

米MITメディアラボ開発 ビジュアル・プログラミング環境 「Scratch」Web版version3 描画の高速化処理

令和7年4月20日
茂出木敏雄

ホームページ <http://www.bekkoame.ne.jp/~modegi/>

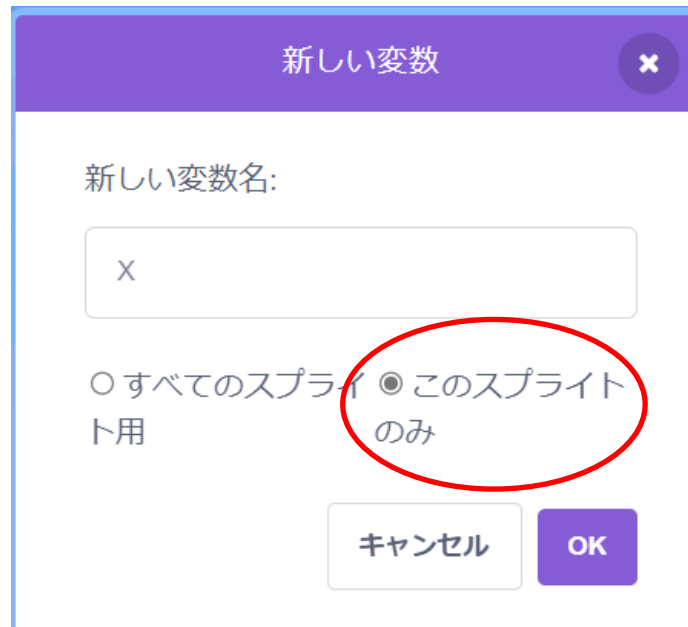
掲載プログラムの全データは上記サイトよりダウンロードできます。
Scratch_example.zip内の¥Scratch¥adv_h¥フォルダ

12. 拡張機能「ペン(2次元グラフィックス)」の高速化(1):サインカーブ(1)

第9章で紹介したサインカーブ”Scratch3_sine.sb3”につき、並列処理に対応できるように書き直したものが、次ページに示す”Scratch3_sine_single.sb3”である。

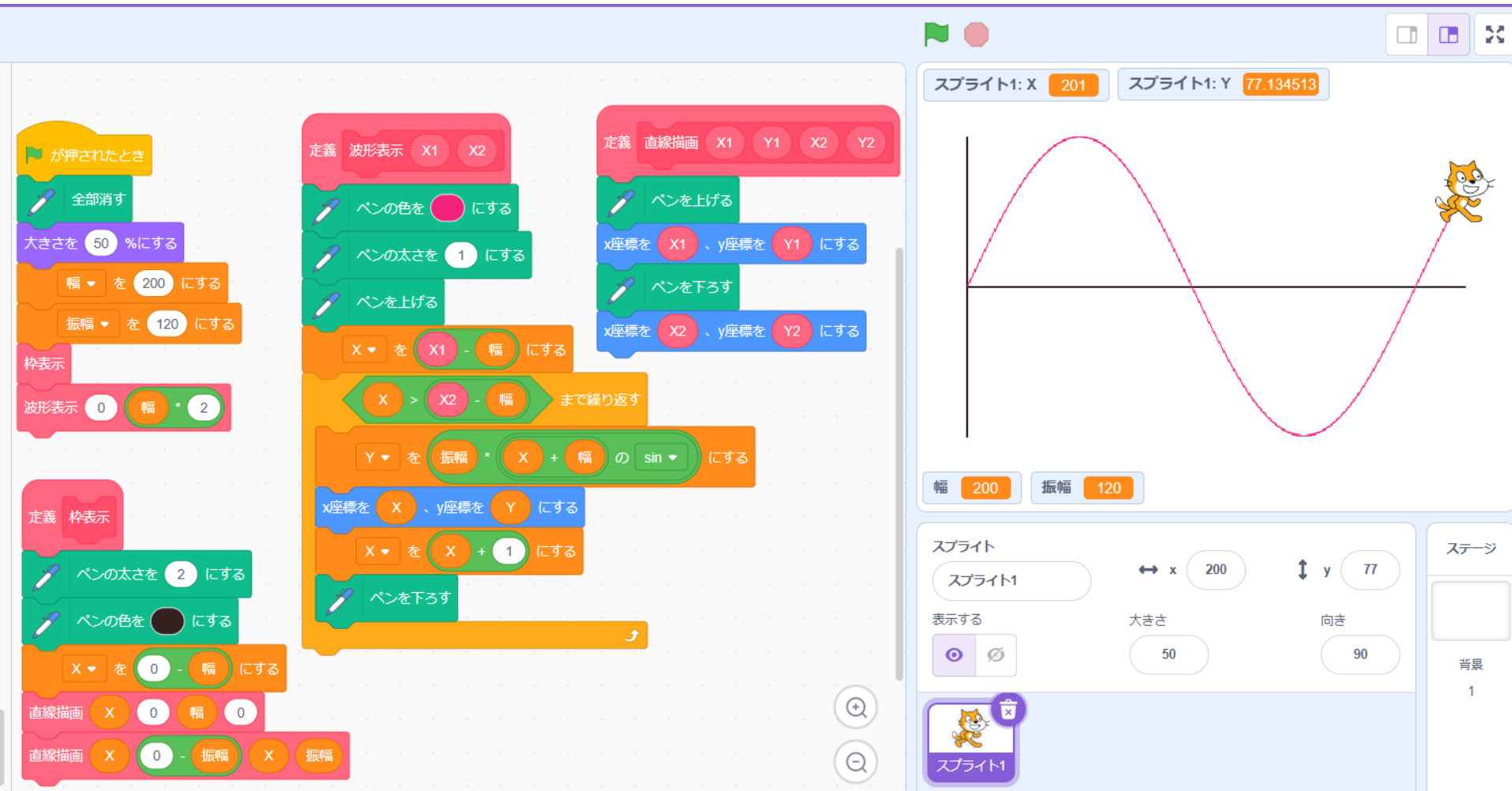
定義ブロック「波形表示」は、[X1,X2]の範囲でサインカーブの一部を描画し、描画範囲を複数個所設定して同時描画が可能になるように、マルチスレッドをサポートさせた。

本プログラムで使用する変数として、幅、振幅、Time、X、Yの5点を定義しているが、「波形表示」ブロック内で使用する変数Xと変数Yは、マルチスレッドで動いている場合、各スレッド(スプライト)ごとに異なる値になるため、変数を定義する際、以下のようにローカル変数(このスプライトのみ)に変更設定する必要がある。他の3つの変数はプログラム内で共通して使用するため、デフォルトのグローバル変数(すべてのスプライト用)にする。



The image shows the '新しい変数' (New Variable) dialog box in Scratch. The title bar is purple with a close button. Below the title, the text '新しい変数名:' (New variable name:) is followed by a text input field containing the letter 'X'. Below the input field, there are two radio button options: '○ すべてのスプライト用' (For all sprites) and '● このスプライトのみ' (For this sprite only). The second option is selected and circled in red. At the bottom, there are two buttons: 'キャンセル' (Cancel) and 'OK'.

12. 拡張機能「ペン」の高速化(1):サインカーブ(2)



The image displays a Scratch3 project titled "Scratch3_sine_single.sb3". The main workspace shows a script for drawing a sine wave. The script begins with a "when green flag clicked" event, followed by "clear all", "set size to 50%", and "set amplitude to 200". A "show ruler" block is also present. The core of the script is a "define wave drawing" block that takes X1 and X2 as inputs. It sets the pen color to red, the pen thickness to 1, and lifts the pen. It then enters a loop from X1 to X2 with a step of 1. Inside the loop, it calculates the Y coordinate as $Y = \text{amplitude} * \sin(X + \text{phase})$ and moves the pen to that position. After the loop, it lowers the pen. A "define line drawing" block is also shown, which sets the pen up, moves to (X1, Y1), and then to (X2, Y2). The right panel shows the stage with a sine wave drawn in red. The stage coordinates are X: 201, Y: 77.134513. The stage properties show a width of 200 and a height of 120. The sprite panel shows a single sprite named "sprite1" with a size of 50 and a rotation of 90 degrees.

が押されたとき

- 全部消す
- 大きさを 50 %にする
- 幅 を 200 にする
- 振幅 を 120 にする
- 枠表示
- 波形表示 0 幅 * 2

定義 枠表示

- ペンの太さを 2 にする
- ペンの色を 黒 にする
- X を 0 - 幅 にする
- 直線描画 X 0 幅 0
- 直線描画 X 0 - 振幅 X 振幅

定義 波形表示 X1 X2

- ペンの色を 赤 にする
- ペンの太さを 1 にする
- ペンを上げる
- X を X1 - 幅 にする
- X > X2 - 幅 まで繰り返す
- Y を 振幅 * X + 幅 の sin にする
- x座標を X 、y座標を Y にする
- X を X + 1 にする
- ペンを下ろす

定義 直線描画 X1 Y1 X2 Y2

- ペンを上げる
- x座標を X1 、y座標を Y1 にする
- ペンを下ろす
- x座標を X2 、y座標を Y2 にする

幅 200 振幅 120

スプライト1: X 201 スプライト1: Y 77.134513

幅 200 振幅 120

スプライト

スプライト1

表示する

大きさ 50 向き 90

ステージ

背景 1

"Scratch3_sine_single.sb3"

12. 拡張機能「ペン」の高速化(1):サインカーブ(3) プログラム部拡大

The image displays a Scratch script for drawing a sine wave. The script is organized into three main sections: initialization, a custom function for drawing a single sine wave, and a custom function for drawing a series of sine waves.

Initialization (Left):

- When the green flag is clicked, clear the stage.
- Set the size to 50%.
- Set the width to 200.
- Set the amplitude to 120.
- Define a custom function 'サイン表示' (Sine Display) with two arguments: '波形表示' (Wave Display) and '幅' (Amplitude).

Custom Function 'サイン表示' (Middle):

- Set the pen thickness to 2.
- Set the pen color to black.
- Set the x-coordinate to 0 minus the amplitude.
- Call the custom function '直線描画' (Draw Line) with arguments: x, 0, amplitude, 0.
- Call the custom function '直線描画' (Draw Line) with arguments: x, 0 minus amplitude, x, amplitude.

Custom Function 'サイン表示' (Right):

- Define a custom function '波形表示' (Wave Display) with two arguments: 'X1' and 'X2'.
- Set the pen color to pink.
- Set the pen thickness to 1.
- Set the pen up.
- Set the x-coordinate to X1 and the y-coordinate to Y1.
- Set the pen down.
- Set the x-coordinate to X2 and the y-coordinate to Y2.
- Set the x-coordinate to X1 minus the amplitude.
- Repeat the following loop until X is greater than X2 minus the amplitude:

 - Set the y-coordinate to amplitude times (X plus amplitude) times the sine of the angle.
 - Set the x-coordinate to X and the y-coordinate to Y.
 - Set the x-coordinate to X plus 1.
 - Set the pen down.

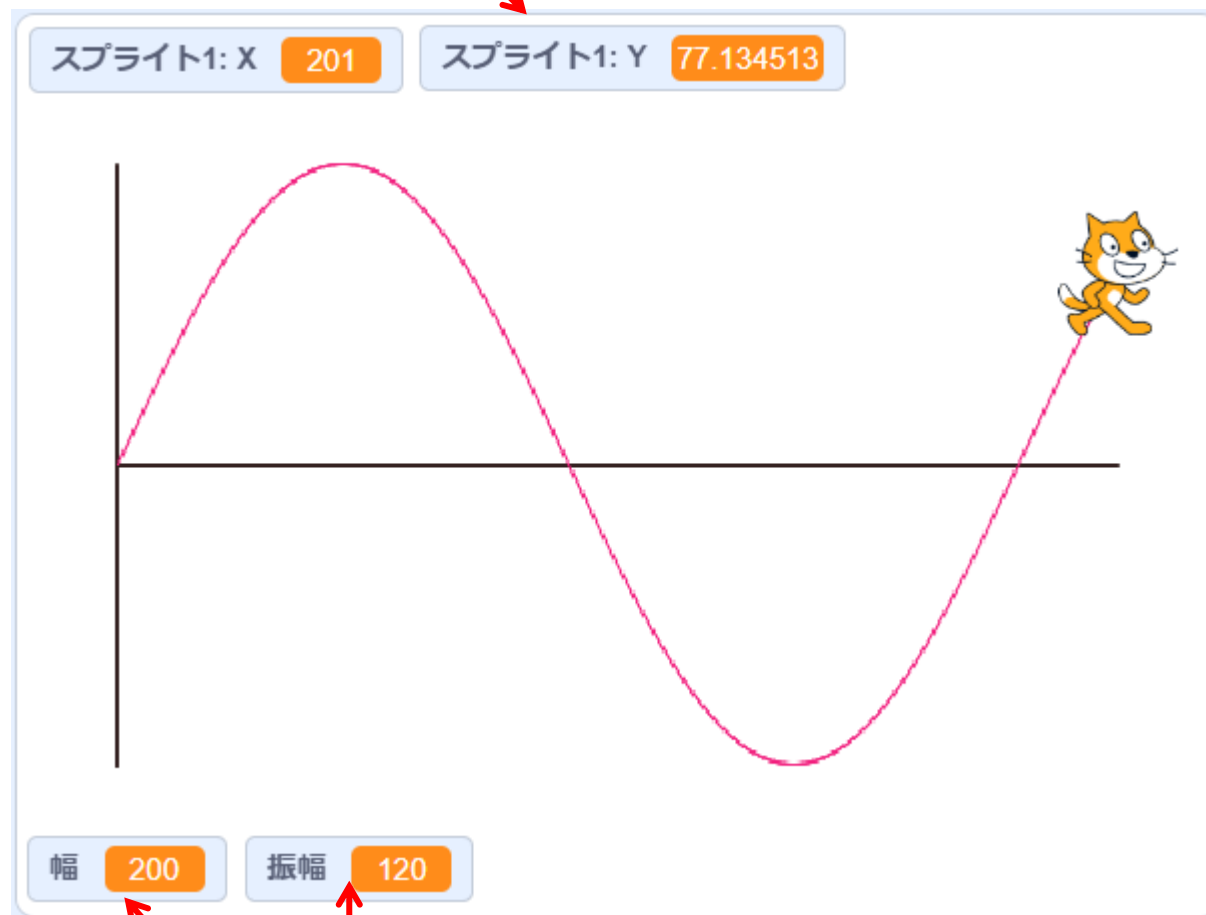
Custom Function '直線描画' (Right):

- Define a custom function '直線描画' (Draw Line) with four arguments: 'X1', 'Y1', 'X2', and 'Y2'.
- Set the pen up.
- Set the x-coordinate to X1 and the y-coordinate to Y1.
- Set the pen down.
- Set the x-coordinate to X2 and the y-coordinate to Y2.

