

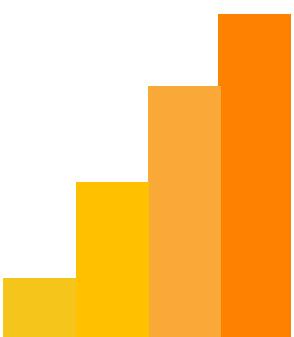
SCRATCH で

ハードルゲームを作ろう！

～上級編～

製作

木更津工業高等専門学校 プログラミング研究同好会



こんにちは！ ゲーム制作体験会に参加してくださりありがとうございます。この体験会では、Scratch(スクラッチ)と呼ばれるものを使って、プログラミングでゲームを作成してみます。

今回は上級編ということで、初級編・中級編をクリアしている必要があります。まだの人は、右のQRコードから初級編・中級編の資料に飛んで、両方をクリアしてから、また戻ってきてください。



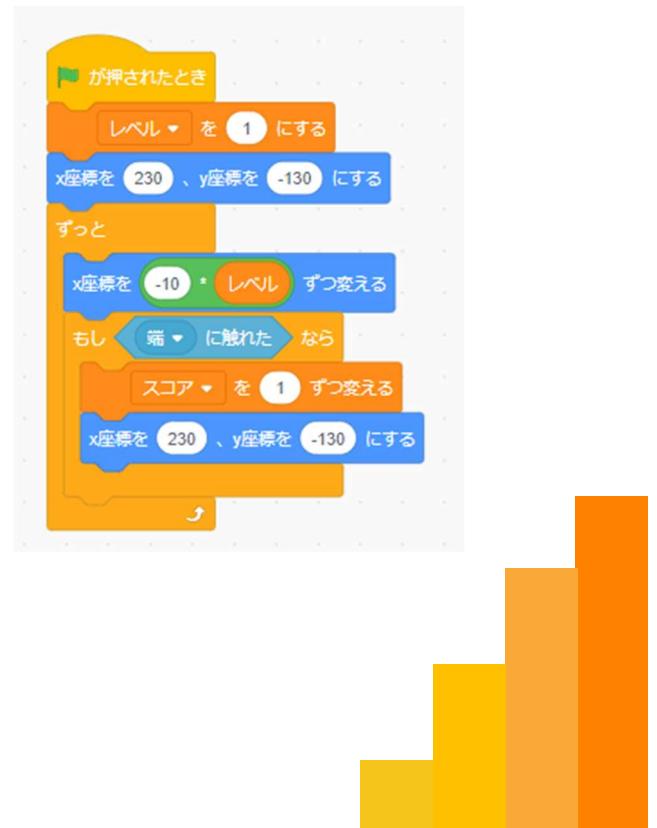
今回は、
・難易度設定
・物理法則を意識したジャンプ
・背景の作成
を行います。

1. 難易度設定

今までではハードルは、比較的ゆっくりであり一定間隔で出現するようにしていました。今回は難易度調節ということで、
・速度がだんだん早くなっていくように
・タイミングにランダム要素を持たせる
の2点で改良をしてみたいと思います。

まずは、速度から実装していきます。
方針としては、内部でレベルを実装し、速度をレベル倍にするように設定したうえで、徐々にレベルを上げていくようにします。
(ほかにも方法はあるので、余裕があったら考えてみよう)

ハードル側で、まずはレベルを用意して初期化。最初は1から始めることとして、速度もレベル×速度になるように改造してあげます。



これにより、レベルに比例して速度も上がるようになりました。

後は、レベルの上がり方を調整するだけです。

僕は右のように設定して、端に触れたときにレベルが上がるようにして、さらに1回あたり0.05上がるようになりました。

皆さんも各自で調整してレベルを設定してみてください。

今のジャンプの実装だとどうしても高レベルが対応できないので、次の章を終わらせてからここに戻ってきてレベル調整するといいかもしれません。



次に、タイミングにランダム要素を持たせる方法です。

もう少し高度な方法として、自身の複製(クローン)を作るという方法もありますが、今回のゲームにおいてそこまで高頻度でハードルを出す必要はないため、ここでは使いません。(背景のところで説明します)

今回は、端に触れた後一度非表示にし(隠す)、ランダム時間経過後初期位置に移動してまた表示する という処理を行いたいと思います。

「見た目」カテゴリには「表示する」「隠す」というブロックがあります。これを使うと、スプライトを表示したり非表示にしたりできます。

また、「演算」カテゴリには「〇から〇までの乱数」というブロックがあります。このブロックは少し特殊で、整数を入れると整数、小数を入れると少数の範囲で乱数が返ってきます。ちなみに、1.0などと整数の最後に「.0」を付けると小数判定になります。

