJamの横に置いて参照しながら進めるように指示してください。

一斉授業を行う場合は、印刷したプリントを各学習者に配布した上で、プリントの内容をプロジェクターに映し、進行に合わせてページを進めるなどしてください。

感染症予防対策について

対面授業の場合、講師は開始前に機材を消毒してください。学習者同士はお互いの機械、特にキーボードに触れないように注意してください。学習者が入れ替わる場合は、その度に機材やテーブルを消毒してください。講師は指導のために生徒のキーボードを触ることがありますが、最低限とし、その前に目や口を触っていないか確認するようにしましょう。

講師、学習者、保護者や見学者などを含めて参加者は全員、入室時に体温を測り、手指を消毒し、マスクを常に着用してください。会場は定期的に換気してください。そのほか、国や自治体および会場の定める感染防止症予防策を厳守して運営してください。

ハードウェアの扱い

学習者には、機械、特にキーボードを大切に扱うこと、講師の許可なくスイッチや電池に触らないよう言い聞かせてください。

IchigoJamは頑丈でほとんど壊れませんが、機械に愛情を持って接する態度が、プログラミングの学習には不可欠です。

他の子のキーボードに触らないことは、感染症予防対策であるのと同時に、他の子の学習機会を奪わないようにする意味合いもあります。

このことは講師が指導する際にも同様で、プログラムを直してあげるときなど、できるだけ本人にさせてください。時間の都合でそう出来ないときは、各行の最後のエンターだけは本人に押させるなど、修正作業を**主体的に行った印象になるよう工夫**してみてください。

入力の間違いとその対処方法

この手引きの「1. IchigoJamでプログラミングしよう！」の「プログラムをみたりなおしたりしよう」をよく読んで、よくある入力間違いの発見方法を身につけておきましょう。

プログラムを打ち終えても動かず、シンタックスエラーも出ない場合は、高確率でエンターキーの押し忘れです。以下のように対処してください。

- ・ F4キーを押してあげてプログラムを表示し、欠けている行があれば、エンターキーの押し忘れである可能性を伝えます。
- ・ 押し忘れた行はおそらくまだ画面上に表示されていますから、上カーソルキーと右カーソルキーで各行の末尾までカーソルを移動させてから、行ごとにエンターキーを押させてあげてください。（実は行末でなくても、行のどこかにカーソルがあればOKです）
- ・ F1を押してしまったなどで、プログラムがもう画面に表示されていなければ、短いからすぐ出来ると励ましつつ、もう一度入力するよう促してください。消えたと聞くとショックを受ける子もいますが、2度目はスムーズに速く入力できることが多いので、打ち直しをし始めさえすれば、すぐに機嫌は直るでしょう。なかなか始められない子には、最初だけ手伝ってあげるか、1行ずつ交互に打とうと提案してみるなど、きっかけを作ってあげてください。

また、うまく行かなかったとき、電源を切って最初からやり直す子がいますが、次はしないように注意してあげてください。デバッグは推理ゲームのようなものですし、直すことこそがプログラミングだったりもします。イライラせず、楽しんで進められるよう誘導してあげてください。

プログラミングを学習する流れ

初学者がコンピュータに触れ、プログラミングの基本を学ぶために、以下のような内容を順に進めます。

1. 命令をひとつずつ入力し、コンピュータの反応を確認します。
2. 短いプログラムを入力し、実行します。
3. プログラムを追加し、実行します。
4. リストを表示して、編集する方法を学びます。
5. 簡単なゲームを作る経験をし、作ったゲームで遊びます。
6. プログラムの実行と停止を制御する方法を体験します。

「1. IchigoJamでプログラミングしよう！」はこれらの内容を踏襲しています。初学者でも無理なく進めることができるでしょう。それ以降の回では、色々な命令を使って行きつつ、段々と難しい仕組みのプログラムに触れ、その考え方を体験的に学んでいきます。

次のページ以降では、各回の狙いやポイントとなる章について解説します。

1. IchigoJamでプログラミングしよう！

主な内容：IchigoJamの接続等、プログラミングの基礎、LED命令、WAIT命令、保存方法



-01-

IchigoJamの操作とプログラミングの基礎を両方学ぶ重要な内容です。この回は注意すべきポイントが多いので、特に詳しく説明します。よく読んでから指導にあってください。