"Scratch3_sine_single.sb3"

12. 拡張機能「ペン」の高速化(1):サインカーブ(4) 実行部

ローカル変数(各スプライト内)



"Scratch3_sine_single.sb3"

グローバル変数(プログラム全体で共有)

12. 拡張機能「ペン」の高速化(1):サインカーブ(5) 描画処理の高速化

本章ではマルチスレッドを用いた高速化手法を紹介するのが主であるが、描画処理の高速化は、後述する並列処理(マルチスレッド)を用いるより、画面更新タイミングの設定変更を行う方が圧倒的に効果的である。デフォルトでは描画コマンドを実行する度に画面更新が行われ、描画経過が可視化される設定になっているが、以下のように、定義ブロック「波形表示」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定する。その設定を行ったのが、“Scratch3_sine_singleh.sb3”である。



12. 拡張機能「ペン」の高速化(1):サインカーブ(6) 動画表示

前ページの画面更新タイミングの設定変更を行うと波形を動的に描画させることが可能になる。

三角関数の角度指定Angに、 $Ang = (X + \text{幅} + Ph) \cdot Cyc$ と、位相Ph変数を加え、 $Ph = 0$ から360までPd間隔(=2)ずつ変化させると、アニメーションを生成できる。前ページまでの事例では波形を1周期強しか表示させなかったが、変数Cyc(=3)を追加して、表示させる波形の周期も指定できるようにした。

このようなコードを追加して、波形を時間軸上に動かして、アニメーション表示させるようにした事例が、次ページに示す“Scratch3_sine_movie_single.sb3”である。

12. 拡張機能「ペン」の高速化(1):サインカーブ(7) 動画表示2

が押されたとき

大きさを 50 %にする

幅 を 200 にする

振幅 を 140 にする

Cyc を 3 にする

Pd を 2 にする

Ph を 0 にする

Ph > 360 まで繰り返す

全部消す

枠表示

波形表示 0 幅 * 2

Ph を Ph + Pd にする

定義 波形表示 X1 X2

ペンの色を 色 にする

ペンの太さを 1 にする

ペンを上げる

X を X1 - 幅 にする

X > X2 - 幅 まで繰り返す

Ang を X + 幅 + Ph * Cyc にする

Y を 振幅 * Ang の sin にする

x座標を X 、y座標を Y にする

X を X + 1 にする

ペンを下ろす

定義 枠表示

ペンの太さを 2 にする

ペンの色を 色 にする

X を 0 - 幅 にする

直線描画 X 0 幅 0

直線描画 X 0 - 振幅 X 振幅

定義 直線描画 X1 Y1 X2 Y2

ペンを上げる

x座標を X1 、y座標を Y1 にする

ペンを下ろす

x座標を X2 、y座標を Y2 にする

スプライト1: X 201

スプライト1: Y 121.243557

Ang 2280

幅 200

振幅 140

Ph 362

Pd 2

Cyc 3

スプライト

スプライト1

表示する

向き 90

大きさ 50

ステージ

背景 1

スプライト1

"Scratch3_sine_movie_single.sb3"

12. 拡張機能「ペン」の高速化(1):サインカーブ(8) マルチスレッド1

本節では4スレッド(スプライト)を並列実行させて描画を高速化する手法を紹介する。自分自身のクローン(スプライト)を3種生成して、自分自身をクローンIDを0として、クローンID:0~3の4種のクローンの各々に以下の設定で定義ブロック「波形表示」を並列実行させる。

クローン0: $0 \sim \text{幅} / 2$

クローン1: $\text{幅} / 2 \sim \text{幅}$

クローン2: $\text{幅} \sim \text{幅} \times 3 / 2$

クローン3: $\text{幅} \times 3 / 2 \sim \text{幅} \times 2$

本プログラムで使用するローカル変数として、IDとXSの2点を追加定義し、IDはローカル変数(このスプライトのみ)に変更設定する必要があるが、XSはデフォルトのグローバル変数(すべてのスプライト用)でもよい。

これらの設定を行ったのが、“Scratch3_sine_4clone.sb3”である。

12. 拡張機能「ペン」の高速化(1):サインカーブ(9) マルチスレッド2

The image displays a Scratch3 project titled "Scratch3_sine_4clone.sb3". The script is designed to draw a sine wave using multiple clones of a cat sprite. The script is organized into three main sections: initialization, cloning, and drawing.

Initialization (When Green Flag Clicked):

- 全部消す (Clear all)
- 大きさを 50 %にする (Set size to 50%)
- 幅 を 200 にする (Set width to 200)
- 振幅 を 130 にする (Set amplitude to 130)
- 枠表示 (Show bounding box)
- ID を 1 にする (Set ID to 1)
- 自分自身 のクローンを作る (Create clone of self)
- ID を 2 にする (Set ID to 2)
- 自分自身 のクローンを作る (Create clone of self)
- ID を 3 にする (Set ID to 3)
- 自分自身 のクローンを作る (Create clone of self)
- 波形表示 0 幅 / 2 (Waveform display 0 width / 2)

Cloning (When Cloned):

- XS を ID * 幅 / 2 にする (Set XS to ID * width / 2)
- 波形表示 XS XS + 幅 / 2 (Waveform display XS XS + width / 2)

Drawing (When Cloned):

- 定義 直線描画 X1 Y1 X2 Y2 (Define line drawing X1 Y1 X2 Y2)
- ペンを上げる (Lift pen)
- x座標を X1 y座標を Y1 にする (Set x coordinate to X1 y coordinate to Y1)
- ペンを下ろす (Lower pen)
- x座標を X2 y座標を Y2 にする (Set x coordinate to X2 y coordinate to Y2)
- 定義 枠表示 (Define bounding box)
- ペンの太さを 2 にする (Set pen thickness to 2)
- ペンの色を 黒 にする (Set pen color to black)
- X を 0 - 幅 にする (Set X to 0 - width)
- 直線描画 X 0 幅 0 (Line drawing X 0 width 0)
- 直線描画 X 0 - 振幅 X 振幅 (Line drawing X 0 - amplitude X amplitude)
- X を X1 - 幅 にする (Set X to X1 - width)
- X > X2 - 幅 まで繰り返す (Repeat until X > X2 - width)
- Y を 振幅 * X + 幅 の sin にする (Set Y to amplitude * X + width of sin)
- x座標を X y座標を Y にする (Set x coordinate to X y coordinate to Y)
- X を X + 1 にする (Set X to X + 1)
- ペンを下ろす (Lower pen)

The right side of the image shows the Scratch3 interface. The stage displays a sine wave drawn by the cat sprite. The stage variables show:

- スプライト1: X -99
- スプライト1: Y 128.025008
- XS 300
- 幅 200
- 振幅 130
- スプライト1: ID 3

The sprite properties show:

- スプライト1
- 表示する (Show)
- 向き 90
- 背景 1

"Scratch3_sine_4clone.sb3"

12. 拡張機能「ペン」の高速化(1):サインカーブ(10) マルチスレッド3・動画

先行ページで紹介した、“Scratch3_sine_movie_single.sb3”と同様に、前ページのマルチスレッドで表示した波形も動的に描画させることが可能である。

“Scratch3_sine_movie_single.sb3”と同様に、三角関数の角度指定Angに、 $Ang = (X + \text{幅} + Ph) \cdot Cyc$ と、位相Ph変数を加え、 $Ph = 0$ から360までPd間隔(=2)ずつ変化させると、アニメーションを生成できる。

マルチスレッドで動画表示を行う際の留意点として、生成した各クローンは処理終了後に削除するようにする。

このようなコードを追加して、波形を時間軸上に動かして、アニメーション表示させるようにした事例が、次ページに示す“Scratch3_sine_movie_4clone.sb3”である。

12. 拡張機能「ペン」の高速化(1):サインカーブ(11) マルチスレッド4・動画2

The image displays the Scratch3 project interface for a sine wave animation. The code is organized into three main sections: initialization, cloning, and drawing.

Initialization (When Green Flag Clicked):

- Set size to 30%.
- Set amplitude to 130.
- Set cycle to 3.
- Set phase to 0.
- Repeat loop: Set phase to phase + phase delta (2) until phase is greater than 360.
- Clear all.
- Clone creation.
- Set phase to phase + phase delta (2).

Cloning (When Cloned):

- Set ID to ID + amplitude / 2.
- Waveform display: XS to XS + amplitude / 2.
- Delete this clone.

Drawing (When Cloned):

- Define line drawing: X1, Y1, X2.
- Pen up.
- Set x-coordinate to X1, y-coordinate to Y1.
- Pen down.
- Set x-coordinate to X2, y-coordinate to Y2.

Cloning (When Green Flag Clicked):

- Define clone creation.
- Set pen thickness to 2.
- Set pen color to black.
- Set X to 0 - amplitude.
- Line drawing: X, 0, amplitude, 0.
- Line drawing: X, 0 - amplitude, X, amplitude.

Cloning (When Green Flag Clicked):

- Define waveform display: X1, X2.
- Set pen color to pink.
- Set pen thickness to 1.
- Pen up.
- Set X to X1 - amplitude.
- Repeat loop: Set X to X + amplitude until X is greater than X2 - amplitude.
- Set angle to X + amplitude + phase * cycle.
- Set Y to amplitude * angle's sin.
- Set x-coordinate to X, y-coordinate to Y.
- Set X to X + 1.
- Pen down.

Cloning (When Green Flag Clicked):

- Define clone creation.
- Set ID to 1.
- Self clone creation.
- Set ID to 2.
- Self clone creation.
- Set ID to 3.
- Self clone creation.
- Waveform display: 0, amplitude / 2.

Stage:

- Sprite 1: X -99, Y -112.583302, XS 300.
- Ang 2286, Ph 362, Pd 2, Cyc 3.
- Amplitude 200, Amplitude 130, Sprite 1: ID 3.

Sprite:

- Sprite 1: X -100, Y -113.
- Size 30, Direction 90.
- Sprite 1: X -100, Y -113.

"Scratch3_sine_movie_4clone.sb3"

12. 拡張機能「ペン」の高速化(1):サインカーブ(11) マルチスレッド5・動画3

The image displays two columns of Scratch code blocks. The left column is triggered by a green flag click, and the right column is triggered by a clone being created. A red arrow points from the 'Delete this clone' block in the right column to a Japanese annotation.

Left Column (Main Loop):

- が押されたとき (When green flag is clicked)
- 大きさを 30 %にする (Set size to 30%)
- 幅 ▾ を 200 にする (Set width to 200)
- 振幅 ▾ を 130 にする (Set amplitude to 130)
- Cyc ▾ を 3 にする (Set Cyc to 3)
- Pd ▾ を 2 にする (Set Pd to 2)
- Ph ▾ を 0 にする (Set Ph to 0)
- ループ: Ph > 360 まで繰り返す (Repeat while Ph > 360)
- 全部消す (Erase all)
- クローン生成 (Create clone)
- Ph ▾ を Ph + Pd にする (Set Ph to Ph + Pd)

Right Column (Clone Logic):

- クローンされたとき (When cloned)
- XS ▾ を ID * 幅 / 2 にする (Set XS to ID * width / 2)
- 波形表示 XS XS + 幅 / 2 (Show waveform XS, XS + width / 2)
- このクローンを削除する (Delete this clone)
- 定義 クローン生成 (Define clone creation)
- 枠表示 (Frame display)
- ID ▾ を 1 にする (Set ID to 1)
- 自分自身 ▾ のクローンを作る (Create clone of self)
- ID ▾ を 2 にする (Set ID to 2)
- 自分自身 ▾ のクローンを作る (Create clone of self)
- ID ▾ を 3 にする (Set ID to 3)
- 自分自身 ▾ のクローンを作る (Create clone of self)
- 波形表示 0 幅 / 2 (Show waveform 0, width / 2)

波形表示の処理終了後に当該クローンを削除する。

"Scratch3_sine_movie_4clone.sb3"

12. 拡張機能「ペン」の高速化(2):回転体・ワイヤーフレームWF(1)

4頂点断面像を3次元的に回転させ、ワイングラスを表現した3次元回転体を、X軸を中心に20度傾けて、ワイヤーフレームで描画させるようにしたものが、次ページに示す”Scratch3_glassWF_single.sb3”である。

4頂点に対応する4個の円のうち、定義ブロック「円表示」で、最上位と最下位の2種の円形をワイヤーフレームで表示させ、隣接する2個の円で形成される3種の側面を定義ブロック「側面表示」でワイヤーフレーム表示させている。

12. 拡張機能「ペン」の高速化(2):回転体・ワイヤーフレームWF(2)

Scratch3 development environment showing a wireframe glass WF (2) project.