これを使って、消えてから少ししてからハードルが出るようにしてみます。

まず、端についたらいったん隠して初期位置に移動させます。

次に、僕は「0.0 秒から 1.0 秒まで の乱数 秒待つ」を入れた後、再度表示させてハードルを進めるようにしてみました。

ここも、皆さんの工夫が問われるところだと思います。色々考えてやってみてください。



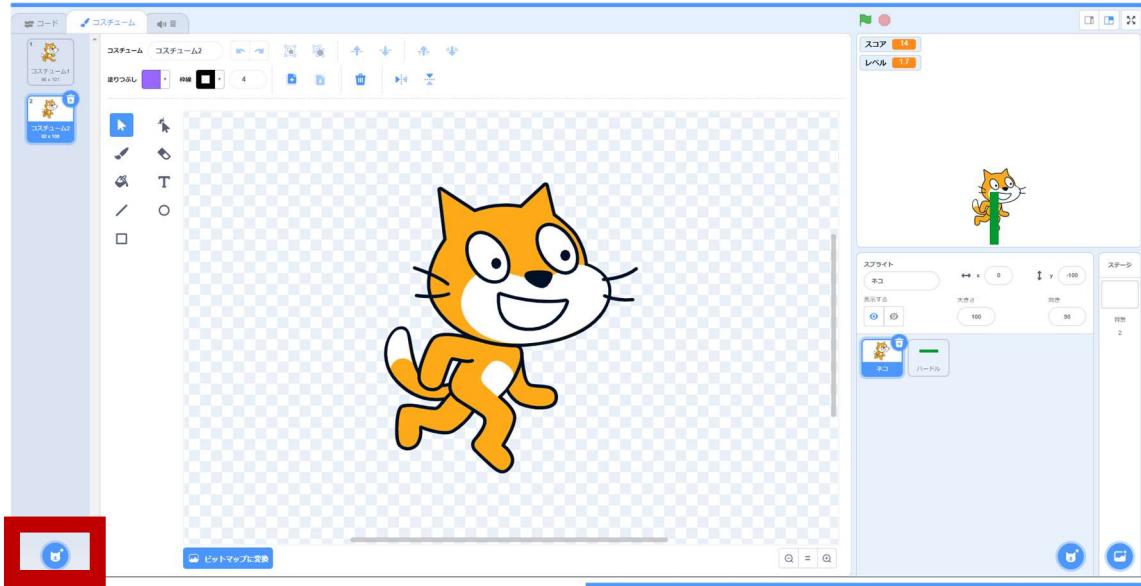
2. 物理法則を意識したジャンプ

ここまで初級編・中級編にわたってジャンプについて改良を加えてきましたが、現実世界のジャンプとはまだ少し違います。今回は、以下の 2 点に変更を加えてみます。

- ・ジャンプ中はジャンプのコスチュームにする
- ・加速度(速度が変化していく)を付けてみる

まずはジャンプ中にコスチュームを変える(=足をじたばたさせない)ようにします。実装方針としては、フラグとなる変数を用意して、その変数が 0 ならコスチュームを高速で切り替え、1 ならジャンプ中コスチュームにする とします。

まずはジャンプ中コスチュームを用意します。上のコスチュームタブから、左下の「コスチュームを選ぶ」ボタンを押してください。



その後、左上の検索欄に「cat」と入れたら、「Cat Flying-b」をクリックしてください。こいつはジャンプ時のコスチュームに向いてそうです。



これでコスチュームが追加されました。今追加されたコスチュームは3番になっているはずです。

では、コスチューム切り替え部分を作っていきます。まずはフラグ変数を用意して、0で初期化をします。

また、ジャンプ開始時に1、終了時に0にするようにします。これにより、ジャンプ中はフラグ変数が1になります。

ついでに、ジャンプ中にはコスチュームを Flying-b に設定しておきましょう。「見た目」カテゴリから、「コスチュームを〇〇にする」ブロックが使えます。



そして、「次のコスチュームにする」の部分を少し改造します。
3番目のコスチュームが追加されたことで「次のコスチュームにする」ブロック
が使えなくなりました。

そこで、「コスチュームを〇にする」ブロックを活用して、現在1番のコスチュームであれば2番、2番のコスチュームであれば1番にするようにします。「見た目」カテゴリにある「コスチュームの番号」ブロックを使うと現在のコスチュームの番号が取得できます。