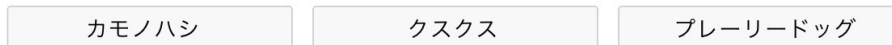
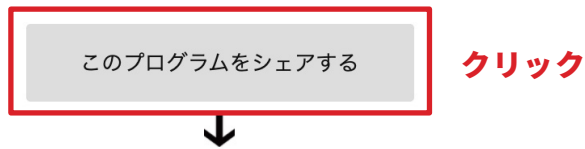
IchigoJamをつなごう

- ・ IchigoJamにキーボードとテレビを繋ぎ、電源を入れて、テレビにIchigoJamの画面が表示されることを確認してください。
- ・ キーボードを触って、文字が入力できることを確認してください。
- ・ 人数の多い授業では、大人が準備しておくトスムーズですが、あえて子供にやらせることも良い経験になるでしょう。

IchigoJam webを使おう

IchigoJam webはIchigoJam開発者の福野泰介氏が開発しホスティングしているウェブ上の無料サービスで、IchigoJam実機を用いるのと同様にプログラミングが可能です。

- ・ 実機より少し遅いなど多少の違いもあります。
- ・ 少しコツがいります。授業前にプログラム入力などして扱いに慣れておきましょう。
- ・ セーブしても、タブを閉じればプログラムは消えてしまいます。
- ・ iPad等ではキーボードを英語入力モードにしてください。日本語入力モードでも一見使えますが、入力したプログラムが消えるなどの誤動作を起こします。



どうぶつシェア機能を使うときは、プログラムに個人情報等が含まれないよう注意してください。また「このプログラムをシェアする」ボタンを押したとき、表示された動物を忘れると二度と復元できません。すぐにメモするように指導してください。

いろいろな文字を入力しよう

- ・ 2つのキーを同時に押す操作は、大人の想像よりずっと難しいようです。
- ・ 1つのキーを押しっぱなしにすること、その間にもう1つのキーを押して、すぐに離すこと、その後、1つめのキーをゆっくりと離すこと。この一連の動作を実演して見せた上で、できない子供がいれば、焦らず繰り返し練習するようにしましょう。
- ・ 片手で操作しようとする子供が多いですが、両手を使う方が簡単です。
- ・ キーボードの種類によって、あるいはIchigoJam webを使う時などには、Altキーでなくoptionキーを使うなどの場合があります。適宜読み替えてください。

命令を入力しよう

- ・ 先にやって見せても良いですが、何も分からないままエンターキーを押し、LEDが光ったことに自分で気づく方が学習効果が高いようです。
- ・ LEDが光っても気づかない子供が多くいます。あらかじめ注目させましょう。
- ・ LEDをLID、ゼロをオーと打ち間違える子供が多いので注意してください。

コンピューターは速い!

- ・ 初めて記号を用います。入力方法が分からない子や、間違ってセミコロン等の他の記号を入力する子供がいるので注意してください。

プログラムを作るう

- ・ 画面が消えるとプログラムも消えたと思う子供が多いです。F4キーで戻せると教えましょう。
- ・ プログラムをメモリー上に記憶して実行するという概念は、小学校高学年でも理解が難しいものです。この教材は、それが分からなくても進められます。

この章までが最低限の内容です。次章以降はできない子供がいても気にしない・気にさせないように指導しましょう。

プログラムをみたりなおしたりしよう

プログラムをみたりなおしたりしよう

F5 **LIST** を押してみましょう。今のプログラムが表示されます。

```
1 0 LED1
2 0 WAIT0
3 0 LED0
```

うまく動かないときは、プログラムをよく見なおしてみましょう。1文字でも違ったら、プログラムは正しく動きません。

- 数字の0ゼロをOオーにいませんか？
- 数字の1をIやLにいませんか？
- LEDやWAITのつづりは間違っていないですか？

違うところを見つけたらなおしてみましょう。カーソルキー（上下左右の矢印）を使ってカーソル（黒い点）が表示されている、点滅している四角をその場所まで動かします。**SPACE**を押すと文字が1つつ消えます。そして正しい文字を入力します。最後に **Enter** を押さなければ、なおした内容は記憶されません。忘れずに押しましょう。