Scripts:

- Click to run script:**
 - 全部消す
 - ペンの色を 黒 にする
 - ペンの太さを 1 にする
 - ペンを上げる
 - Ang を 20 にする
 - 円表示 150 150
 - 側面表示 150 150 0 10
 - 側面表示 0 10 -120 10
 - 側面表示 -120 10 -150 100
 - 円表示 -150 100
- Click to run script:**
 - 定義 円表示 Y R
 - A を 0 にする
 - A > 360 まで繰り返す
 - XY計算 Y R A
 - x座標を X 、y座標を Y にする
 - ペンを下ろす
 - XY計算 Y2 R2 A
 - x座標を X 、y座標を Y にする
 - ペンを上げる
 - A を A + 10 にする
 - ペンを上げる

Functions:

- 定義 XY計算 Y R A**
 - X を R * A の cos にする
 - Y を Y にする
 - Z を R * A の sin にする
 - Y を Z * Ang の sin + Y * Ang の cos にする
 - X を X * 0.7 にする
 - Y を Y * 0.7 にする
- 定義 側面表示 Y R Y2 R2**
 - A を 0 にする
 - A > 360 まで繰り返す
 - XY計算 Y R A
 - x座標を X 、y座標を Y にする
 - ペンを下ろす
 - XY計算 Y2 R2 A
 - x座標を X 、y座標を Y にする
 - ペンを上げる
 - A を A + 10 にする
 - ペンを上げる

Stage:

- Ang: 20, A: 370
- X: 70, Y: -98.667725, Z: 0
- スプライト: スプライト1
- 表示する: スプライト1
- 向き: 90
- サイズ: 50
- 位置: x: 70, y: -99

"Scratch3_glassWF_single.sb3"

12. 拡張機能「ペン」の高速化(2):回転体・WF(3)プログラム部拡大

The image displays a Scratch3 script with the following components:

- When clicked (黄色旗が押されたとき):**
 - 全部消す (Erase everything)
 - ペンの色を 黒 にする (Set pen color to black)
 - ペンの太さを 1 にする (Set pen thickness to 1)
 - ペンを上げる (Lift pen)
 - Ang を 20 にする (Set Ang to 20)
 - 円表示 150 150 (Circle display 150 150)
 - 側面表示 150 150 0 10 (Side view display 150 150 0 10)
 - 側面表示 0 10 -120 10 (Side view display 0 10 -120 10)
 - 側面表示 -120 10 -150 100 (Side view display -120 10 -150 100)
 - 円表示 -150 100 (Circle display -150 100)
- Define: 円表示 (Y, R):**
 - A を 0 にする (Set A to 0)
 - While loop: A > 360 (Repeat until A > 360)
 - XY計算 Y R A (XY calculation Y R A)
 - x座標を X, y座標を Y にする (Set x coordinate to X, y coordinate to Y)
 - ペンを下ろす (Lower pen)
 - A を A + 10 にする (Set A to A + 10)
 - ペンを上げる (Lift pen)
- Define: 側面表示 (Y, R, Y2, R2):**
 - A を 0 にする (Set A to 0)
 - While loop: A > 360 (Repeat until A > 360)
 - XY計算 Y R A (XY calculation Y R A)
 - x座標を X, y座標を Y にする (Set x coordinate to X, y coordinate to Y)
 - ペンを下ろす (Lower pen)
 - XY計算 Y2 R2 A (XY calculation Y2 R2 A)
 - x座標を X, y座標を Y にする (Set x coordinate to X, y coordinate to Y)
 - ペンを上げる (Lift pen)
 - A を A + 10 にする (Set A to A + 10)
 - ペンを上げる (Lift pen)
- Define: XY計算 (Y, R, A):**
 - X を R * A の cos にする (Set X to R * A cos)
 - Y を Y にする (Set Y to Y)
 - Z を R * A の sin にする (Set Z to R * A sin)
 - Y を Z * Ang の sin + Y * Ang の cos にする (Set Y to Z * Ang sin + Y * Ang cos)
 - X を X * 0.7 にする (Set X to X * 0.7)
 - Y を Y * 0.7 にする (Set Y to Y * 0.7)

"Scratch3_glassWF_single.sb3"

12. 拡張機能「ペン」の高速化(2):回転体・WF(4) マルチスレッド

先行ページに示す、ワイングラスを表現した3次元回転体に対して、X軸を中心に20度傾けて、ワイヤーフレーム表示させる際、最上位と最下位の2種の「円表示」と、隣接する2個の円で形成される3種の「側面表示」の5種の処理をマルチスレッドで並列実行させるようにしたものが、次ページに示す”Scratch3_glassWF_5clone.sb3”である。

12. 拡張機能「ペン」の高速化(2):回転体・WF(5) マルチスレッド2

が押されたとき

全部消す

大きさを 20 %にする

ペンの色を 黒 にする

ペンの太さを 1 にする

ペンを上げる

Ang を 20 にする

クローン生成

定義 クローン生成

ID を 1 にする

ID > 4 まで繰り返す

自分自身 のクローンを作る

ID を ID + 1 にする

円表示 -150 100

定義 XY計算 Y R A

X を R * A の cos にする

Y を Y にする

Z を R * A の sin にする

Y を Z * Ang の sin + Y * Ang の cos にする

X を X * 0.7 にする

Y を Y * 0.7 にする

クローンされたとき

もし ID = 1 なら

円表示 150 150

でなければ

もし ID = 2 なら

側面表示 150 150 0 10

でなければ

もし ID = 3 なら

側面表示 0 10 -120 10

でなければ

側面表示 -120 10 -150 100

定義 円表示 Y R

A を 0 にする

A > 360 まで繰り返す

XY計算 Y R A

x座標を X , y座標を Y にする

ペンを下ろす

A を A + 10 にする

ペンを上げる

定義 側面表示 Y R Y2 R2

A を 0 にする

A > 360 まで繰り返す

XY計算 Y R A

x座標を X , y座標を Y にする

ペンを下ろす

XY計算 Y2 R2 A

x座標を X , y座標を Y にする

ペンを上げる

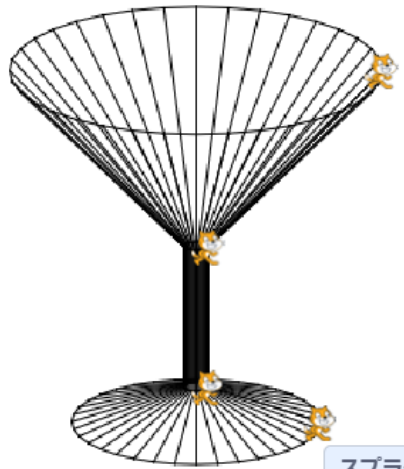
A を A + 10 にする

ペンを上げる

Ang 20

スプライト1: ID 5

スプライト1: A 370



スプライト1: Z 0

スプライト1: X 70

スプライト1: Y -98.667725

スプライト

スプライト1

表示する

サイズ 20

向き 90

スプライト1

ステージ

背景 1

"Scratch3_glassWF_5clone.sb3"

18

12. 拡張機能「ペン」の高速化(2):回転体・WF(6) マルチスレッド2 コード部

The image displays two columns of Scratch code blocks. The left column is triggered by a 'when green flag clicked' event, and the right column is triggered by a 'when cloned' event.

Left Column (Main Script):

- when green flag clicked
- 全部消す (Erase all)
- 大きさを 20 %にする (Set size to 20%)
- ペンの色を 黒 にする (Set pen color to black)
- ペンの太さを 1 にする (Set pen thickness to 1)
- ペンを上げる (Lift pen)
- Ang を 20 にする (Set Ang to 20)
- クローン生成 (Clone)
- 定義 クローン生成 (Define clone)
- ID を 1 にする (Set ID to 1)
- ID > 4 まで繰り返す (Repeat until ID > 4)
- 自分自身 のクローンを作る (Create clone of self)
- ID を ID + 1 にする (Set ID to ID + 1)
- 円表示 -150 100 (Draw circle at -150, 100)

Right Column (Cloned Script):

- クローンされたとき
- もし ID = 1 なら (If ID = 1)
- 円表示 150 150 (Draw circle at 150, 150)
- でなければ (Else)
- もし ID = 2 なら (If ID = 2)
- 側面表示 150 150 0 10 (Draw side view at 150, 150, 0, 10)
- でなければ (Else)
- もし ID = 3 なら (If ID = 3)
- 側面表示 0 10 -120 10 (Draw side view at 0, 10, -120, 10)
- でなければ (Else)
- 側面表示 -120 10 -150 100 (Draw side view at -120, 10, -150, 100)

"Scratch3_glassWF_5clone.sb3"

12. 拡張機能「ペン」の高速化(2):回転体・WF(7) 動画表示

先行ページに示す、ワイングラスを表現した3次元回転体を、シングルスレッドおよびマルチスレッドにより、各々X軸中心に20度傾けてワイヤフレーム表示する処理に対して、次ページに示すコードを追加することにより、X軸中心に、-45度から+45度まで2度刻みで、回転させるように動画表示させたものが、"Scratch3_glassWF_movie_single.sb3" および"Scratch3_glassWF_movie_5clone.sb3"である。

12. 拡張機能「ペン」の高速化(2):回転体・WF(8) 動画表示2



"Scratch3_glassWF_movie_single.sb3"



"Scratch3_glassWF_movie_5clone.sb3"

12. 拡張機能「ペン」の高速化(3):回転体・シェーディングSH(1)

4頂点断面像を3次的に回転させ、ワイングラスを表現した3次元回転体を、X軸を中心に20度傾けて、陰影処理してシェーディング表示させるようにしたものが、次ページに示す”Scratch3_glassSH_single.sb3”である。

隣接する2個の円で形成される3種の側面を定義ブロック「側面表示」でワイヤーフレーム表示させている。

12. 拡張機能「ペン」の高速化(3):回転体・シェーディングSH(2)

The image displays the Scratch3 code editor with a script for drawing a glass using the 'Pen' extension. The script is organized into three main sections: initialization, drawing the glass body, and shading.

Initialization:

- When the green flag is clicked, reset the pen (全部消す).
- Set pen color to black (ペンの色を 黒 にする).
- Set pen thickness to 1 (ペンの太さを 1 にする).
- Set pen position to (0, 0) (ペンを上げる).
- Set angle to 20 (Ang を 20 にする).
- Set side view coordinates (側面表示) to (-120, 10, -150, 100).
- Set side view coordinates (側面表示) to (0, 10, -120, 10).
- Set side view coordinates (側面表示) to (150, 150, 0, 10).

Drawing the Glass Body:

- Define a function '側面表示' (Side View) with parameters Y, R, Y2, R2.
- Define a function '基本輝度' (Basic Brightness) with parameters Y, R, Y2, R2.
- Calculate the angle A (輝度計算 A).
- Calculate the XY coordinates (XY計算 Y, R, A).
- Set the x and y coordinates (x座標を X, y座標を Y にする).
- Draw the glass body (ペンを下ろす).
- Calculate the XY coordinates (XY計算 Y2, R2, A).
- Set the x and y coordinates (x座標を X, y座標を Y にする).
- Draw the glass body (ペンを上げる).
- Set the angle A to A + 0.5 (A を A + 0.5 にする).
- Repeat the drawing process until the angle A reaches 360 degrees.

Shading:

- Define a function '輝度計算' (Brightness Calculation) with parameter A.
- Calculate the brightness V (V を Vh * A の sin にする).
- Set the pen brightness to V (ペンの 明るさ を V にする).

Stage:

- The stage shows a 3D glass with a cat sprite on it.
- The coordinates are X: 7, Y: 0, Z: 0.
- The stage is labeled 'ステージ' (Stage).

12. 拡張機能「ペン」の高速化(3):回転体・シェーディングSH(3) コード部

The image displays a Scratch3 script for a glass sphere shading extension, organized into several defined functions and a main event sequence.

