

ポン (PONG)

はじめに

ポン (PONG) は、二人で遊ぶ卓球ゲームです。キーボードを使ってキャラクターを操作して、ボールを交互に打ち合います。

キャラクター設定

3つのキャラクターを使います。ここでは必ずネコ、カエル、ボールを選んでください。

あと、それぞれのキャラクターの X 座標 (横)、Y 座標 (縦)、向き、角度を設定するのですが、下の図の通りにせず、おおよそ近い値になったら、プログラムを作成してください。

cat1



X: 0 Y: 183
向き: → 0°



frog1



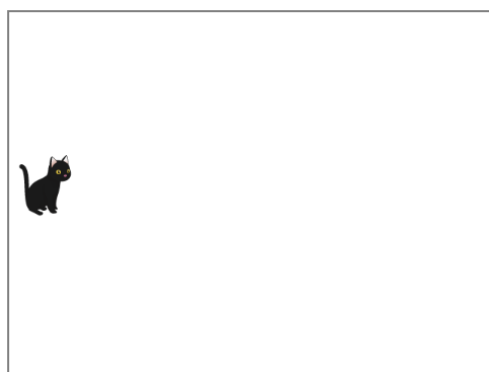
X: 544 Y: 183
向き: → 180°



ball1



X: 464 Y: 210
向き: ↻ 225°

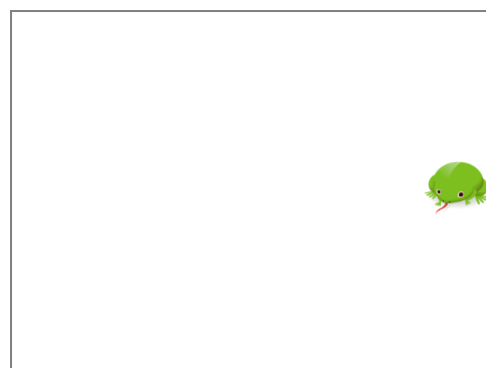


名前 cat1

X座標 0

Y座標 183

向き → 0°

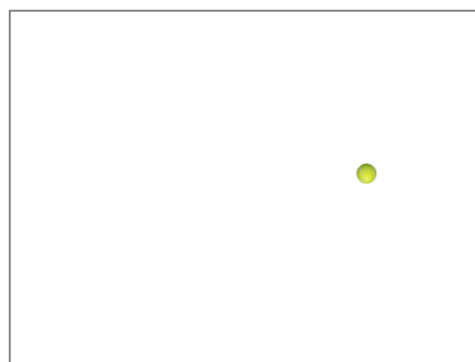


名前 frog1

X座標 544

Y座標 183

向き → 180°



名前 ball1

X座標 464

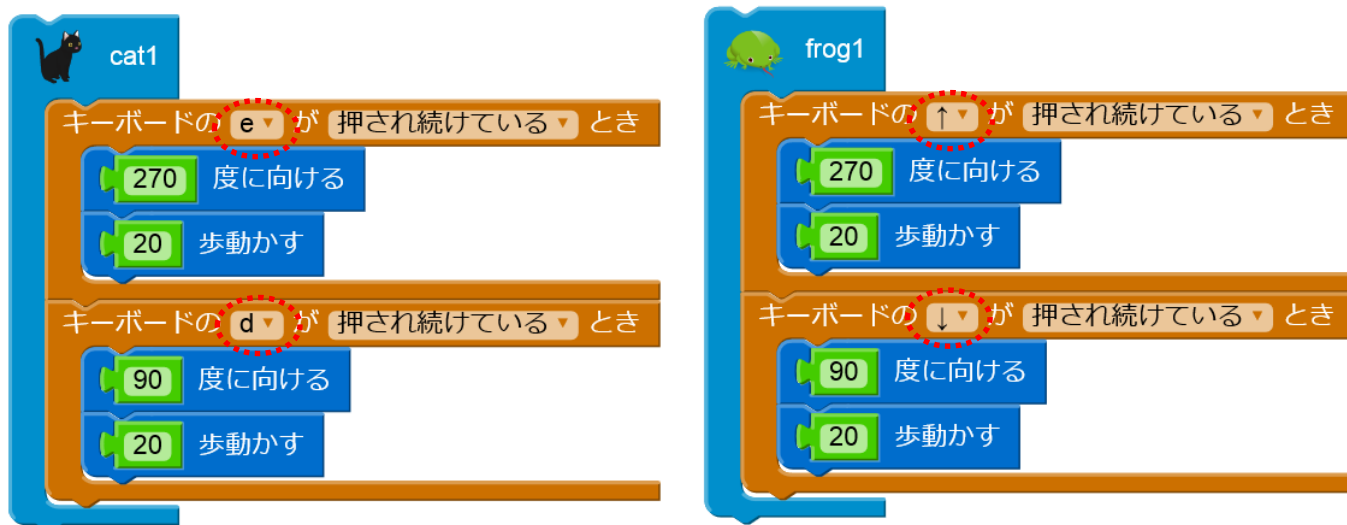
Y座標 210

向き ↻ 225°



プログラム①

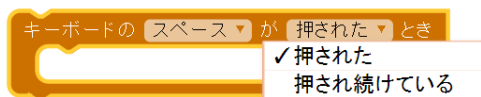
プログラムが長いので、プログラムをいくつかに分けています。プログラム①が終わったら、そのまま続けて、プログラム②を作ってください。間違っまちがてプログラム①を消さないでくださいね♪