ほかにも、次のような間違いが考えられます。

- **F5** **LIST** で表示されない行があるなら、行の最後に **Enter** を押すのを忘れたのかもしれない。もう一度入力しましょう。行を追加するときは、どんな順序で入力しても大丈夫です。**Enter** を押してから **F5** **LIST** で確認すれば、自動的に行番号の小さい順に並びます。
- 10、20、30以外の変な行はありませんか？ いらぬ行があったら、その行番号を入力して、**Enter** を押して消しましょう。もう一度 **F5** **LIST** を押して、ちゃんと消えたかを確認します。
- **Enter** を押さずにスペースをたくさん押して、次の行に移動していませんか？ その場合は、その行をもう一度入力しましょう。

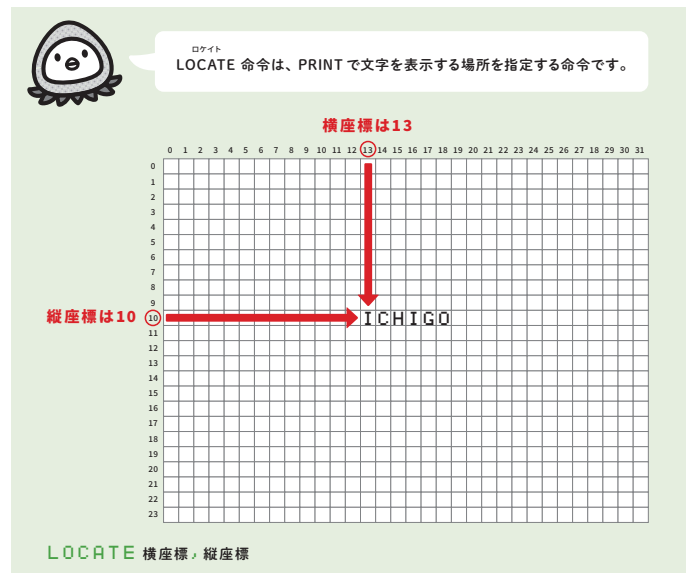
-08-

- ・プログラムは作っても動かないことがあって当然で、直すことも含めてプログラミングです。そう伝えて、うまく動かないことへのネガティブな気持ちを緩和しましょう。
- ・間違いは悪いことではなく、エラーは間違いを探すゲームだ、というくらいの気持ちで、講師自らが楽しそうな様子で指導にあたりましょう。
- ・F5で実行して、シンタックスエラーが表示されたらその行を教材とよく見くらべて、自分で間違いを見つけて直すように手伝いましょう。
- ・シンタックスエラーが出ない場合は、講師が間違いを見つけてアドバイスしてください。その場合、最後の行のエンターキー押し忘れをまず疑いましょう。これが最も多い間違いです。
- ・そのほか、このページの内容をよく把握して、よくある間違いに素早く気付けるように慣れておきましょう。
- ・シンタックスエラー時に複数の行が表示されたら、上の図の一番下の間違いです。

2. アニメーションを作ろう

主な内容：PRINT命令、LOCATE命令、表示する座標、アニメーションの作成方法

LOCATEを使おう



ゲーム作りなどで基本となる座標の概念とLOCATE命令の使い方を学びます。

- ・教材の内容だけでなく、画面のこのあたりに表示するためにはどうしたら良いか、逆に、この命令でどのあたりに表示されると思うか、などをゲームのようにして練習しましょう。

\考えよう！／

答えは『30 LOCATE14,12:PRINT"🍷JAM"』です。しかし正解できなくても構いません。自分で考えてやってみること、それによって結果が変化する体験が重要です。

人を走らせよう

ここで、ちょっとした技を使ってみましょう。

F4 LIST を押してプログラムを表示します。カーソルキーでカーソルを移動して60行の先頭の6を消し、代わりに7を入力して **Enter** を押します。もう一度 **F4 LIST** を押すと、60行が70行にコピーされています！