Main Event Sequence (triggered by 'When clicked'):

- 全部消す (Clear all)
- ペンの色を 黒 にする (Set pen color to black)
- ペンの太さを 1 にする (Set pen thickness to 1)
- ペンを上げる (Lift pen)
- Ang を 20 にする (Set Ang to 20)
- 側面表示 -120 10 -150 100 (Set side view coordinates)
- 側面表示 0 10 -120 10 (Set side view coordinates)
- 側面表示 150 150 0 10 (Set side view coordinates)

Defined Functions:

- 定義 側面表示 Y R Y2 R2 (Side View):**
 - 基本輝度 Y R Y2 R2 (Basic brightness)
 - A を 0 にする (Set A to 0)
 - ループ: A > 360 まで繰り返す (Repeat while A > 360)
 - 輝度計算 A (Brightness calculation A)
 - XY計算 Y R A (XY calculation Y R A)
 - x座標を X 、y座標を Y にする (Set x coordinate to X, y coordinate to Y)
 - ペンを下ろす (Lower pen)
 - XY計算 Y2 R2 A (XY calculation Y2 R2 A)
 - x座標を X 、y座標を Y にする (Set x coordinate to X, y coordinate to Y)
 - ペンを上げる (Lift pen)
 - A を A + 0.5 にする (Set A to A + 0.5)
 - ペンを上げる (Lift pen)
- 定義 基本輝度 Y R Y2 R2 (Basic Brightness):**
 - Y を Y - Y2 にする (Set Y to Y - Y2)
 - もし Y < 0 なら (If Y < 0)
 - Y を 0 - Y にする (Set Y to 0 - Y)
 - X を R - R2 にする (Set X to R - R2)
 - X を X * X + Y * Y にする (Set X to X * X + Y * Y)
 - X を X の平方根 にする (Set X to square root of X)
 - Vh を Y / X にする (Set Vh to Y / X)
 - Vh を 100 + Vh の平方根 にする (Set Vh to 100 + square root of Vh)
- 定義 輝度計算 A (Brightness Calculation A):**
 - V を Vh * A の sin にする (Set V to Vh * sin of A)
 - もし V < 0 なら (If V < 0)
 - V を 0 - V にする (Set V to 0 - V)
 - ペンの 明るさ を V にする (Set pen brightness to V)
- 定義 XY計算 Y R A (XY Calculation Y R A):**
 - X を R + A の cos にする (Set X to R + cos of A)
 - Y を Y にする (Set Y to Y)
 - Z を R + A の sin にする (Set Z to R + sin of A)
 - Y を Z * Ang の sin + Y * Ang の cos にする (Set Y to Z * sin of Ang + Y * cos of Ang)
 - X を X * 0.7 にする (Set X to X * 0.7)
 - Y を Y * 0.7 にする (Set Y to Y * 0.7)

"Scratch3_glassSH_single.sb3"

12. 拡張機能「ペン」の高速化(3):回転体・SH(4) マルチスレッド

先行ページに示す、ワイングラスを表現した3次元回転体に対して、X軸を中心に20度傾けて、シェーディング表示させる際、隣接する2個の円で形成される3種の「側面表示」の5種の処理をマルチスレッドで並列実行させるようにしたものが、次ページに示す“Scratch3_glassSH_3clone.sb3”である。

12. 拡張機能「ペン」の高速化(3):回転体・SH(5) マルチスレッド2

The image displays a Scratch3 script designed to create a multi-threaded SH(5) rotation body. The script is organized into several sections:

- Initial Setup:** When the green flag is clicked, the pen is cleared, color is set to black, thickness to 1, and the pen is lifted. The angle Ang is set to 20. A clone is generated.
- Clone Definition:** A function for cloning is defined, setting the clone ID to 1, creating a clone of itself, setting the ID to 2, and creating another clone of itself.
- Stage Setup:** The stage is set to show 150 clones, with a width of 150 and a height of 10.
- Clone Loop:** A loop that iterates over the clones, calculating their position and rotation. The loop uses trigonometric functions to calculate the coordinates X , Y , and Z based on the angle Ang and the clone ID.
- Rotation Calculation:** A function for calculating the rotation V is defined, using the formula $V = V_h \cdot A \cdot \sin$.
- Pen Control:** The pen is lowered, the color is set to black, and the thickness is set to 1. The pen is then moved to the calculated coordinates X and Y .

The stage shows a 3D funnel-like shape with three cat sprites at its base. The stage settings are as follows:

- Ang: 20
- スプライト1: A: 360.5
- スプライト1: V: 0
- スプライト1: Vh: 85.50177
- スプライト1: ID: 2
- スプライト1: X: 7
- スプライト1: Y: 0
- スプライト1: Z: 0

The stage also shows a list of sprites, including a cat sprite labeled "スプライト1".

"Scratch3_glassSH_3clone.sb3"

12. 拡張機能「ペン」の高速化(3):回転体・SH(6) マルチスレッド3 コード部

The image displays a complex Scratch 3 script designed for drawing a multi-threaded SH(6) rotation body. The script is organized into several functional blocks:

- Initialization (Top Left):** Triggered by a "when clicked" event, it sets the canvas to "clear all", sets the pen color to black, the pen thickness to 1, and the pen up. It also sets the angle (Ang) to 20 and creates a clone.
- Clone Creation (Middle Left):** A "define clone creation" block that sets the clone ID to 1, creates a clone of itself, sets the ID to 2, creates another clone, and sets the side view coordinates to (150, 150, 0, 10).
- Clone Activation (Bottom Left):** A "when cloned" block that checks if the clone ID is 1. If true, it sets the side view coordinates to (-120, 10, -150, 100). Otherwise, it sets them to (0, 10, -120, 10).
- Side View Calculation (Top Middle):** A "define side view" block that takes Y, R, Y2, and R2 as inputs. It calculates the basic thickness (基本厚度) and the angle (A) to 0. It then enters a loop that continues until A is greater than 360. Inside the loop, it calculates the thickness (厚度計算), XY coordinates (XY計算), and sets the x and y coordinates (x座標を X, y座標を Y にする). It then moves the pen down, recalculates XY coordinates with Y2 and R2, moves the pen up, increments A by 0.5, and moves the pen up again.
- XY Calculation (Bottom Middle):** A "define XY calculation" block that takes Y, R, and A as inputs. It calculates X as $R \cdot A \cdot \cos$, Y as Y, Z as $R \cdot A \cdot \sin$, and then updates Y as $Z \cdot \text{Ang} \cdot \sin + Y \cdot \text{Ang} \cdot \cos$. It also updates X and Y by multiplying them by 0.7.
- Basic Thickness Calculation (Top Right):** A "define basic thickness" block that takes Y, R, Y2, and R2 as inputs. It calculates the basic thickness as $Y - Y2$. It then enters a loop that continues until Y is less than 0. Inside the loop, it updates Y as $0 - Y$, calculates X as $R - R2$, updates X as $X \cdot X + Y \cdot Y$, calculates X as the square root of X, calculates Vh as Y / X , and updates Vh as $100 \cdot Vh \cdot \text{square root}$.
- Thickness Calculation (Bottom Right):** A "define thickness calculation" block that takes A as input. It calculates V as $Vh \cdot A \cdot \sin$. It then enters a loop that continues until V is less than 0. Inside the loop, it updates V as $0 - V$ and sets the pen's brightness (明るさ) to V.

"Scratch3_glassSH_3clone.sb3"

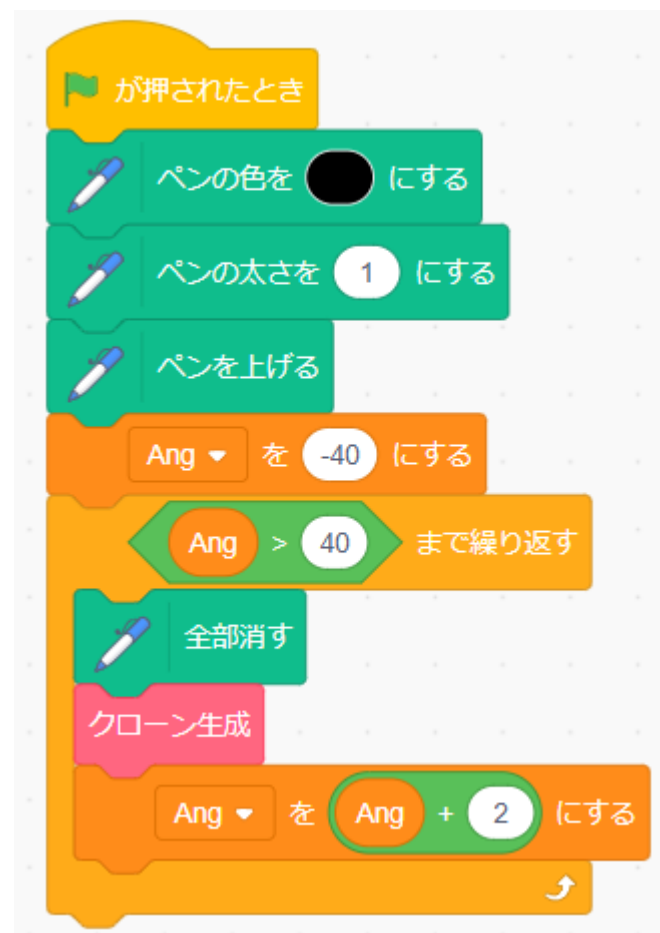
12. 拡張機能「ペン」の高速化(3):回転体・SH(7) 動画表示

先行ページに示す、ワイングラスを表現した3次元回転体を、シングルスレッドおよびマルチスレッドにより、各々X軸中心に20度傾けてシェーディング表示する処理に対して、次ページに示すコードを追加することにより、X軸中心に、-40度から+40度まで2度刻みで、回転させるように動画表示させたものが、"Scratch3_glassSH_movie_single.sb3" および "Scratch3_glassSH_movie_3clone.sb3" である。

12. 拡張機能「ペン」の高速化(3):回転体・SH(8) 動画表示2



"Scratch3_glassSH_movie_single.sb3"



"Scratch3_glassSH_movie_3clone.sb3"

12. 拡張機能「ペン」の高速化(4):シェーディング球(1)

陰影をつけた球体のイメージをピクセル単位に算出して描画する処理を、並列実行できるように作成したものが、次ページに示す“Scratch3_Sphere_single.sb3”である。

定義ブロック「球描画」は、 (X_s, Y_s) から (X_e, Y_e) の矩形領域($X_s < X_e$, $Y_s < Y_e$)で球体の一部を描画し、描画範囲を複数個所設定して同時描画が可能になるように、マルチスレッドをサポートさせた。

球体の原点を $(0,0)$ 、半径をSizeとして、ピクセル (X,Y) における球体の陰影 V (明るさ)は、 $R = (X^2 + Y^2)^{1/2}$ を算出し、 $R < \text{Size}$ の場合、 $V = 100 \cdot (\text{Size}^2 - R^2)^{1/2} / \text{Size}$ で与える。

本プログラムで使用する変数として、Size、X、Y、R、Vの5点を定義しているが、「球描画」ブロック内で使用する変数X、Y、R、Vは、マルチスレッドで動いている場合、各スレッド(スプライト)ごとに異なる値になるため、変数を定義する際、ローカル変数(このスプライトのみ)に変更設定する必要がある。変数Sizeはプログラム内で共通して使用するため、デフォルトのグローバル変数(すべてのスプライト用)にする。

12. 拡張機能「ペン」の高速化(4):シェーディング球(2)

Scratch Script Area:

```

when green flag clicked
  clear
  move pen up
  set line width to 1
  set pen color to black
  set size to 20

  draw circle 0 - size 0 - size size size

  define function: draw circle Xs Ys Xe Ye
    move pen to Ys
    while Y > Ye
      move pen to Xs
      while X > Xe
        move pen to (X * X + Y * Y)
        move pen to (R of sqrt R)
      if R < size
        move pen to (size * size - R * R of sqrt)
        move pen to (100 * V / size of floor)
        set pen brightness to V
        move pen to (X, Y)
        move pen down
        move pen up
        X = X + 1
        Y = Y + 1
    
```

Scratch Stage Area:

Sprite 1: X 21, Y 21, R 28.284271, Size 20

Sprite 1: V 8, ID 3

Sprite: Sprite1, x 6, y 19, size 10, direction 90

Stage: Background 1

"Scratch3_Sphere_single.sb3"

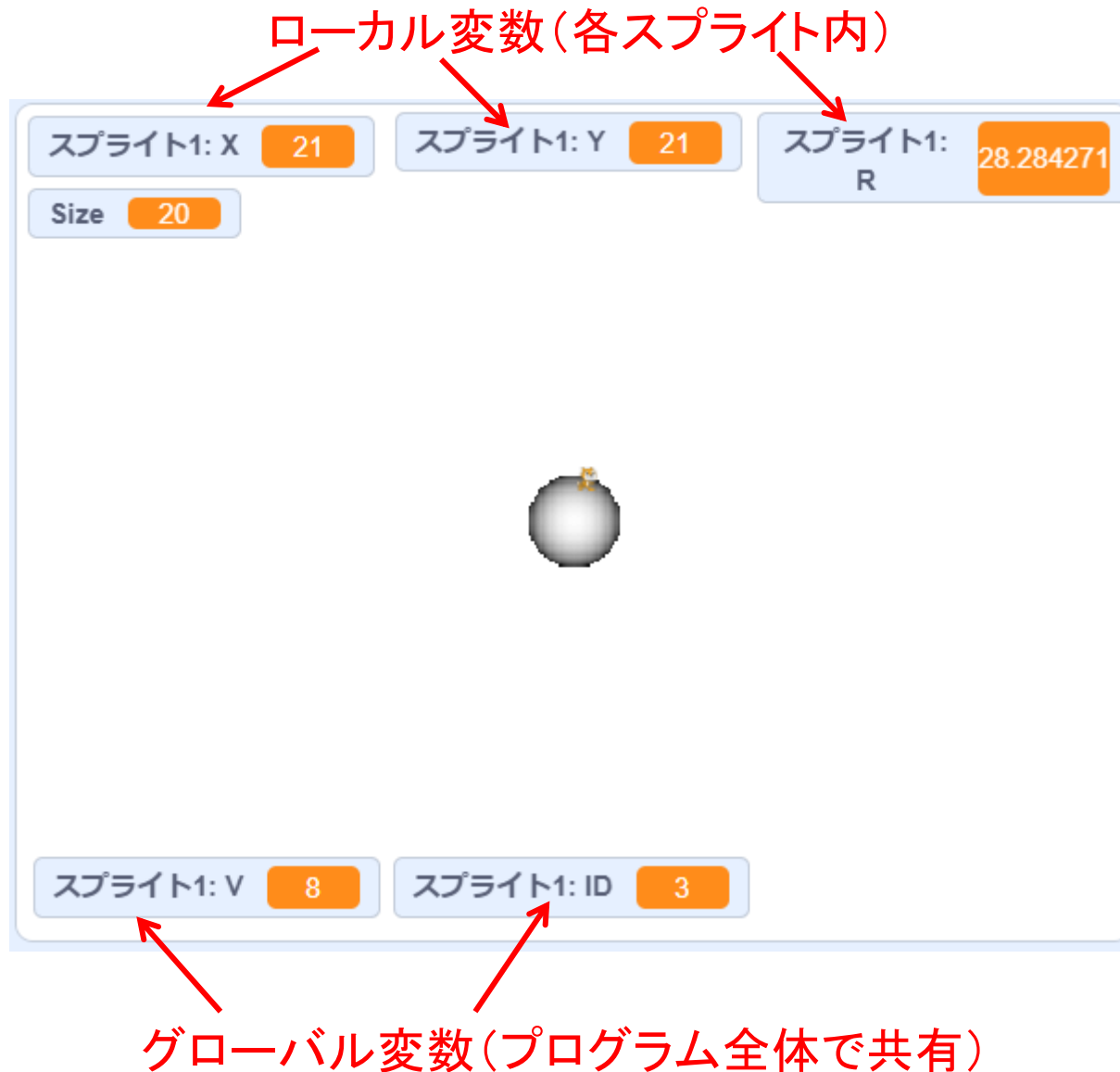
12. 拡張機能「ペン」の高速化(4):シェーディング球(3) プログラム部拡大

The image displays two sections of Scratch code. The left section, titled 'が押されたとき' (When clicked), contains initialization steps: '全部消す' (Erase everything), 'ペンを上げる' (Lift pen), 'ペンの太さを 1 にする' (Set pen thickness to 1), 'ペンの色を 黒 にする' (Set pen color to black), and 'Size を 20 にする' (Set size to 20). Below these is a '球描画' (Draw sphere) block with four 'Size' inputs, the first of which is set to 0. A red arrow points from the text '球の半径を設定' (Set sphere radius) to this first 'Size' input. The right section, titled '定義 球描画' (Define draw sphere), shows the main drawing loop. It sets 'Y' to 'Ys' and loops until 'Y' is greater than 'Ye'. Inside, it sets 'X' to 'Xs' and loops until 'X' is greater than 'Xe'. It then calculates the radius 'R' as the square root of $X^2 + Y^2$. A conditional block 'もし R < Size なら' (If R < Size then) calculates a new size 'V' as the square root of $Size^2 - R^2$, then scales it by 100 and divides by 'Size' using a '切り下げ' (Floor) block. It then sets the pen's '明るさ' (Brightness) to 'V', moves the pen to coordinates (X, Y), and lowers the pen. Finally, it increments 'X' and 'Y' by 1.