それを使って、「制御」カテゴリの「もし〇〇なら、でなければ～」ブロックで条件分岐させましょう。このブロックは、条件を満たさなかったら「でなければ～」のほうに入る条件分岐のブロックです。

ついでに、フラグ変数が1の場合は空中に居てコスチューム変更禁止ですから、その場合は処理そのものをスキップするように、「もし〇〇なら」ブロックを使って作っておきましょう。

これらを反映させると、右のようになります。実行してみると、見事ジャンプ中は手を伸ばすコスチュームになっています。



次に、加速度を反映させたジャンプを実現してみます。
加速度とは速度の変化のこと、ジャンプの時は「一定の速度から速度を引いて行って、速度0になるまで上に上がる」「速度0から一定の速度を足していくて、地面にぶつかるまで下に下がる」ような運動をします。

物理学的には鉛直投げ上げという運動であり、高専1年生で勉強する内容です。
今回はそれを簡単に実装してみます。

まずは、速度を管理する速度変数を作成します。

次に、ジャンプの指令が出たときに以下のような処理を行います。

1. 速度の初期値セット(僕は 15 にしました)
2. 端から離れるまで「y 座標を速度ずつ変え、速度を-0.98 する」を繰り返す
3. 端につくまで「y 座標を速度ずつ変え、速度を-0.98 する」を繰り返す

この処理を具体的に説明すると、「端から離れるまで繰り返す」「端につくまで繰り返す」の 2 つを並べることで、「端から一度離れた後もう一度つくまで繰り返す」という処理になります。

その中の処理は、速度をだんだん減らしながらその分移動していくというものの繰り返しです。途中でマイナスになるので、下向きに変化し、その結果もう一度端につくことになるのです。

ジャンプの高さを挙げたければ初期値を、重力を変えたければ速度の変化をいじることで変更ができます。

この時ポイントになるのが、「端から離れる」「〇〇まで繰り返す」という 2 つのプログラムです。

「端から離れる」は、端から離れる=端に触れた ではない と捉えることができるので、「演算」カテゴリにある「〇〇ではない」と組み合わせて実現できます。

「〇〇まで繰り返す」は、「制御」カテゴリにある「〇〇まで繰り返す」ブロックを使えます。これを使うと、〇〇が成り立つまでの間繰り返し処理が行われます。

なお、このプログラムによりネコの一番下の座標が元の座標から変わり、 $y=-140$ ぐらいになります。ネコの初期座標も $y=-140$ にしておきましょう。



3. 背景の作成

次に、背景を作っていきます。ただ背景を設定するだけではなく、奥にものを配置してゆっくり流していきたいと思います。

この章は自由度が高いので、プログラム例は一つの参考だと思ってください。

まずは、スプライトエリアのところから「ステージ」を選択した後、左上からコストチュームに行き、ネコの時と同様の方法で好きな背景を選びます。なんでもいいです。



次に、右のコードを削除します。これは、座標がわからなくなったら簡単に座標を確認できる背景と切り替えるためのものですが、もう使わないと思うので消します。



次に、背景の動くオブジェとなるスプライトを追加します。ステージエリアの右下のボタンからスプライトを追加できます。

マウスカーソルを合わせると色々選択肢が出てくるので、すでに用意されているの以外に、自分で書いてもOKです(背景も同様です)。



いくつか、現実であれば距離感の異なるスクリプトを追加してあげるといいでしよう。

僕の場合は、木を2つ、それと雲を追加しました。

ここから、それぞれのスクリプトに対してプログラムを組んでいきます。どれも共通なので、追加したもののは性質に合わせて値を調節してみてください。

具体的には、以下のような処理を行います。

1. ゲームが開始されたら $x=230$ に移動 & 隠す & 最背面へ移動する
2. ランダムなタイミングで、事前設定した範囲内でランダムな y 座標に移動したり、大きさをランダムに変えたり、左右反転させたり、色を少し変えたりしてからクローンを作成する
3. クローンを左端まで移動させてから削除する

今回は個数が複数になることもあると思うので、クローン(自身の複製)を使った処理を行います。

まずは、ゲームが開始したら木を隠して、「見た目」カテゴリにある「最背面へ移動する」ブロックを使います。これは、「最前面へ移動する」ブロックの▼を押すと変更できます。