この技を使って、60行とまったく同じプログラムを70行、80行にコピーしておきましょう。

行をコピーする、BASICの古典的なテクニックです。少し難しく、この後も用いる場面は限られるため、できなければ講師が手伝って進めて構いません。

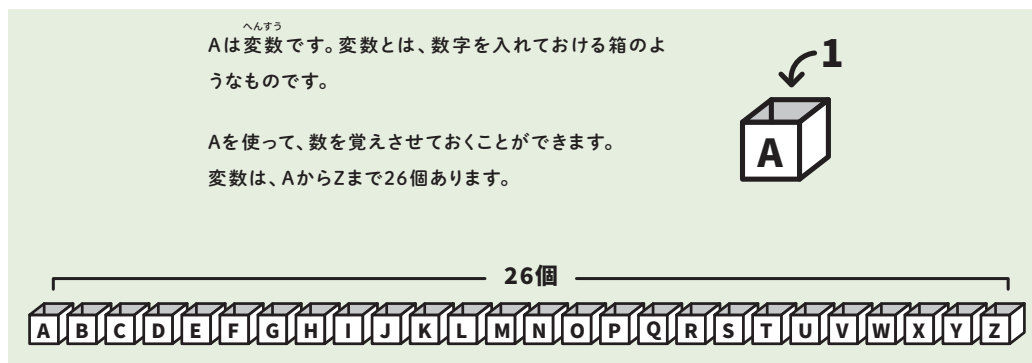
完成したプログラム

時間が余った子には、WAITの数を変えてスピードを変えたり、座標を変えたり、オリジナルの行を追加したりして、自由に遊ばせてあげましょう。自然とプログラム編集の練習にもなります。

3. 数あてゲームを作ろう

主な内容：変数の概念、IF、条件分岐の概念、INPUT命令、それらを用いたゲーム作成

変数を使おう



変数の概念を図で解説しています。何か実際の箱を使って数字を入れておくイメージを説明したり、生徒に変数の役をやらせて数を覚えさせておいて後から聞いたりなど、変数のイメージを掴めるように工夫しましょう。

- ・ IchigoJamでは-32766～32767の整数を扱えます。小数はありません。
- ・ 文字変数はありません。
- ・ 変数への代入と、PRINTで用いるなどの文法は、あまり重要ではありません。変数に数が記憶でき、その数が後から利用できるという概念の体験と捉えてください。

IF（もしも）を使おう

- ・ 今回の主眼は変数です。IFもすぐに登場していますが、すべてをすぐに理解することは難しいので、IFは変数を理解するための体験のひとつくらいに捉えて進めて良いでしょう。
- ・ 特に小学校低学年において、実行時に状態が変わるという考え方は理解が困難なようです。ただし、すんなりと理解できる子もいます。

INPUTを使おう

- ・ ?が表示されても、何をして良いか分からない子が多くいます。先にやってみせると良いでしょう。

完成した『数あてゲーム』遊んでみてね

- ・ 時間が余った子には、何回で正解することができるか挑戦させたり、1回目や2回目に何を入力すると最も効率が良いか考えさせたりしましょう。

4. 乱数でいろいろなものを作ろう

主な内容：RND命令、乱数の概念、おみくじの作成方法、乱数を用いるサンプルプログラム

この回は内容が短めです。1回だけの体験学習や短時間のイベントにも適しています。時間が足りなければおみくじを省略してかわくだりの入力に進むなどして工夫してください。

乱数とジャンケンしよう

この回で**最も重要なのは、乱数という機能を体験的に学習すること**です。終わってから、コンピューターと数字でジャンケンした印象が残っていれば、成功と言えます。



- ・ジャンケンを何度かしたら、0,1,2のうち何が出やすいか、どのくらい勝てるかなど質問して、乱数の印象を深めてください。子供同士で話し合ってもらうのも良いでしょう。

サイコロを作るう

- ・0~5に常に1を足すことで1~6の結果を得るというのは、大人にとっても難しい考え方です。
- ・方法を考えさせても良いでしょう。
- ・0のときだけ6とみなす、6に変える、という解決策を思いつく子がいます。立派なひとつの正解です。褒めてあげた上で、他にも方法があると教材の方法を紹介しましょう。

おみくじを作るう

- ・50行を入力しても、10行を変えなければキョウが表示されることはない、ということに気づく初学者はほとんどいません。
- ・一種のデバッグの体験です。楽しい雰囲気です。推理するように進めてみてください。