球の半径を設定

"Scratch3_Sphere_single.sb3"

12. 拡張機能「ペン」の高速化(4):シェーディング球(4) 実行部



"Scratch3_Sphere_single.sb3"

12. 拡張機能「ペン」の高速化(4):シェーディング球(5) 描画処理の高速化

本章ではマルチスレッドを用いた高速化手法を紹介するのが主であるが、描画処理の高速化は、後述する並列処理(マルチスレッド)を用いるより、画面更新タイミングの設定変更を行う方が圧倒的に効果的である。デフォルトでは描画コマンドを実行する度に画面更新が行われ、描画経過が可視化される設定になっているが、以下のように、定義ブロック「球描画」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定する。その設定を行ったのが、“Scratch3_Sphere_singleh.sb3”である。



12. 拡張機能「ペン」の高速化(4):シェーディング球(6) マルチスレッド1

本節では4スレッド(スプライト)を並列実行させて描画を高速化する手法を紹介する。自分自身のクローン(スプライト)を3種生成して、自分自身をクローンIDを0として、クローンID:0~3の4種のクローンの各々に以下の矩形範囲の設定で定義ブロック「球描画」を並列実行させる。

クローン0: (-Size, -Size) ~ (0, 0)

クローン1: (0, -Size) ~ (Size, 0)

クローン2: (-Size, 0) ~ (0, Size)

クローン3: (0, 0) ~ (Size, Size)

本プログラムで使用するローカル変数として、IDを追加定義し、IDはローカル変数(このスプライトのみ)に変更設定する必要がある。

これらの設定を行ったのが、“Scratch3_Sphere_4clone.sb3”である。

12. 拡張機能「ペン」の高速化(4):シェーディング球(7) マルチスレッド2

The image displays the Scratch3_Sphere_4clone.sb3 project interface. The main workspace shows a sphere with a gradient, representing the result of the drawing process. The script area on the left contains a multi-threaded drawing routine. The top right panel shows the sprite's properties, including its position (X: 1, Y: 1, R: 0) and size (40). The bottom right panel shows the sprite's appearance, including its name (Sprite1), position (x: 0, y: 0), size (10), and rotation (90).

Script Area:

- When clicked:**
 - Clear
 - Pen up
 - Pen thickness to 1
 - Pen color to black
 - Size to 40
 - ID to 1
 - Clone itself
 - ID to 2
 - Clone itself
 - ID to 3
 - Clone itself
- When cloned:**
 - Define: Sphere drawing (Xs, Ys, Xe, Ye)
 - Y to Ys
 - Y > Ye loop: Y to Ys, X to Xs, X > Xe loop: R to $\sqrt{X^2 + Y^2}$, R to $\sqrt{R}$. If $R < Size$, V to $\sqrt{Size^2 - R^2}$, V to $100 * V / Size$ (rounded down), Pen brightness to V, x to X, y to Y, Pen down, Pen up, X to X + 1, Y to Y + 1.
- When ID = 1:** Sphere drawing (0, 0 - Size, Size, 0)
- When ID = 2:** Sphere drawing (0 - Size, 0, 0, Size)

Sprite Panel:

- Sprite1: X: 1, Y: 1, R: 0, Size: 40
- Sprite1: V: 100, Sprite1: ID: 3
- Sprite1: x: 0, y: 0, size: 10, rotation: 90

"Scratch3_Sphere_4clone.sb3"

12. 拡張機能「ペン」の高速化(5):市松模様球(1)

前ページまでの例題では、マルチスレッドによる並列処理の効果が、定義ブロックにおける画面更新タイミングの設定変更に比べて弱く、わかりにくかった。

本例題では描画負荷の大きい動画表示にて、マルチスレッドの効果を確認する。

市松模様のテクスチャを貼り付けた球体のイメージをピクセル単位に算出して描画する処理を、次ページ”Scratch3_SphereTex_single.sb3”に示す。

P.13と同様に、描画処理は、シングルスレッドで実行させ、定義ブロック「球描画」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定している。

グローバル変数Bs(変数BhはBs/2)で、市松模様テクスチャの格子ピッチ(単位:角度)を設定できるようにしている。

球体の半径Size内に含まれる描画する画素(X,Y)ごとに、球体におけるX方向の角度AxとY方向の角度Ayを360度単位で算出する。

各々Bsで割った余りを $Ax \% Bs$, $Ay \% Bs$ とすると、

$Ax \% Bs > Bs/2 - 1$ かつ $Ay \% Bs > Bs/2 - 1$ 、または $Ax \% Bs < Bs/2$ かつ $Ay \% Bs < Bs/2$ の場合、画素(X,Y)を黒色で塗りつぶすようにした。

12. 拡張機能「ペン」の高速化(5):市松模様球(2)

The image shows a Scratch3 project titled "Scratch3_SphereTex_single.sb3". The main stage displays a sphere with a black and white checkerboard texture. The code is organized into three main sections: initialization, point drawing, and angle calculation.

Code Sections:

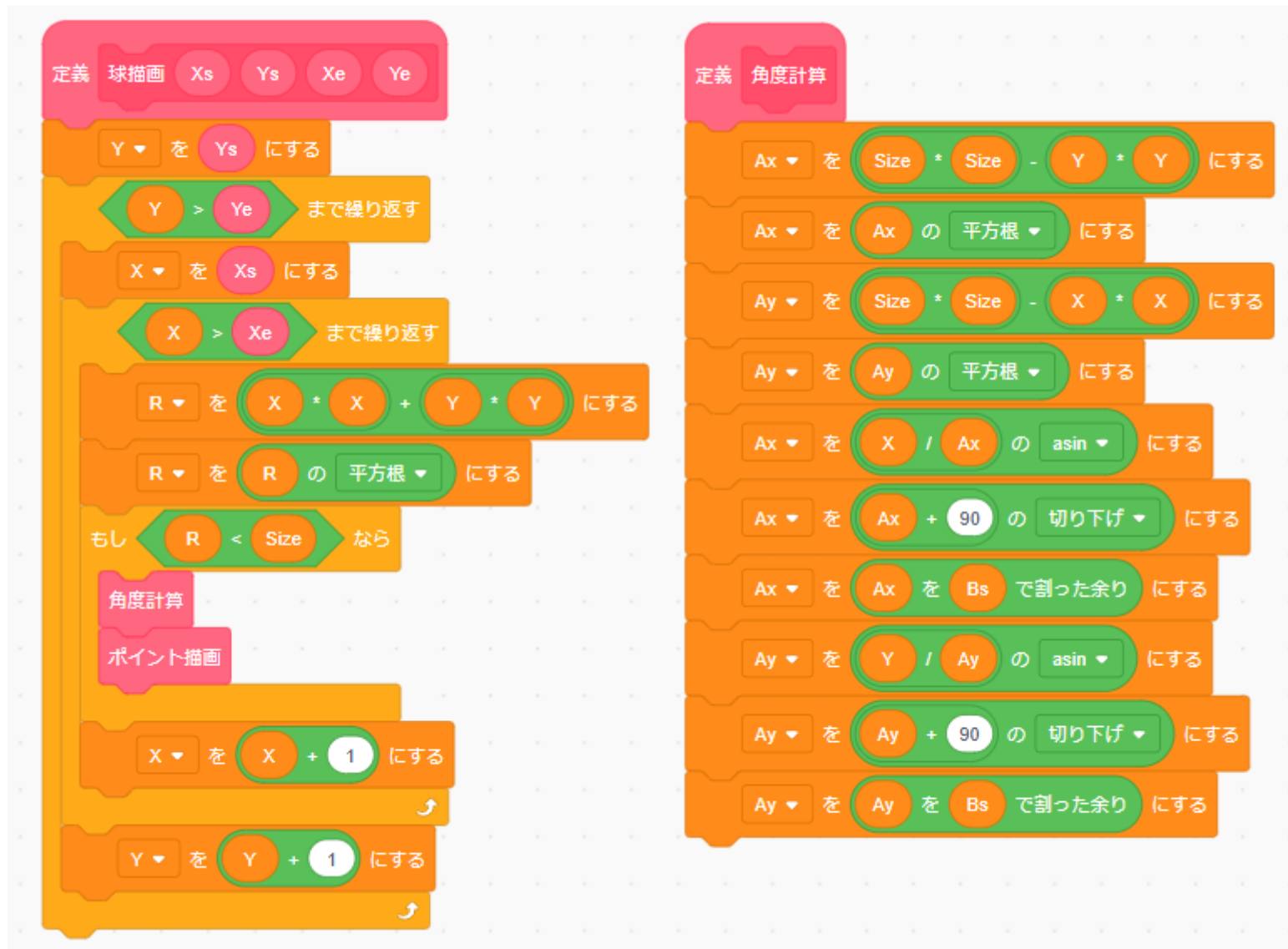
- When Green Flag Clicked:**
 - Set size to 10%.
 - Clear.
 - Set pen up.
 - Set pen size to 1.
 - Set pen color to black.
 - Set size to 100.
 - Set Bs to 20.
 - Set Bh to Bs / 2.
 - Set sphere radius to 0.
- Define: Sphere Drawing:**
 - Set Xs, Ys, Xa, Ya.
 - Loop: Y > Ys. If true, set Y = Ys, then X = Xs, then X > Xa. If true, set R = X * X + Y * Y, then R = R's square root. If R < Size, calculate angle, draw point, X = X + 1, Y = Y + 1.
- Define: Angle Calculation:**
 - Set Ax = Size * Size - Y * Y.
 - Set Ax = Ax's square root.
 - Set Ay = Size * Size - X * X.
 - Set Ay = Ay's square root.
 - Set Ax = X / Ax's asin.
 - Set Ax = Ax + 90's wrap.
 - Set Ax = Ax's remainder by Bs.
 - Set Ay = Y / Ay's asin.
 - Set Ay = Ay + 90's wrap.
 - Set Ay = Ay's remainder by Bs.
- Define: Point Drawing:**
 - Condition: Ax > Bh - 1 and Ay > Bh - 1, or Ay < Bh and Ax < Bh. If true, set pen color to black.
 - Condition: Ax > Bh - 1 and Ay < Bh, or Ay > Bh and Ax < Bh - 1. If true, set pen color to white.
 - Set x coordinate to X, y coordinate to Y.
 - Set pen down.
 - Set pen up.

Stage Properties:

- Sprite 1: X: 101, Y: 101, Size: 100, Bs: 20, Bh: 10.
- Sprite 1: Ax: 12, Ay: 18.
- Stage: x: 14, y: 99, size: 10, direction: 90.

"Scratch3_SphereTex_single.sb3"

12. 拡張機能「ペン」の高速化(5):市松模様球(3) プログラム部拡大1



"Scratch3_SphereTex_single.sb3"

12. 拡張機能「ペン」の高速化(5):市松模様球(4) プログラム部拡大2

The image displays a Scratch3 script for drawing a sphere with a checkerboard pattern. The script is divided into two main sections: initialization and a custom drawing function.

Initialization Section (triggered by a green flag click):

- Set size to 10%.
- Erase everything.
- Pen up.
- Set pen thickness to 1.
- Set pen color to black.
- Set Size to 100. (An annotation "球の半径を設定" (Set sphere radius) points to this block.)
- Set Bs to 20. (An annotation "格子ピッチを設定" (Set grid pitch) points to this block.)
- Set Bh to Bs / 2.
- Draw the sphere using a loop: repeat 0 times, Size 0, Size 0, Size, Size.

Custom Function Section (named "ポイント描画"):

- Conditional logic: If $Ax > Bh - 1$ and $Ay > Bh - 1$ or $Ay < Bh$ and $Ax < Bh$, then set pen color to black.
- Otherwise, set pen color to white.
- Set x-coordinate to X and y-coordinate to Y.
- Pen down.
- Pen up.