そのため、ゲームを開始したら「○から○までの乱数 秒待つ」「自分自身のクローンを作成する」を繰り返すプログラムを作成します。

「自分自身のクローンを作成する」ブロックは「制御」カテゴリにあります。

また、この時乱数の範囲は皆さんで決めてください。

そして、このループの中で y 座標をランダムに変えたり、大きさを変えたり、左右反転させたり、見た目を変えてもいいと思います。

大きさを変えるには、「見た目」カテゴリの「大きさを〇%にする」ブロックを使います。また、左右反転させるには、「動き」カテゴリにある「回転方向を左右のみにする」を使った後に、同じく「動き」カテゴリにある「〇度に向ける」で-90 度に向けることで実現できます。



左右どちらかに向けたい場合は、「0から1までの乱数が0なら～、出なければ～」という処理で実現可能です。「演算」カテゴリにある「〇=〇」ブロックで値の比較ができます。

見た目の変更は、「見た目」カテゴリにあるブロックでできます。色々試してみてください。

ここまで出来たら、後は右から左へと移動させるプログラムを作ります。初めに表示してから、右から左へと移動させます。

内容はハードルとあまり変わりませんが、「端に触れたら」ではなく、「x<230」で判定をしたほうが良いです。なぜなら、左端以外のステージの端に触れる可能性があるからです。自身の座標は「動き」カテゴリの「x座標」「y座標」ブロックで調べることができます。

「制御」カテゴリの「クローンされたとき」を使うと、クローンを制御できます。また、同じく「制御」カテゴリの「このクローンを削除する」でクローンの削除もできます。

右にあるコード例は僕が雲に乗せたコードです。

このコードでは、大きさに応じて速さを変えるというのも工夫をしました。

小さいほど遠くにある=流れが遅くなる ということになりますから、「見た目」カテゴリの「大きさ」ブロックをうまく活用してみてください。

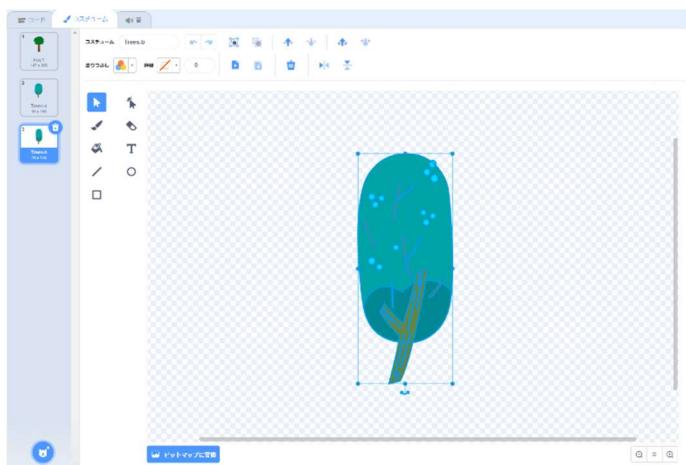


それと、ハードルに当たった時に送られる「ストップを受け取った時、スプライトのほかのスクリプトを止める」のコードも忘れずに作っておいてください。



大きさを変更させるときに、もしかしたら「中央を基準に大きさが変わってしまって、浮いてしまう」という悩みがあるかもしれません。

その場合は、「中心」を変えることで解決します。コスチュームタブを開いて、エディタ内で Ctrl キーと A キーを同時に押ししてください。



全てが選択されるので、その状態で絵のどこかをクリックしながら移動させてください。すると、右の図のようなマークが出てきます。これが中心です。
例えば中心をイラストの下に合わせると、拡大縮小しても浮かないようになります。活用してください。



このように背景のオブジェを作りこんだら、背景も完成です。
僕は背景の色と似ていてハードルが見づらくなっていたので、ハードルの色も
変更してみました。

4.最後に

ゲームの改良はどうだったでしょうか?

物足りない・もっとレベルアップさせたいと思った人は、自分で調べたり考えたりして、ゲームを改良させていってください。

Scratch ではまだまだいろんなことができます。ぜひ、限界に挑戦してみてください。

今回この資料はインターネットにて共有しているので、オンラインでいつでも見ることができます。また、この作品も Scratch の Web サイトで共有されています。さらに、初級編・上級編へのリンクや、Scratch の利用方法などを乗せたパンフレットのリンクも乗せておくので、是非見てみてください。

本日は木更津高専 プログラミング研究同好会のゲーム製作体験会に参加していただき、ありがとうございました。



ゲーム体験会パンフレット

<https://github.com/nitkc-proken/game-development-trial/blob/main/pamphlet.pdf>



上級編 PDF リンク

<https://github.com/nitkc-proken/game-development-trial/blob/main/advanced.pdf>



今回作成した作品

<https://scratch.mit.edu/projects/912102486/>