オリジナルおみくじを作るう

- ・時間が十分にあれば、オリジナルのおみくじを作って遊びましょう。
- ・時間がない子は、アイデアを出すだけにして、作成しなくて構いません。

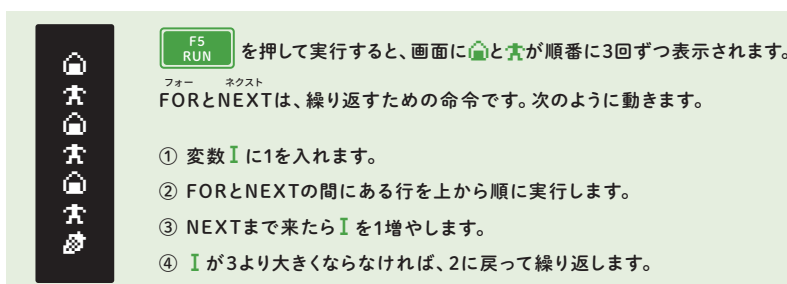
5. 九九ゲームを作ろう

主な内容：FORを用いる繰り返しのプログラム方法、九九の表示とゲーム作成、TICK命令

楽しいゲームですが、やや実用ソフト寄りのプログラムと言えます。また、算数の授業とリンクする内容であるため、算数の授業時間に実施したり、九九の苦手な子にゲームとして遊ばせて九九の習得に役立てるなど、活用できそうです。

繰り返しのプログラム

- ・ 繰り返しまたは反復と呼ばれる、コンピューターの基本機能のひとつです。
- ・ FOR-NEXTの構文とその動作はやや理解しづらいものです。ここではあまり深く考えず、プログ



ラムを正しく入力して、繰り返し機能を体験することを重視しましょう。

- ・ FOR-NEXT構文は、プログラムの各行の実行順序や回数を制御するためのものですが、子供は、表示される文字が3倍になるといったイメージで捉えがちです。詳しく説明してもピンと来ないことが多いので、あまりこだわらず進めましょう。

2の段を作ってみよう

- ・ 20行を変更するだけで、30行の答えも勝手に変わるはず、という誤解がよく見られます。特に小さな子は、問題と答えが合わないのは、コンピューターがおかしいと考えがちです。コンピューターが間違っているのではなく、プログラムが間違っていると説明しましょう。
- ・ 20行を変更したつもりで30行を変更したり、その逆といった間違いがよく見られます。エンターキーの押し忘れも組み合わさってグチャグチャ、いっそ最初からやり直す…という子もいますが、落ち着かせて、手伝って、よく見て直す経験をさせてあげましょう。
- ・ この教材では、ダブルクォーテーションの説明をあまり詳しく行っていません。詳しく説明しても、きちんと理解して自分で考えて使うのはかなり難しいためです。

時間を計ろう

- ・ どんなゲームでも、この2行を付け加えるだけでタイムアタックでき、格段に面白くなります。

完成した『九九ゲーム』遊んでみてね

- ・ 時間があまるようなら、繰り返しの数を大きくして遊んでみましょう。
- ・ 何十回もの繰り返しをもし人間が行ったら、どうなるかを考えましょう。

＼考えよう！／

3箇所ある『2*』を『3*』に変えます。エンターキーを押し忘れやすいので注意しましょう。

6. ネコをつかまえる

主な内容：カーソルキーでキャラクターを動かすプログラムとそれを用いたゲーム作成

小学校低学年向けの教材としては最後の回です。低学年の子にはかなり難しい内容ですが、プログラムを入力して遊ぶだけでも、より高度な内容への入り口として良い体験になるでしょう。

キー入力で位置を変えよう

キー入力で位置を変えよう

次のように、20行、40行、100行のプログラムを追加しましょう。

```
20 I = INKEY()
40 IF I = 29, X = X + 1
100 GOTO 20
```

Enter を押して実行すると、カーソルキーの → を押すたびに、鳥が右に増えていきます。

うまく動かないときは、ESC を押して実行を止めて、F5 を押してプログラムを表示し、20行、40行、100行のプログラムに間違いがないかよく確認しましょう。

INKEY() は、キーボードのどのキーが押されているか調べる命令です。
右矢印キーが押されている場合、変数 I には 29 が入ります。何も押されていない場合は 0 が入ります。

