

ゲーム制作に今すぐ使える！

スクラッチ

Scratch

プログラミング

ゲーム

テクニック集



キッズアイティ合同会社
鍋藤優 著

付録

シューティングゲームの
プログラム／解説

プレイヤー操作

弾の発射・連射

敵の動き・攻撃・弾幕

HP・ダメージなど

ジャンプ・落下

背景スクロール

Scratch3.0

対応

まえがき

2020年時点ではスクラッチプログラミングの入門書が多数出版されており、以前に比べてプログラミング教室に通わずとも気軽にプログラミングをスタートできる環境が整っております。

入門書に記載されている内容を本質的に理解できれば、それを応用して様々な作品を作ることができます。本質的に理解するためには入門書を一通り終わらすだけでなく、継続してプログラミングに取り組み、パターンを覚え、繰り返し使うことで理解を深める必要があります。

お子様がプログラムを作る動機として最も多いのが、「ゲームを作る」ということです。ゲームを作る目的でモチベーションを高め、様々な作品作りにチャレンジする事が本質を理解しながら力をつける近道だと考えております。

私が運営しておりますキッズプログラミング教室では、小学生（または中学生）に教えるにあたり基礎的な命令の使い方や考え方を積み上げる形で、論理的思考力や想像力を身に付けてもらっています。基礎力が身に着いた後は、ゲームなどの作品作りを中心に取り組んで頂いていますが、そこでよくある質問をテクニック集としてまとめました。

様々な用途別に、簡単なテクニックから難易度の高いテクニックまで多数紹介しておりますので、必要に応じてご利用ください。

また付録としてシューティングゲームの作り方も紹介しております。これらをカスタマイズしてゲームの骨組みなどを理解いただければ幸いです。

キッズアイティ合同会社
代表 鍋藤 優

本書の使い方

テクニック毎にチャプターを分けておりますので、作りたいゲームの用途に応じてご利用ください。（先頭から順番に実施して、テクニックを1つずつマスターして頂いても問題ございません）

プログラムに対して可能な範囲で説明を入れておりますが、スクラッチの操作方法や、命令ブロック個別の説明など基礎的な部分は省略しておりますので、入門書等と併せてご利用頂くと効率良く学べます。また、各プログラムは特典サイトよりダウンロードできます。ダウンロードURLは巻末に記載しております。

目次

1章 キャラクターを動かそう

テクニック1-1	矢印キーでスプライトを動かす	6
レベルアップ	滑らかに動かす方法	8
テクニック1-2	マウスでスプライトを動かす	9
レベルアップ	マウスポインターに触れたら止める	11
テクニック1-3	ジャンプしよう	12
レベルアップ	【ずっと】命令に含める場合	14
テクニック1-4	弾を発射しよう(単発・連射)	15
レベルアップ	クローンで連射しよう	18
レベルアップ	向いている方向に発射しよう	19
テクニック1-5	移動できない場所を作ろう	20
レベルアップ	プログラムをまとめよう	24

2章 敵を作って動かそう

テクニック2-1	プレイヤーを追いかける敵	26
テクニック2-2	決まった動きをする敵	29
レベルアップ	条件に応じて動きを変えよう	33
テクニック2-3	ランダムに発生する敵	34
レベルアップ	プレイヤーが敵に触れた判断	37
テクニック2-4	敵の攻撃を作ろう	39
レベルアップ	弾の発射をランダムにしよう	42
レベルアップ	弾幕を作ってみよう	43

3章 イベントを作ろう

テクニック3-1	オープニング/ゲームオーバー	45
テクニック3-2	ダメージをカウントしよう	50
レベルアップ	ダメージに応じて動く速さを変えよう	54
レベルアップ	HPをバー形式で見えるようにしよう	55
テクニック3-3	タイマーを理解しよう	56

4章 応用テクニック

テクニック4-1	スクロールする背景を作ろう	60
テクニック4-2	重力を想定したジャンプ・落下	64
テクニック4-3	リストを使って動かそう	74
レベルアップ	スプライトの動きをリストに記録	77