ここまでできたら 実行 してみよう♪

どうやったらキャラクターを動かすことができるかわかるかな？

めいれい 命令ブロック



<イベント（～のとき）>

「キーボードの[▼キー]が[▼押された]とき」ブロック

これはキーボードのキーが一回押されたとき、またはずっと押され続けているときに行いたいことを表現するブロックです。

スペースキーを押したら続けるといったときは「押された」を使い、カーソルキーを押している間、キャラクターを移動させるときは「押され続けている」を使います。




<回転>

「[角度]度に向ける」ブロック

これはキャラクターの向きを指定した角度にすることを表現するブロックです。

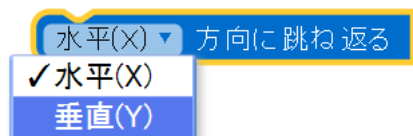
プログラム②（プログラム①の続き）

プログラム①を修正してボールも動かしてみよう！できた人は  実行 して、隣の人と対戦してみよう♪



この「128」や下の「501」はどうやって計算したのだろう？わかるかな。

命令ブロック



<回転>

「[▼水平(X)]方向に跳ね返る」ブロック

これはキャラクターの向きのうち、横(水平: X)または縦(垂直: Y)だけを反対方向にすることを表現するブロックです。



<移動>

「x座標を[0~639の数値]にする」「y座標を[0~479の数値]にする」ブロック

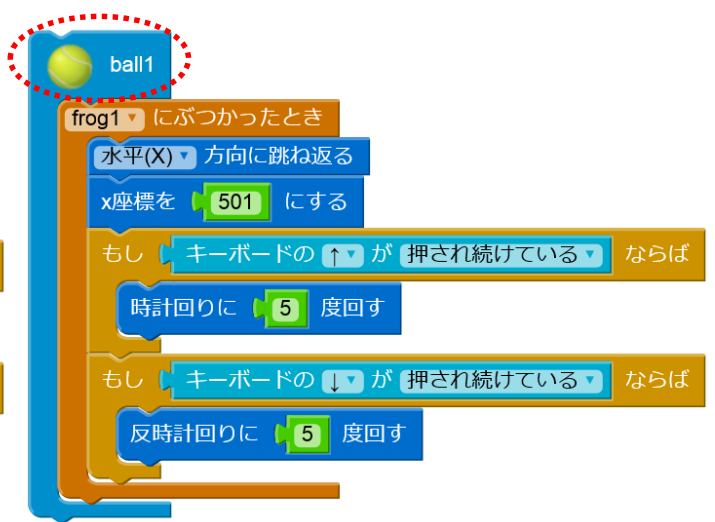
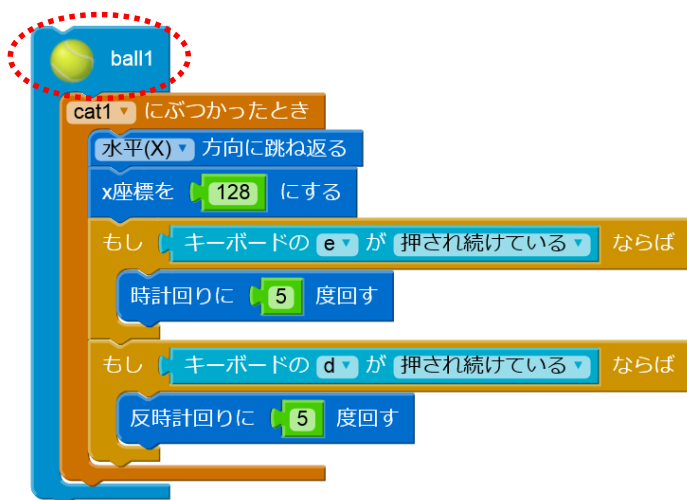
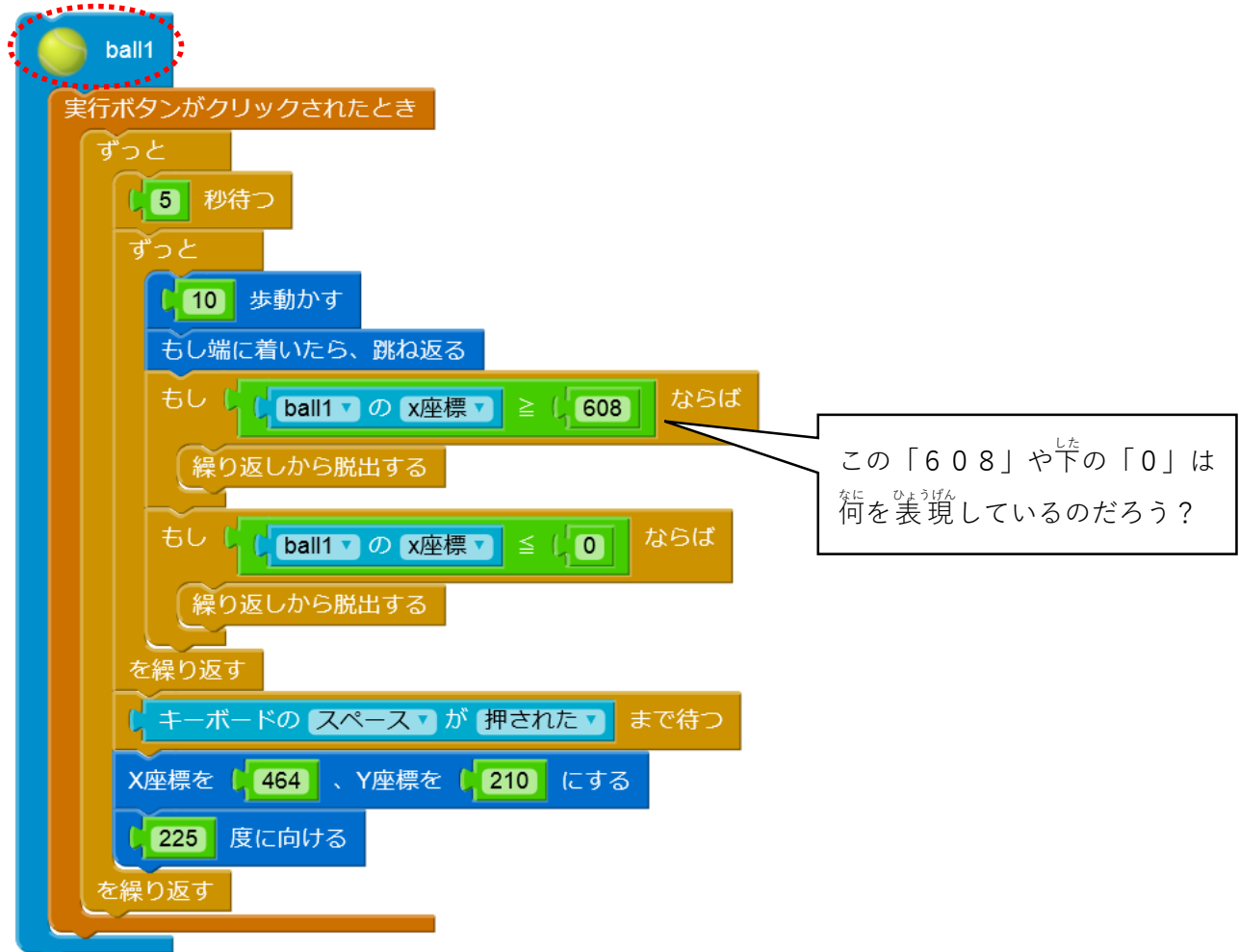
これはキャラクターの位置を指定することを表現するブロックです。x座標が横、y座標が縦になります。