40行のIF(もしも)によって、Iに29が入っているときだけ、 $X = X + 1$ が実行されます。
 $X = X + 1$ は、変数式ですね。この+は、左向き矢印のように考えてください。

押されたキーに対応する数が入る

計算して

入れ直す

-04-

- ・「複数の命令を組み合わせて、目的の動作をさせる」という説明に相応しいプログラムです。
- ・このキーが押されたらこうなる、という仕組みを用意しておいて、人間が操作をするまで待ち構える、という考え方も盛り込まれています。

縦にも動かそう

＼考えよう！／

□には『-』（マイナス）が入ります。

縦横で動かな？

この章までが最低限の内容です。ここで完成したプログラムは、IchigoJamでキャラクターを動かすゲーム等のプログラムの基礎と言える内容です。

時間があまったら1:ゲームにしよう

ごく簡単な機能を付け加えて、単に上下左右にキャラクターを動かせるプログラムが、面白いゲームになるという好例です。自分でもシンプルなアイデアでゲームが作れると説明してあげてください。

7. スカッシュゲームを作ろう

※小学校高学年（4～6年生）向け、中学生向けのみ

主な内容：画面上でキャラクターを自動的に動かすプログラムとそれを用いたゲーム作成

かなり長い内容です。45分で終わることは難しいかもしれません。時間が足りなければ2回に分けるなどしてください。重要な内容は前半に集中していますので、途中で終わらせても結構です。急ぐあまりに、プログラムを入力するだけにならないように注意してください。

長いので、完成形を最初に見せて、モチベーションを上げてあげても良いかもしれません。

●を左右に動かしてみよう

また **ESC** を押して止めて、160行を次のように変えてみましょう。

```
160 X=X+A
```

Aもやはり変数で、-1が入っているので●が左に動きます。
では、Aに1を入れたら、どっちに動くと思いますか？
そう、右です。

もしも、●が画面の左端まで行ったときに、Aに1を入れることができたなら、●が右に進み始めます。
次のプログラムを追加してみましょう。●が画面の左端で跳ね返ったら成功です！

```
110 IF X+A=0 A=1
```

●が画面の右端31の位置まで行った時に、左に跳ね返るには、どうしたらよいでしょうか？
110行とほとんど同じ、120行を作ってみましょう。
120の部分には、何が入るでしょうか？

```
120 IF X+A=31 A=-1
```