付録	シューティングゲームを作ろう	78
----	----------------	----

1章

キャラクターを

動かそう

テクニック

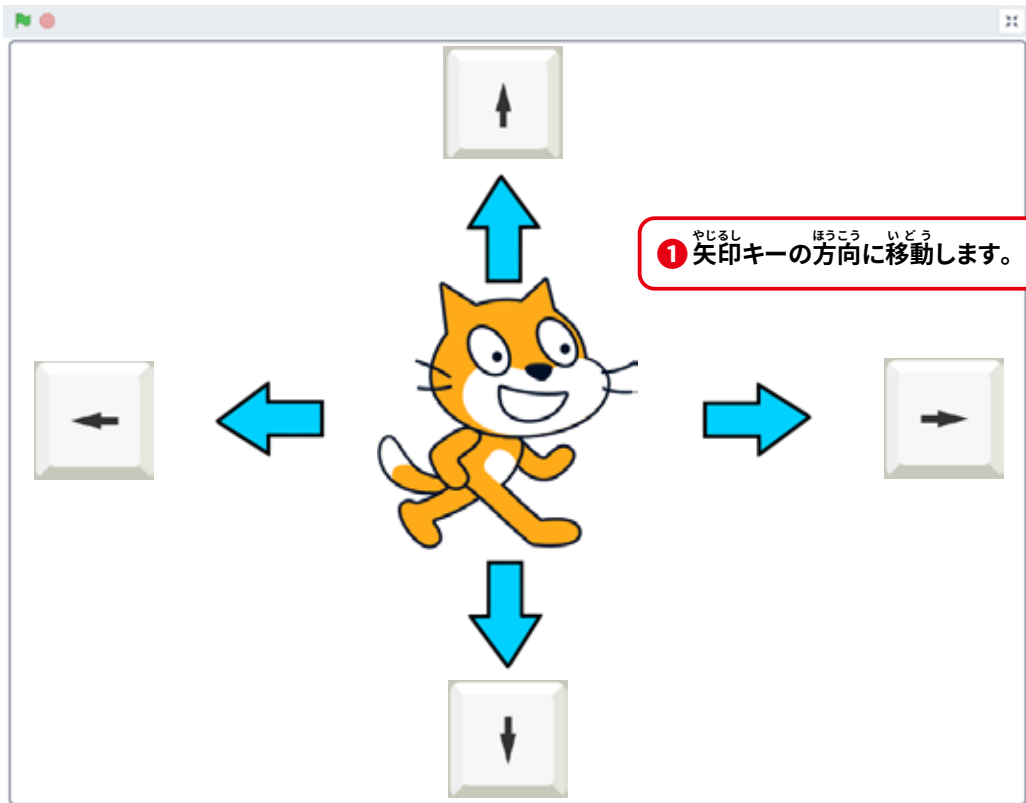
矢印キーでスプライトを動かす

1-1

プログラムをスタートするイベントにキーを設定するとキー毎のプログラムを作れます。

ゲームを作る際に、キャラクターを自分で動かす事がよくあります。

キーボードの矢印キーを使って、スプライトを動かす方法をマスターしましょう。



プログラムを作るポイント

プログラムを動かすキッカケの一つに【～キーが押されたとき】命令ブロックが用意されています。キー毎に動かすプログラムが違う場合は、この命令を使うと便利です。

この場合、スペースキーが押された場合に、このブロックに繋がっている命令ブロックが実行されます。

スペースキー以外のキーにて設定したい場合は、▽をクリックしてキーを選びましょう。

スペース ▾ キーが押されたとき

STEP 01

まず1つの方向のキーから作りましょう

まず、上向きの矢印キー（↑）を押した時のプログラムを作ります。

この場合、画面の上方向にスプライト（ネコ）を動かしたいので、上向き（0度）に向けてから移動します。



① プログラムが動くキッカケを「上向き矢印」のキーにします。

② スプライトを上に向けてから、その方向に移動します。

STEP 02

他の3方向のキーも同じように作りましょう

上向きキー用のプログラムが完成したら、同じように他のキーも作ります。

考え方は同じで、それぞれのキー毎に、動かしたい方向に向けて移動させます。



POINT

今回のように似たようなプログラムをいくつか作る場合、プログラムをコピーすると効率よくプログラミングできます。
コピーしたいプログラムを右クリックして、「複製」を選びましょう。