プログラム③（プログラム②の^{つづ}き）

プログラム②を^{しゅうせい}修正して、よりゲームらしくしよう。どのように^か変わったかわかるかな？

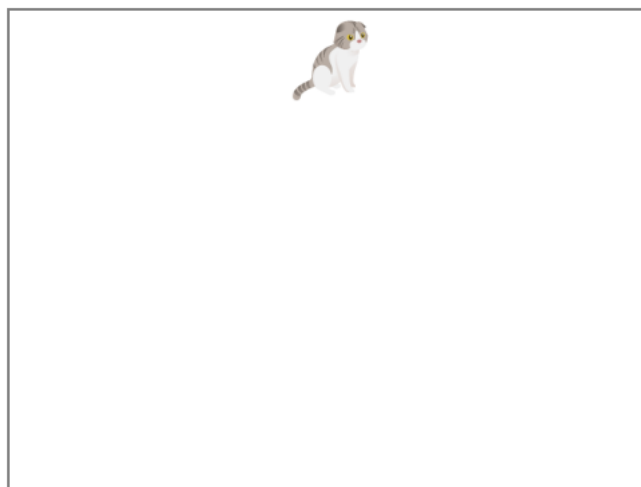
^{した}下にはボールが3つあるけど、よく見ると同じ「ball1」だね。キャラクターは追加せず^{ついか}に、キャラクターブロックを作ると、こんな風にプログラムを3つに分けることができるんだよ♪



プログラム④（プログラム③の続き）

まずは新しいキャラクター（審判）を追加して、ゲームが始まったらコートの中真ん中に線を描くようにします。

cat2



名前 cat2

X座標 275

Y座標 1

向き 90°

あれ？
なぜここは「0」で
はなく「1」なのか
な？？

cat2

実行ボタンがクリックされたとき

- ペンの色を 赤 にする
- 90 度に向ける
- ペンを下ろす
- 端に着いた まで
- “レディー・・・” と言う
- 20 歩動かす
- ほんの少し待つ
- を繰り返す
- “ゴー！” と言う
- 1 秒待つ
- 消滅する