"Scratch3_SphereTex_single.sb3"

12. 拡張機能「ペン」の高速化(5):市松模様球(5) 実行部



"Scratch3_SphereTex_single.sb3"

12. 拡張機能「ペン」の高速化(6):市松模様球の回転(1)

市松模様のテクスチャを貼り付けた球体のイメージで、テクスチャの格子を移動させながら複数回上書き描画させることにより、回転させた球体を描画させる処理を次ページ”Scratch3_SphereTexMovie_single.sb3”に示す。

P.13と同様に、描画処理は、シングルスレッドで実行させ、定義ブロック「球描画」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定している。

グローバル変数Bs(変数BhはBs/2)で、市松模様テクスチャの格子ピッチ(単位:角度)を設定するとともに、

定義ブロック「球描画」のパラメータ(Bx,By)で、X方向のテクスチャの位相BxとY方向のテクスチャの位相Byを指定できるようにしている。

球体の半径Size内に含まれる描画する画素(X,Y)ごとに、球体におけるX方向の角度AxとY方向の角度Ayを360度単位で算出する。

テクスチャの位相(Bx,By)を考慮して、各方向の角度(Ax,Ay)をBsで割った余りを $(Ax+Bx)\%Bs$, $(Ay+By)\%Bs$ とすると、

$(Ax+Bx)\%Bs > Bs/2 - 1$ かつ $(Ay+By)\%Bs > Bs/2 - 1$ 、または $(Ax+By)\%Bs < Bs/2$ かつ

$(Ay+By)\%Bs < Bs/2$ の場合、画素(X,Y)を黒色で塗りつぶすように、

これに該当しない場合、画素(X,Y)を背景色(白色)で塗りつぶすようにした。

テクスチャの位相(Bx,By)の各々を0～Bs-1の範囲で1ずつ変化させて球体をBs回再描画させると、球体が45度方向に回転するアニメーションを表現できる。

12. 拡張機能「ペン」の高速化(6):市松模様球の回転(2)

Scratch3 スクリプトと設定のスクリーンショット。

スクリプト:

- が押されたとき**
 - 大きさを 10 %にする
 - 全部消す
 - ペンを上げる
 - ペンの太さを 1 にする
 - ペンの色を 黒 にする
 - Size を 100 にする
 - Bs を 20 にする
 - Bh を Bs / 2 にする
 - C を 0 にする
 - C > 19 まで繰り返す
 - 球描画 C C 0 - Size 0 - Size Size Size
 - C を C + 1 にする
- 定義: 球描画 Bx By Xs Ys Xa Ye**
 - Y を Ye にする
 - Y > Ye まで繰り返す
 - X を Xs にする
 - X < Xa まで繰り返す
 - R を X - X + Y - Y にする
 - R を R の平方根 にする
 - もし R < Size なら
 - 角度計算**
 - Ax を Size * Size - Y * Y にする
 - Ax を Ax の平方根 にする
 - Ay を Size * Size - X * X にする
 - Ay を Ay の平方根 にする
 - Ax を X / Ax の asin にする
 - Ax を Ax + 90 の 切り下げ にする
 - Ax を Ax を Bs で割った余り にする
 - Ay を Y / Ay の asin にする
 - Ay を Ay + 90 の 切り下げ にする
 - Ay を Ay を Bs で割った余り にする
 - 定義: 位相補正 Bx By**
 - Ax を Ax + Bx にする
 - もし Ax > Bs - 1 なら
 - Ax を Ax - Bs にする
 - Ay を Ay + By にする
 - もし Ay > Bs - 1 なら
 - Ay を Ay - Bs にする
 - ポイント描画
 - X を X + 1 にする
 - Y を Y + 1 にする

設定:

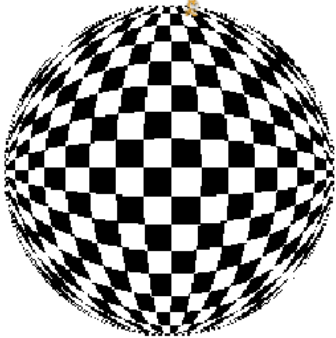
- スプライト1: X 101
- スプライト1: Y 101
- スプライト1: R 141.421356
- C 20
- Bs 20
- Bh 10
- スプライト1: Ax 11
- スプライト1: Ay 17
- スプライト1: ID 3

スプライト:

- スプライト1
- 表示する: 表示
- 大きさ: 10
- 向き: 90

ステージ:

- 背景: 1



”Scratch3_SphereTexMovie_single.sb3”

12. 拡張機能「ペン」の高速化(6):市松模様球の回転(3)プログラム部拡大

The image displays a Scratch3 script for a sphere texture movie, organized into several defined functions and a main loop.

が押されたとき (When clicked):

- 大きさを 10 %にする
- 全部消す
- ペンを上げる
- ペンの太さを 1 にする
- ペンの色を 黒 にする
- Size を 100 にする
- Bs を 20 にする
- Bh を $Bs / 2$ にする
- C を 0 にする
- もし $C > 19$ まで繰り返す
- 球描画 C C 0 - Size 0 - Size Size Size
- C を $C + 1$ にする

定義 ポイント描画 (Define Point Drawing):

- もし $Ax > Bh - 1$ かつ $Ay > Bh - 1$ または $Ay < Bh$ かつ $Ax < Bh$ なら
- ペンの色を 黒 にする
- でなければ
- ペンの色を 白 にする
- x座標を X、y座標を Y にする
- ペンを下ろす
- ペンを上げる

定義 球描画 (Define Sphere Drawing):

- Y を Ys にする
- もし $Y > Ye$ まで繰り返す
- X を Xs にする
- もし $X > Xe$ まで繰り返す
- R を $X \cdot X + Y \cdot Y$ にする
- R を R の平方根 にする
- もし $R < Size$ なら
- 角度計算
- 位相補正 Bx By
- ポイント描画
- X を $X + 1$ にする
- Y を $Y + 1$ にする

定義 角度計算 (Define Angle Calculation):

- Ax を $Size \cdot Size - Y \cdot Y$ にする
- Ax を Ax の平方根 にする
- Ay を $Size \cdot Size - X \cdot X$ にする
- Ay を Ay の平方根 にする
- Ax を X / Ax の asin にする
- Ax を $Ax + 90$ の 切り下げ にする
- Ax を Ax を Bs で割った余り にする
- Ay を Y / Ay の asin にする
- Ay を $Ay + 90$ の 切り下げ にする
- Ay を Ay を Bs で割った余り にする

定義 位相補正 (Define Phase Correction):

- Ax を $Ax + Bx$ にする
- もし $Ax > Bs - 1$ なら
- Ax を $Ax - Bs$ にする
- Ay を $Ay + By$ にする
- もし $Ay > Bs - 1$ なら
- Ay を $Ay - Bs$ にする

"Scratch3_SphereTexMovie_single.sb3"

12. 拡張機能「ペン」の高速化(6):市松模様球の回転(4)プログラム部拡大2



C=0からC=19まで変化させながら、球体を20回再描画させてアニメーション動作をさせる。

12. 拡張機能「ペン」の高速化(6):市松模様球の回転(5) 実行部



"Scratch3_SphereTexMovie_single.sb3"

12. 拡張機能「ペン」の高速化(6):市松模様球の回転(6) マルチスレッド1

球体のサイズが大きいと動画表示がぎこちなくなる。

そこで、P.14と同様に表示対象画像を4分割し、各画像を表示する際、4スレッド(スプライト)に分割して並列実行させ、描画を高速化する手法を紹介する。

自分自身のクローン(スプライト)を3種生成して、自分自身をクローンIDを0として、クローンID:0~3の4種のクローンの各々に以下の矩形範囲の設定で定義ブロック「球描画」を並列実行させる。

クローン0: (-Size, -Size) ~ (0, 0)

クローン1: (0, -Size) ~ (Size, 0)

クローン2: (-Size, 0) ~ (0, Size)

クローン3: (0, 0) ~ (Size, Size)

本プログラムで使用するローカル変数として、IDを追加定義し、IDはローカル変数(このスプライトのみ)に変更設定する必要がある。

これらの設定で実装したのが、“Scratch3_SphereTexMovie_4clone.sb3”である。

次ページに示す、マルチスレッド対応の定義ブロック「球描画M」を追加して、定義ブロック「球描画」を呼ぶ代わりに、定義ブロック「球描画M」を呼ぶようにしている。定義ブロック「球描画」以下の処理は、“Scratch3_SphereTexMovie_single.sb3”と同様である。

12. 拡張機能「ペン」の高速化(6): 市松模様球の回転(7) マルチスレッド2

The image displays two Scratch scripts. The left script is the main event handler, and the right script is a custom block named '球描画M' (Sphere Drawing M).

Main Script (Left):

- When green flag is clicked (が押されたとき)
- Set size to 10% (大きさを 10 %にする)
- Erase everything (全部消す)
- Pen up (ペンを上げる)
- Set pen thickness to 1 (ペンの太さを 1 にする)
- Set pen color to black (ペンの色を 黒 にする)
- Set size to 100 (Size を 100 にする)
- Set brush size to 20 (Bs を 20 にする)
- Set brush height to Bs / 2 (Bh を Bs / 2 にする)
- Set counter C to 0 (C を 0 にする)
- Repeat loop: while C > 19, loop (C > 19 まで繰り返す)
- Call custom block '球描画M' (球描画M)
- Increment counter C by 1 (C を C + 1 にする)

Custom Block '球描画M' (Right):

- Define block '球描画M' (定義 球描画M)
- Set ID to 1 (ID を 1 にする)
- Create clone of self (自分自身 のクローンを作る)
- Set ID to 2 (ID を 2 にする)
- Create clone of self (自分自身 のクローンを作る)
- Set ID to 3 (ID を 3 にする)
- Create clone of self (自分自身 のクローンを作る)
- Draw sphere (球描画 C C 0 0 - Size 0 - Size 0 0)
- When cloned (クローンされたとき)
 - If ID = 1 (もし ID = 1 なら)
 - Draw sphere (球描画 C C 0 0 - Size Size 0 0)
 - Otherwise (でなければ)
 - If ID = 2 (もし ID = 2 なら)
 - Draw sphere (球描画 C C 0 - Size 0 0 Size)
 - Otherwise (でなければ)
 - Draw sphere (球描画 C C 0 0 Size Size)
 - Delete this clone (このクローンを削除する)

12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(1)

市松模様テクスチャの下地の上に透明球を配置したイメージで、屈折を考慮した球体の透明感を表現した球体を描画させる処理を

次ページ”Scratch3_SphereTrD_single.sb3”に示す。

P.13と同様に、描画処理は、シングルスレッドで実行させ、定義ブロック「球描画」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定している。

グローバル変数Bs(変数BhはBs/2)で、Sizeo×2のサイズの正方形の市松模様テクスチャの格子ピッチ(単位:角度)を設定している。

球体の半径Size内(Size<Sizeo)に含まれる描画する画素(X,Y)ごとに、球体を横切る光線の深さVを算出し、整数値0～100で正規化する。

屈折を考慮して、画素(X,Y)に対して、対応するテクスチャの座標(XX,YY)を以下式で算出する。Defは屈折強度を%値で設定するパラメータで、屈折なしの場合は0にする。

$$XX=X-X\cdot V\cdot \text{Def}/13000$$

$$YY=Y-Y\cdot V\cdot \text{Def}/13000$$

テクスチャの座標(XX,YY)に対応する画素値Vt(0:黒または100:白)を取得し、画素(X,Y)に対応する輝度値V'を以下式で算出する。Transは透明度を%値で設定するパラメータで、透明度が低い場合は100未満の値を設定する。

$$V'=Vt\cdot (110-V)\cdot \text{Trans}/10000$$

12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(2)

The image displays the Scratch3 code editor with a script for drawing a sphere on a checkerboard background. The script is organized into several sections:

- Initialization:** Sets the stage background to a checkerboard pattern, initializes the pen tool, and sets the sphere's size and color.
- Drawing Loop:** A loop that draws the sphere's outline by moving the pen and drawing arcs.
- Transparency Optimization:** A block that calculates the sphere's transparency based on its size and position, ensuring it appears transparent when overlapping the checkerboard.

The right side of the image shows the Scratch3 interface with the following settings:

- Sprite1:** X: 141, Y: 141, Size: 120, Bs: 20, Bh: 10, Trans: 100, Def: 100.
- Sprite1: XX:** 0, **Sprite1: YY:** 0, **Sprite1: V:** 110, **Sprite1: ID:** 0, **Sprite1: Vt:** 100.
- Stage:** Background: 1.