-03-

この回の最も重要な部分です。理解しているかを確認めながら進めてください。

- ・ 等速運動と跳ね返りという「動き」を表現するプログラムです。
- ・ 2つの変数を、1つを座標、もう1つを移動量として、組合わせて使っています。

●が上下に動くようにしよう

斜めに動くようにするにはどうしたら良いか、答えは次の章のプログラムです。

●が斜めに動き、画面の端で跳ね返り、音が鳴る。

8. 的あてゲームを作ろう

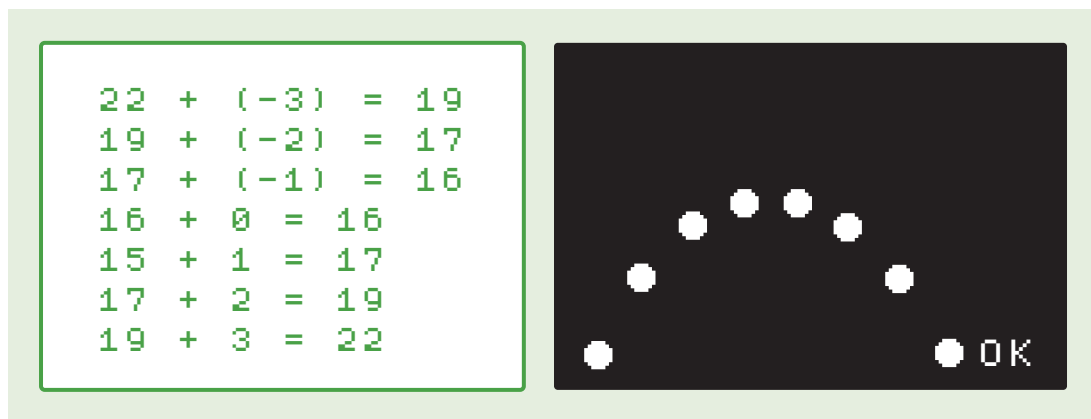
※小学校高学年（4～6年生）向け、中学生向けのみ

主な内容：中学理科・数学で習う放物線をシミュレートするプログラムとそれを用いたゲーム作成

「7. スカッシュゲームを作ろう」の内容に加速度運動を加えて、放物線のプログラムを作成します。放物線は中学の理科・数学で学習する内容ですので、それと関連づけて扱っても良いでしょう。

重力をプログラミングしよう

- ・変数Yを縦方向の座標、Vを移動量（速度）として動きを計算し、さらに加速度1でVを変化させることで、加速度運動をプログラミングします。
- ・単純な計算ですが、うまく使うと自然な弾道やジャンプなどを表現できます。
- ・内容をすんなりと理解することは難しいプログラムですが、計算結果の数値と画面の動きを確認することで、プログラムの概要とその結果を体験的に把握することが狙いです。



的を撃とう

- ・70行の $V = -3$ を変更すると、縦方向の初速が代わり、到達距離が変わります。
- ・いろいろな数に変更して試してみましょう。どうすれば的に当たるようになるでしょうか。

あたり判定をつけよう

- ・完成するとかなり面白いゲームになります。
- ・的の形をデザインして変えてみましょう。

9. 記憶力の限界に挑戦！

※小学校高学年（4～6年生）向け、中学生向けのみ

主要内容：配列の概念と使用方法、記憶力ゲームの作成、問題を増やすよう改造する方法

変数と配列

・変数のおさらいの後、配列の概念と使い方を、変数と比較することで説明しています。

変数と配列

変数は、数を入れておくことのできる箱のようなものでしたね。

IchigoJamでは、AからZまで26個の変数を使うことができます。
変数とよく似ていてちょっと違う、**配列**というものもあります。例えていうなら、**変数は家、配列はマンション**のようなものです。

へんすう
変数

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

はいれつ
配列

1 2 3 4 5 6 7 8 9 10 11.....99 100

記憶力ゲームを作ろう

- ・まず答えが常に1になる記憶力ゲームを作り、基本的な仕組みを学びます。
- ・変数、PRINT、WAIT、INPUT、IFなど、これまでの命令がたくさん組み合わせられています。

2問目を作ってみよう

- ・このあと配列を用いてたくさんの問題を出題するようにしますが、説明のために、その前段階として変数Bを用いて2問目を作っています。
- ・2問目の答え合わせのプログラムは省略しているため、間違ってもマチガイ！とは表示されません。

20問にしよう 記憶力の限界に挑戦!

- ・10問は全問正解できる子が多くいますが、20問はほぼ不可能なようです。
- ・時間が足りなければ、この章は省略して構いません。

問題が増えていくようにしよう

- ・ 100問まで問題が追加されていく、スリリングなゲームになります。
- ・ 配列を用いることで、たくさんのデータを扱うプログラムが作れたことを強調して説明しましょう。