命令ブロック

ペンの色を 赤 にする

<ペン>

「ペンの色を[▼色]にする」ブロック

これはキャラクターが持っているペンの色を設定することを表現するブロックです。

ペンを下ろす

<ペン>

「ペンを下ろす」ブロック

これはキャラクターが持っているペンを下した状態にして、キャラクターが移動したときにステージに線を描けるようにすることを表現するブロックです。

また、下したペンを上げて線を描かないようにすることを表現する<ペン>「ペンを上げる」ブロックもあります。



プログラム④の続き

次にキャラクターを上下だけでなく、左右にも動くようにします。

それぞれのキャラクターブロックを新たに追加して、それぞれキーボードの「f」「s」「→」「←」を押したときに左右に動くようにします。



プログラム④の続き

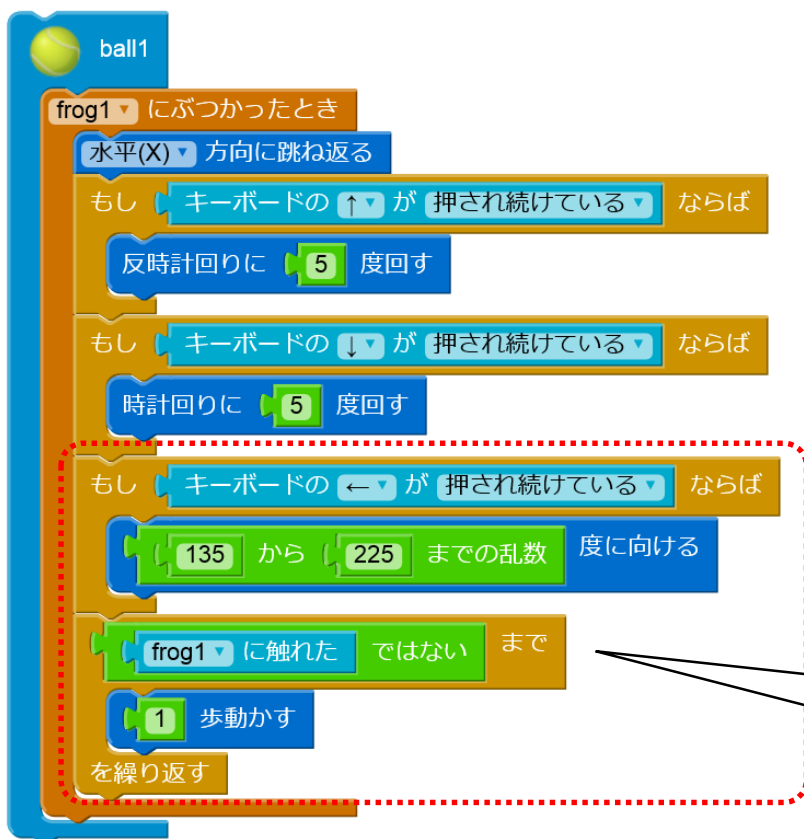
次にキャラクターとボールが当たった時の動作を修正します。

キャラクターが前に進みながらボールに当たると、ボールは前の方に飛んでいきます。このとき毎回同じ方向に飛ぶのではなく、ランダムな方向に飛んでいくよ♪

それとキャラクターとボールが当たった時にボールが自然に跳ね返るように工夫しています。



「cat1に触れたではないまで1歩動かすを繰り返す」となっているけど、ここでは何をやっているのだろう？



ここもだね。