"Scratch3_SphereTrD_single.sb3"

12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(3):プログラム部拡大1

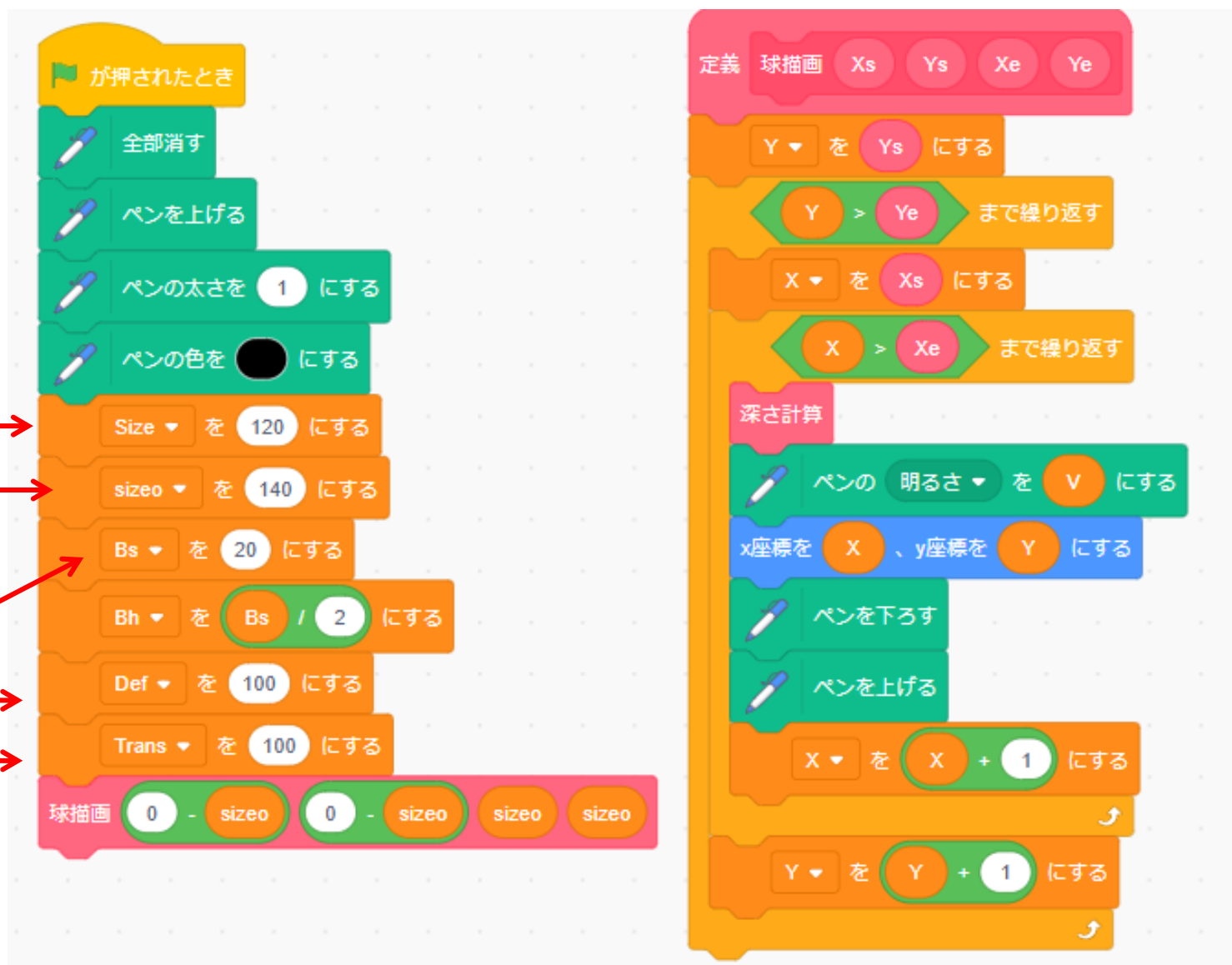
球の半径

テクスチャの
大きさ(1/2幅)

格子のピッチ

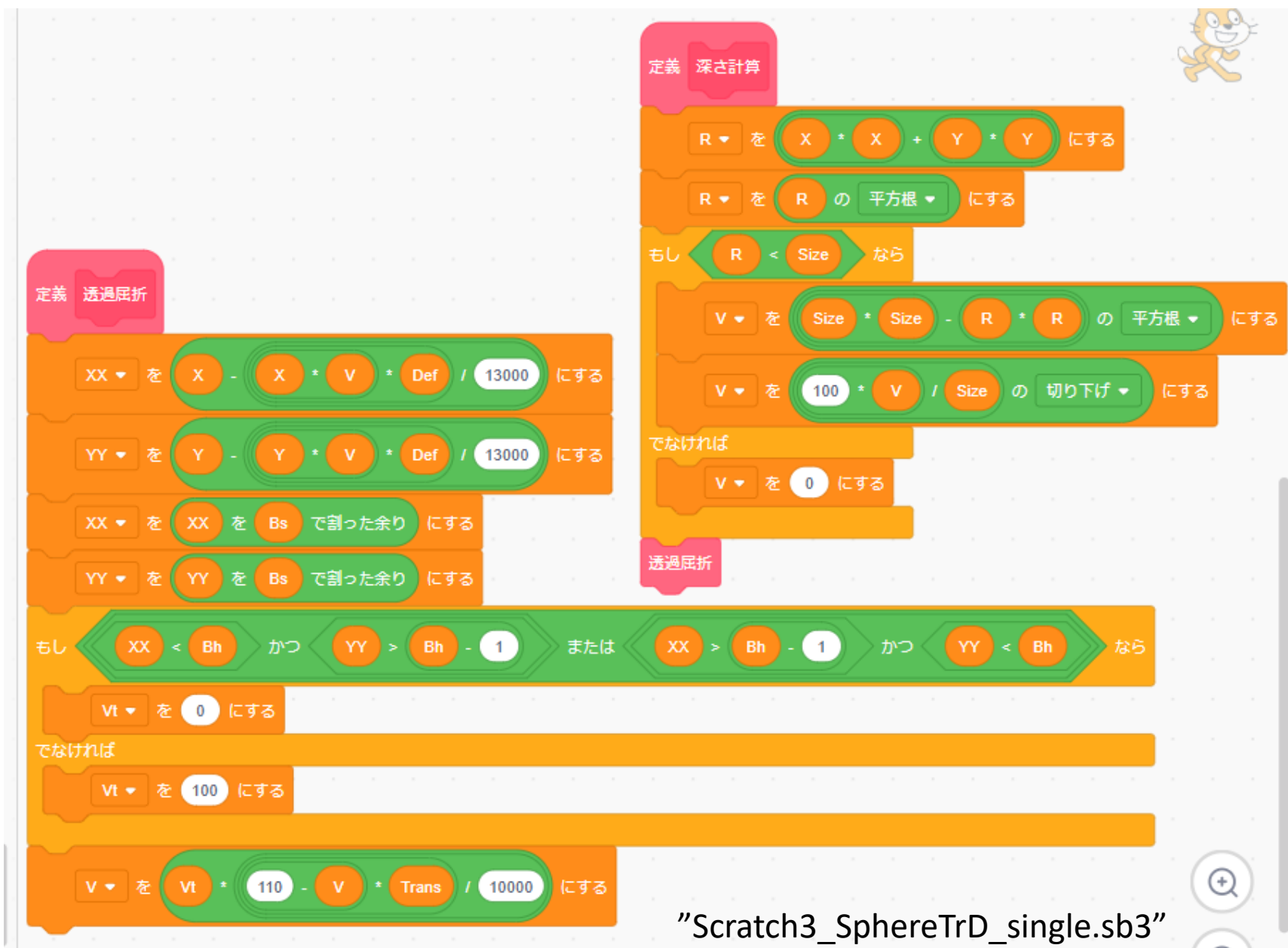
屈折強度

透明強度



"Scratch3_SphereTrD_single.sb3"

12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(4):プログラム部拡大2



The image displays a Scratch script for a sphere drawing function, organized into two main sections: '定義 透過屈折' (Definition: Refraction) and '定義 深さ計算' (Definition: Depth Calculation).

定義 透過屈折 (Definition: Refraction):

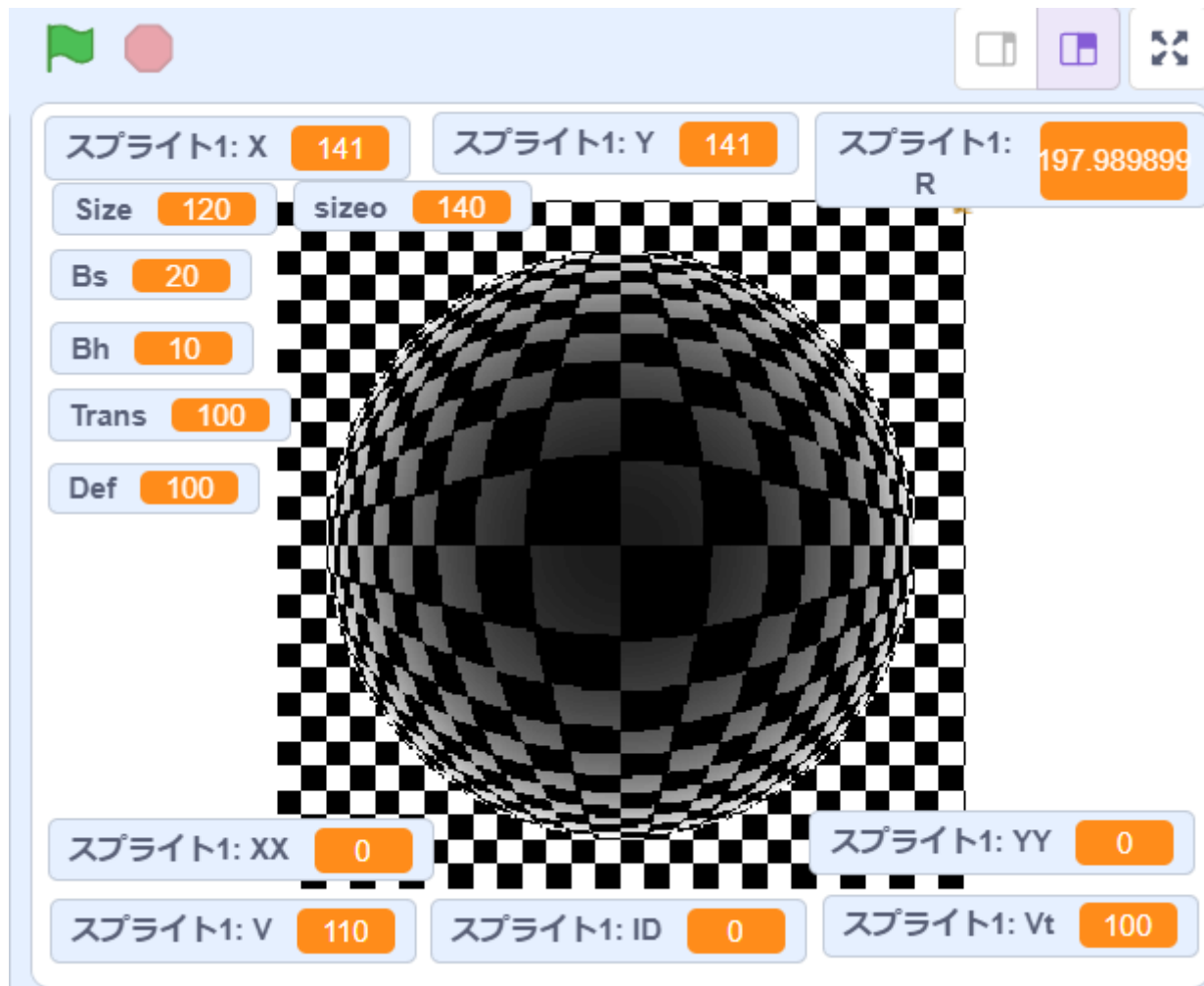
- Calculate XX and YY based on X and Y coordinates, scaled by Def and divided by 13000.
- Calculate the remainder of XX and YY when divided by Bs .
- Check if $XX < Bh$ and $YY > Bh - 1$, or if $XX > Bh - 1$ and $YY < Bh$. If true, set Vt to 0.
- Otherwise, set Vt to 100.
- Calculate V as $Vt * (110 - V * Trans) / 10000$.

定義 深さ計算 (Definition: Depth Calculation):

- Calculate R as $X^2 + Y^2$.
- Calculate the square root of R .
- If $R < Size$, calculate V as $\sqrt{Size^2 - R}$ and then V as $100 * V / Size$ (clamped).
- Otherwise, set V to 0.

The script is titled "Scratch3_SphereTrD_single.sb3".

12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(5):実行部



12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(6):マルチスレッド1

球体のサイズが大きいと動画表示がぎこちなくなる。

そこで、P.14と同様に表示対象画像を4分割し、各画像を表示する際、4スレッド(スプライト)に分割して並列実行させ、描画を高速化する手法を紹介する。

自分自身のクローン(スプライト)を3種生成して、自分自身をクローンIDを0として、クローンID:0~3の4種のクローンの各々に以下の矩形範囲の設定で定義ブロック「球描画」を並列実行させる。

クローン0: (-Sizeo, -Sizeo) ~ (0, 0)

クローン1: (0, -Sizeo) ~ (Sizeo, 0)

クローン2: (-Sizeo, 0) ~ (0, Sizeo)