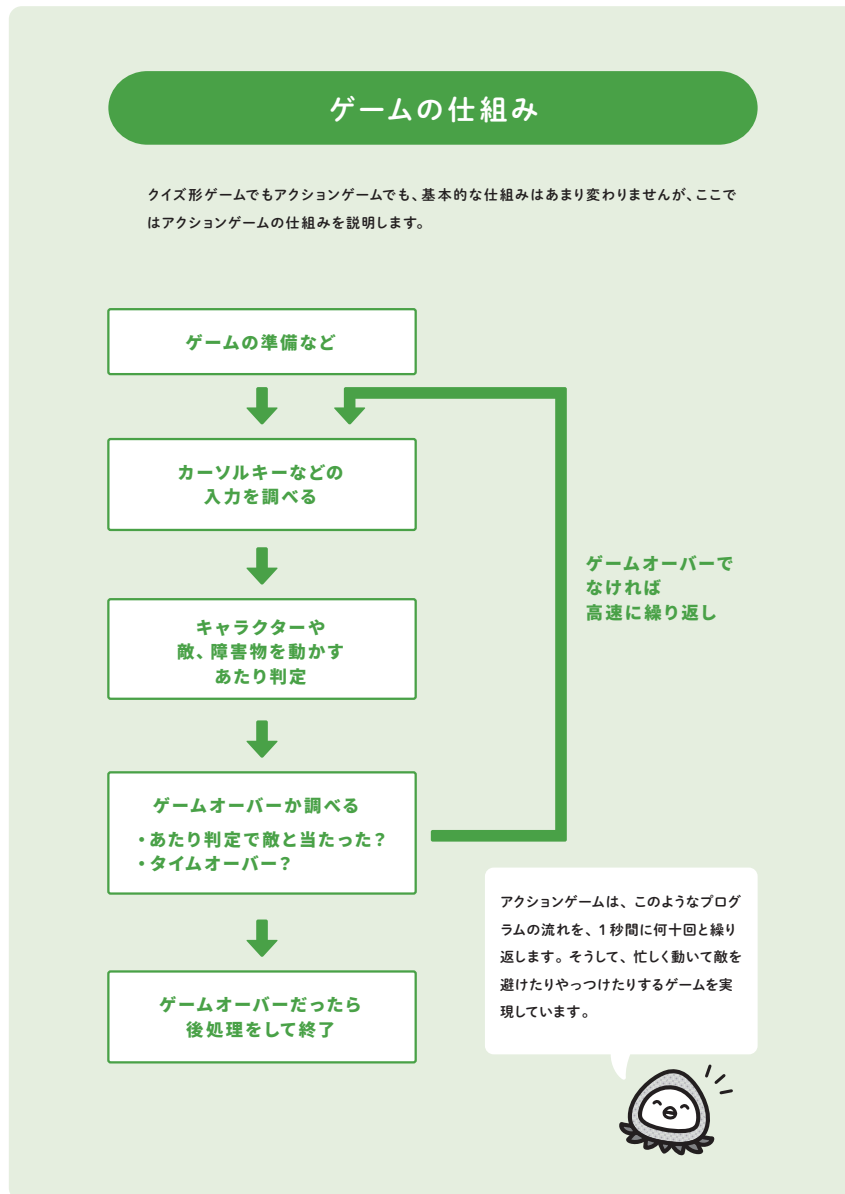
10. オリジナルゲームを作ろう

※小学校高学年（4～6年生）向け、中学生向けのみ

主な内容：ゲームの仕組みの理解、あたり判定、シンプルさの重要性、ゲームのアイデア

ゲームの仕組み

この図はとても重要です。実はゲームだけでなく、ほとんど全てのコンピューターソフトウェアに共通の、基礎となる仕組みを表しています。



-02-

ゲームを作ろう

・これ以降の章では、ただ入力するだけでなく、それぞれのプログラムが上図のどこに相当するのか、全体の中でどのような役割を果たしているのかを、考えさせながら進めましょう。

ゲームを改造しよう

- ・ 荒削りな、あまり面白くないゲームが、改造していくことで品質が高まり、格段に面白くなっていく過程を体験させましょう。
- ・ 左右に抜けることで無敵になる裏技があります。どうすればそれを防げるか、考えてみましょう。

シンプルなオリジナルゲームをつくらう

- ・ 小さく始めて徐々に改造していくことは、自分でプログラムを作るときにとっても重要な考え方です。
- ・ ここで例示しているプログラムは、「6. ネコをつかまえる」のゲームに近く、そのプログラムを流用して、実際に作ることもできるでしょう。

ゲームのアイデアを考えよう

- ・ 既存のゲームを参考にするのも良いですが、既視感のある、似通ったものになりがちです。
- ・ 日常生活をヒントに発想してみると、思わぬ面白そうなアイデアが出る場合があります。
- ・ お互いのアイデアを言い合ったり、人のアイデアを膨らませたりすることで、良いアイデアが出たりします。
- ・ 最初はあまり制限せずに面白いゲームを考え、ある程度アイデアが出たところで、それらのシンプルな最初のバージョンはどのようなものか、考えたり相談したりしてみましょう。