なめらかに動かす方法

先ほどのイベントにてキーを押した判断をする方法だと、動き始めの反応が悪くなってしまいます。そこで、【ずっと】命令を使い、キーを押したことを判断する方法に変えると、滑らかに動かすことができます。また、左方向に移動した時に逆さまになってしまうので、これも対策しましょう。

```

    旗が押されたとき
    回転方法を 左右のみ にする
    ずっと
    もし 上向き矢印 キーが押された なら
        0 度に向ける
        10 歩動かす
    もし 下向き矢印 キーが押された なら
        180 度に向ける
        10 歩動かす
    もし 右向き矢印 キーが押された なら
        90 度に向ける
        10 歩動かす
    もし 左向き矢印 キーが押された なら
        -90 度に向ける
        10 歩動かす
  
```

1 プログラムが動くキッカケを緑の旗がクリックされたときにします。

2 左に移動する際に、逆さまにならないように、回転方法を指定します。

3 常にキーを押した判断が必要となるため【ずっと】で囲みます。

4 それぞれの方向キーが押されたときに移動する命令を入れます。

POINT

回転方法を 回転しない にする

向きを変えた時の表示方法については他にも【回転方法を回転しないにする】という命令ブロックがあります。この命令ブロックを実行した後は、スプライトは常に右向きの状態となります。ただ、見た目は右向きですが、内部的にはそれぞれの方向に向いており、進むとその方向に移動します。

テクニック

マウスでスプライトを動かす

1-2

マウスポインターの方向に向ける命令と動く命令を組み合わせて作ります。

スプライトを動かす方法として、キーボードだけでなくマウスポインターで動かすこともできます。マウスの方向に向けて動く仕組みを作りましょう。



プログラムを作るポイント

スプライトの向きを、マウスポインターや他のスプライトに向ける命令が用意されています。この命令でマウスポインターの方向に向けてから動かす命令で移動させます。

マウスポインターだけでなく、用意された他のスプライトに向けることも可能です。

鬼ごっこのように、追いかける仕組みを作るときはこの命令が便利です。

マウスのポインター ▼ へ向ける

STEP 01

マウスポインターに向けて動かす

仕組みはとても簡単で、マウスポインターに向けてから動かします。
連続してマウスポインターに向けて移動するため、【ずっと】命令ブロックで囲みましょう。



① 常に動かすので【ずっと】で囲みます。

② スプライトをマウスポインターに向けてから動かします。

STEP 02

動く速さを調節しよう

あとはスプライトの動く速さを調節します。
速くしたいのであれば【10歩動かす】の数字を15や20などに増やしましょう。



速くしたい ⇒ 数字を増やす
遅くしたい ⇒ 数字を減らす

よくある間違い

スプライトを動かすスピードを遅くしたい場合に、【1秒待つ】を入れて調整してしまう場合があります。この方法だと、動きがその都度止まってしまい、滑らかな動きにならないので注意しましょう。

待つ命令を入れてしまうと動きが悪くなる



マウスポインターに触れたら止める

マウスポインターを追いかけるプログラムを作りましたが、スプライトがマウスポインターに到達した後も連続して動き続けるため、ガタガタ動いてしまいます。マウスポインターに触れたら一旦止まるように改良します。作り方を2つ紹介しますので、使いやすいパターンを選んでください。

パターン1

が押されたとき

ずっと

マウスのポインター ▼ に触れた ではない まで待つ

マウスのポインター ▼ へ向ける

10 歩動かす

マウスポインターに触れている間は動かす必要が無いので、条件を入れて待ちます。

パターン2

が押されたとき

ずっと

マウスのポインター ▼ に触れた まで繰り返す

マウスのポインター ▼ へ向ける

10 歩動かす

マウスポインターに触れていない間のみ繰り返します。