何をやっているのだろう？



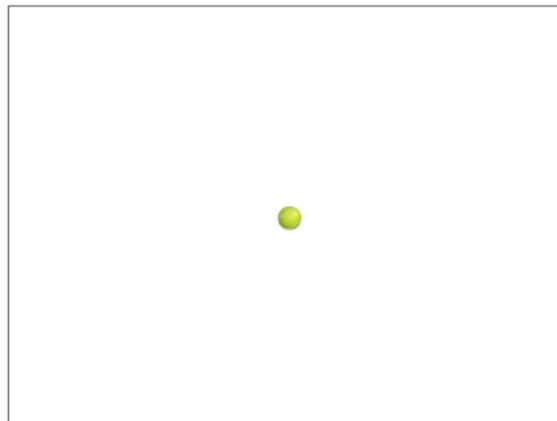
プログラム④の続き

最後にボールの最初の位置と方向を修正したら完成です。できた人は実行して、隣の人と対戦してみよう♪

ball1



X: 309 Y: 220
向き: ↗ 225°

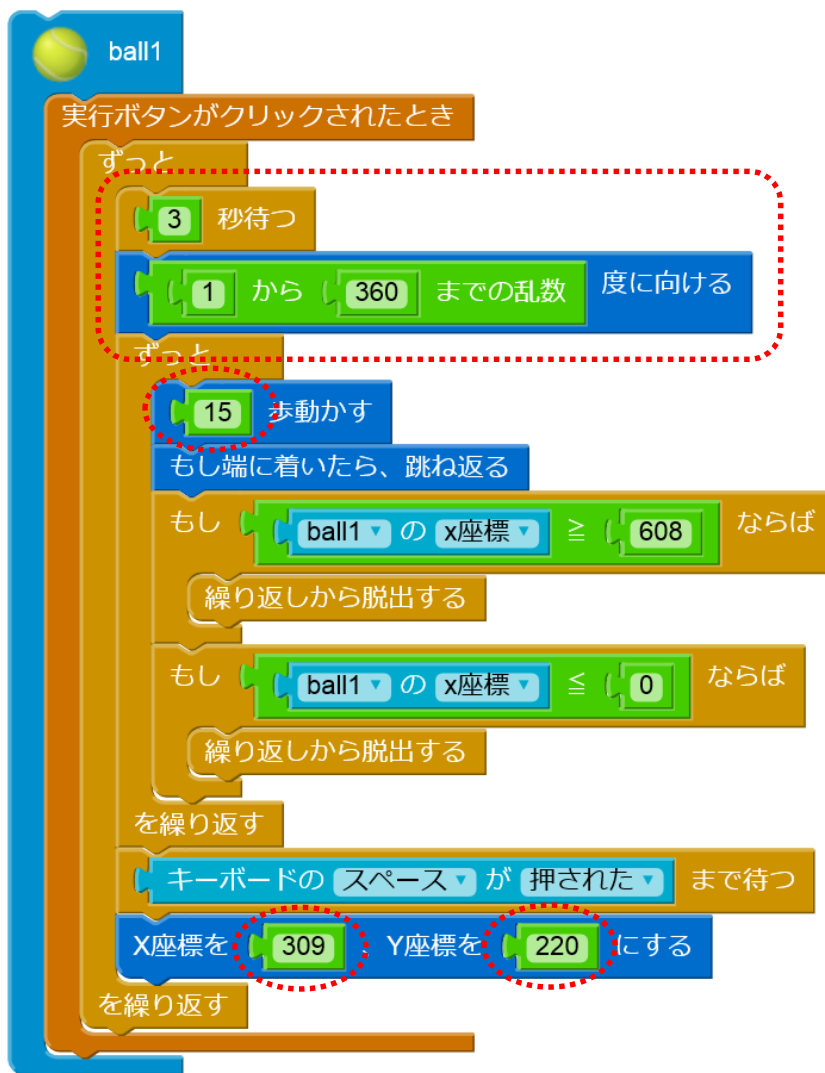


名前 ball1

X座標 309

Y座標 220

向き ↗ 225°



ベストスマウルビープログラム集^{しゅう}

著作者 = R u b y プログラミング少年団^{しょうねんだん}

代表者 = 高尾宏治^{たかおこうじ}

連絡先 = contact@smalruby.jp

公式サイト = <http://smalruby.jp>

この文書は、これからプログラミングを学びたい人であれば誰でも使ってもらえるように、この文書をコピーしたり、他の人に渡したりすることができます。それだけではなくて、この文書を参考にして別の文書を作り、それを販売することもできます。



©Ruby プログラミング少年団, 2014

この作品はクリエイティブ・コモンズ 表示 4.0 国際ライセンスの下に提供されています。

ライセンスの詳細は <http://creativecommons.org/licenses/by/4.0/deed.ja> をご確認ください。