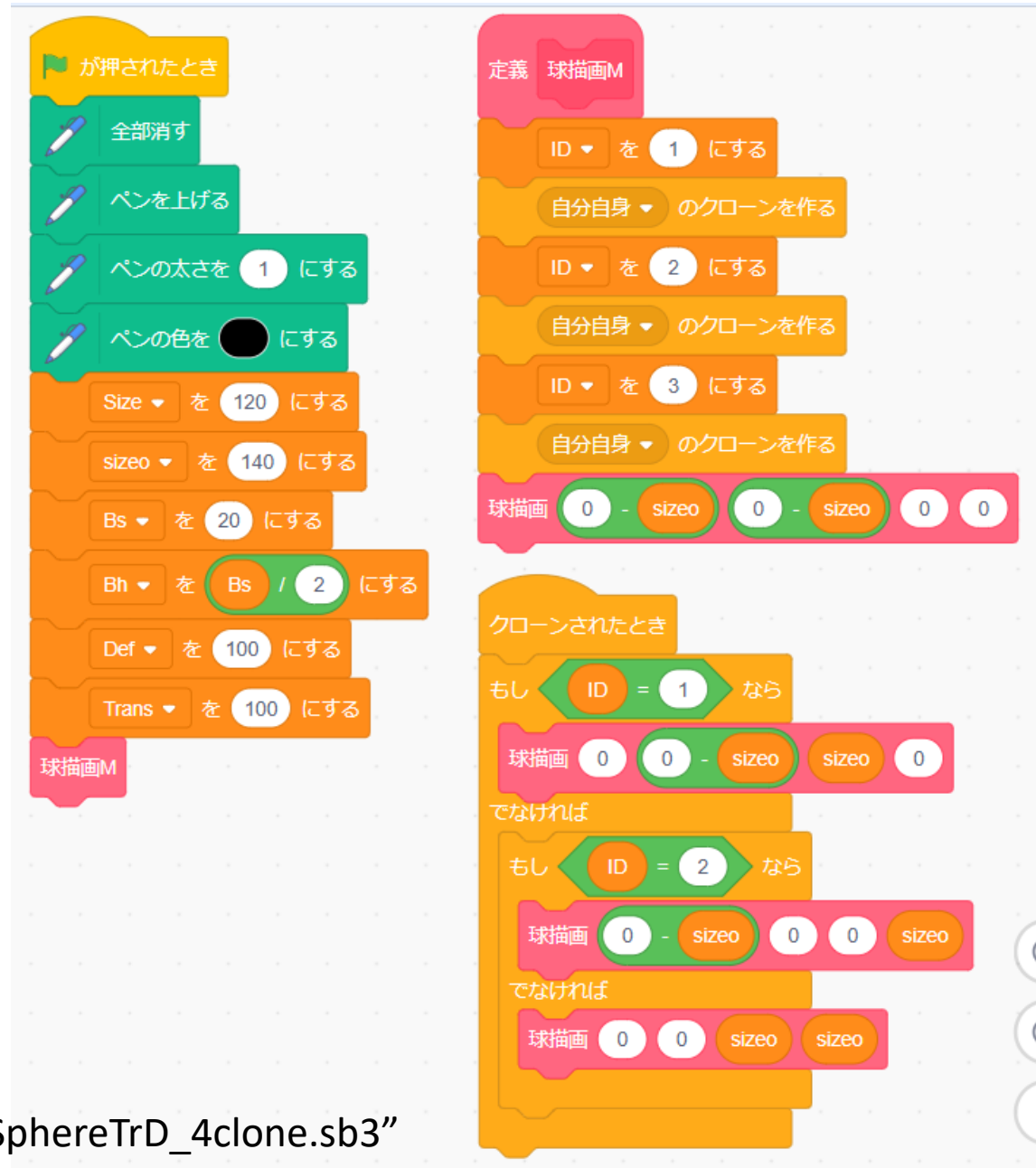
クローン3: (0, 0) ~ (Sizeo, Sizeo)

本プログラムで使用するローカル変数として、IDを追加定義し、IDはローカル変数(このスプライトのみ)に変更設定する必要がある。

これらの設定で実装したのが、“Scratch3_SphereTrD_4clone.sb3”である。

次ページに示す、マルチスレッド対応の定義ブロック「球描画M」を追加して、定義ブロック「球描画」を呼ぶ代わりに、定義ブロック「球描画M」を呼ぶようにしている。定義ブロック「球描画」以下の処理は、“Scratch3_SphereTrD_single.sb3”と同様である。

12. 拡張機能「ペン」の高速化(7):市松模様上の透明球(7):マルチスレッド2



"Scratch3_SphereTrD_4clone.sb3"

12. 拡張機能「ペン」の高速化(8):フラクタル葉(1)

フラクタル手法による自然画像生成の代表例である「バーンスレイのシダ」を計算描画する処理を、次ページ”Scratch3_barnsley_single.sb3”に示す。

P.13と同様に、描画処理は、シングルスレッドで実行させ、定義ブロック「Func」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定している。

これは座標(x_n, y_n)に対して、以下4通りの座標変換を所定の確率でランダムに選択して実行し、座標変換後の座標(x, y)をプロットするとともに、その座標(x, y)を(x_n, y_n)として同様な座標変換を繰り返す手法である。初期座標を(0,0)とし、座標変換の反復を一定回数以上繰り返すと、シダの絵が生成できる。

[座標変換1] 選択確率:1%

$$x = 0$$

$$y = 0.16 * y_n$$

[座標変換2] 選択確率:85%

$$x = 0.85 * x_n + 0.04 * y_n$$

$$y = -0.04 * x_n + 0.85 * y_n + 1.6$$

[座標変換3] 選択確率:7%

$$x = 0.20 * x_n - 0.26 * y_n$$

$$y = 0.23 * x_n + 0.22 * y_n + 1.6$$

[座標変換4] 選択確率:7%

$$x = -0.15 * x_n + 0.28 * y_n$$

$$y = 0.26 * x_n + 0.24 * y_n + 0.44$$

12. 拡張機能「ペン」の高速化(8):フラクタル葉(2)

The image shows a Scratch3 project titled "12. 拡張機能「ペン」の高速化(8):フラクタル葉(2)". The main workspace contains a script for drawing a fractal leaf using the "Pen" extension. The script is organized into several nested loops and conditional blocks.

Script Details:

- Define Wfxy:** A function that takes `num`, `x`, and `y` as arguments. It contains three conditional blocks for `num = 1`, `num = 2`, and `num = 3`, each with specific coordinate calculations for `X` and `Y`.
- Define Func:** A function that takes `iter`, `x`, and `y` as arguments. It contains a loop for `iter` from 0 to 99, with a conditional block for `R < 1` that calls `Wfxy` with different values.
- Point:** A function that takes `x` and `y` as arguments. It sets the `x` coordinate to `x` and the `y` coordinate to `y - 180`.
- When clicked:** A sequence of actions: set size to 20%, clear all, set pen color to green, and lift the pen.
- Function Call:** A call to `Func` with arguments 200000, 0, and 0.

Stage and Sprite Panel:

- Stage:** The background is a green fern-like shape. The stage coordinates are `x: 19` and `y: 97`. The stage size is `20` and the orientation is `90`.
- Sprites:** The "Sprites" panel shows a single sprite named "Sprite1" with a green color and a size of 20. The "Backgrounds" panel shows a single background named "Background 1".

"Scratch3_barnsley_single.sb3"

12. 拡張機能「ペン」の高速化(8):フラクタル葉(3) プログラム部拡大1

The image displays two Scratch code blocks for a fractal leaf program. The left block is a sub-procedure named 'WFxy' with parameters 'num', 'x', and 'y'. It contains conditional logic for 'num' values 1, 2, and 3, each with specific coordinate transformations. The right block is a sub-procedure named 'Func' with parameters 'iter', 'x', and 'y'. It includes a loop for 'iter' and a random number 'R' to determine the next sub-procedure to call ('WFxy' with 'num' 1, 2, 3, or 4). Both blocks use orange 'if' and 'if-else' blocks, green 'if' blocks for conditions, and orange 'set' blocks for variables. The 'WFxy' block ends with a 'Point' block showing 'X' and 'Y' coordinates. The 'Func' block ends with a 'loop' block.

定義 WFxy num x y

もし $num = 1$ なら

X を 0 にする

Y を $0.16 * y$ にする

でなければ

もし $num = 2$ なら

X を $x * 0.85 + y * 0.04$ にする

Y を $x * -0.04 + y * 0.85 + 1.6$ にする

でなければ

もし $num = 3$ なら

X を $x * 0.2 - y * 0.26$ にする

Y を $x * 0.23 + y * 0.22 + 1.6$ にする

でなければ

X を $x * -0.15 + y * 0.28$ にする

Y を $x * 0.26 + y * 0.24 + 0.44$ にする

Point X * 35 Y * 35

定義 Func iter x y

X を x にする

Y を y にする

iter 回繰り返す

R を 0 から 99 までの乱数 にする

もし $R < 1$ なら

WFxy 1 X Y

でなければ

もし $R < 86$ なら

WFxy 2 X Y

でなければ

もし $R < 93$ なら

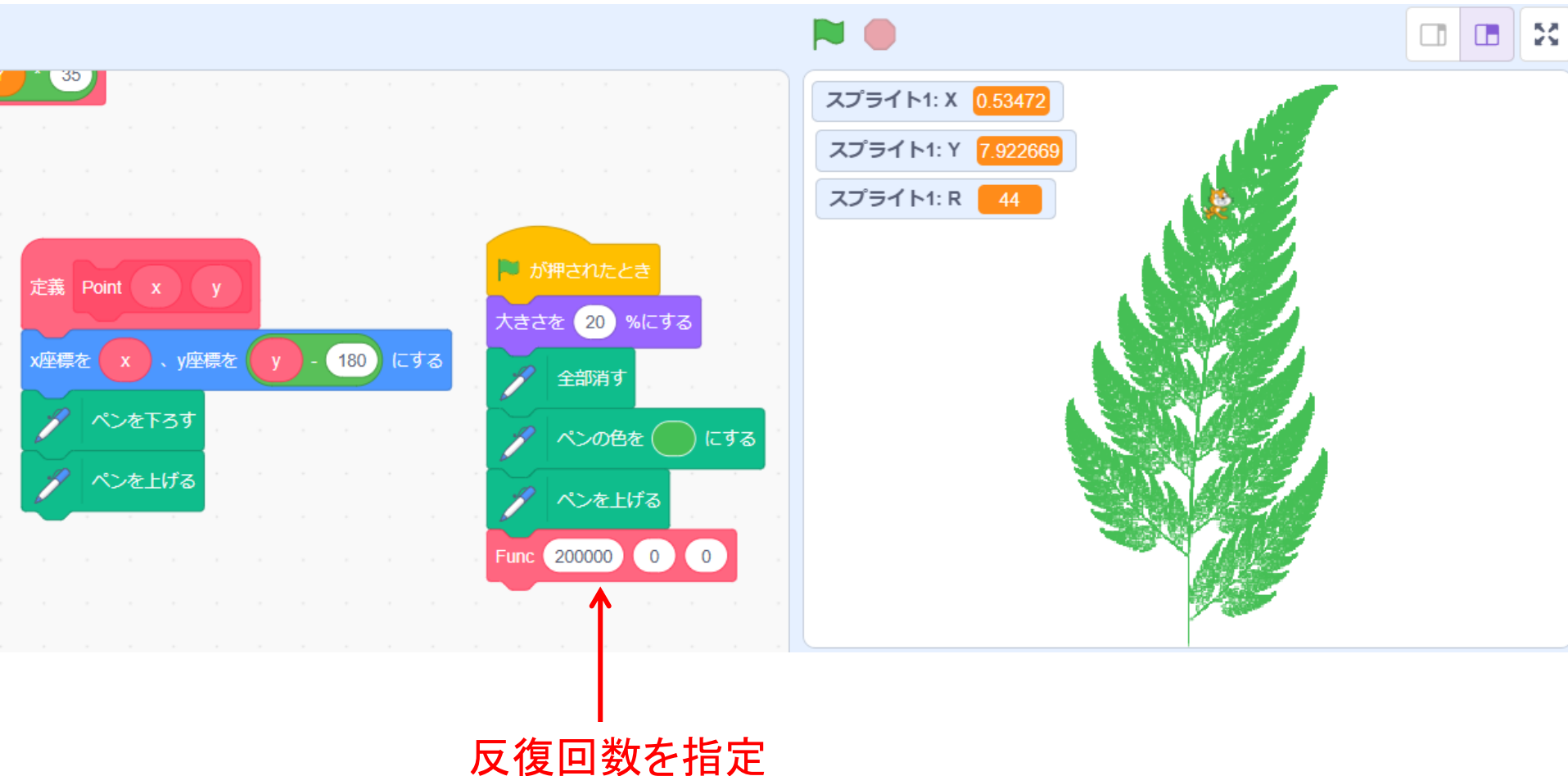
WFxy 3 X Y

でなければ

WFxy 4 X Y

"Scratch3_barnsley_single.sb3"

12. 拡張機能「ペン」の高速化(8):フラクタル葉(4) プログラム部拡大2



The image shows a Scratch3 code editor with a script for drawing a fractal leaf. The script is as follows:

- 定義 Point x y
- x座標を x 、y座標を y - 180 にする
- ペンを下ろす
- ペンを上げる

When green flag clicked:

- 大きさを 20 %にする
- 全部消す
- ペンの色を 緑 にする
- ペンを上げる
- Func 200000 0 0

A red arrow points to the 'Func' block with the value 200000, labeled 反復回数を指定 (Specify the number of repetitions).

On the right, the stage shows a green fractal leaf. The sprite properties are:

- スプライト1: X 0.53472
- スプライト1: Y 7.922669
- スプライト1: R 44

12. 拡張機能「ペン」の高速化(8):フラクタル葉(5) マルチスレッド1

反復回数により、描画処理時間が結構かかるが、以下のようにマルチスレッドにより高速化することができる。

前ページの例では、定義ブロック「Func」において、座標変換の反復回数を”200000”に設定していたが、定義ブロック「Func」は一度に実行せずに、

”Scratch3_barnsley_4single.sb3”のように、反復回数を1/4の”50000”に設定して、4回に分けて実行させることもできる。

そこで、本節では4スレッド(スプライト)で定義ブロック「Func」を並列実行させて描画を4倍に高速化する手法を紹介する。

本プログラムで使用する変数X, Y, Rは、いずれもローカル変数(このスプライトのみ)に設定する必要がある。

これらの設定を行ったのが、”Scratch3_barnsley_4clone.sb3”である。

定義ブロック「WFxy」「Func」「Point」は、”Scratch3_barnsley_single.sb3”と同じである。

12. 拡張機能「ペン」の高速化(8):フラクタル葉(6) マルチスレッド2

The image shows a Scratch3 script for drawing a fractal leaf using multiple threads. The script is organized into several sections:

- Define WFxy:** A function that takes `num`, `x`, and `y` as arguments. It contains conditional logic for `num` values 1, 2, and 3, each with specific coordinate calculations for `x` and `y`.
- Define Func:** A function that takes `iter`, `x`, and `y` as arguments. It uses a loop to generate random values for `R` and calls `WFxy` with different `num` values (1, 2, 3, 4) based on the value of `R`.
- Main Script:** Triggered by a "when green flag clicked" event, it sets the stage background to "background 1", sets the pen color to green, and sets the pen size to 20. It then calls `Func` with `50000` iterations, `0` for `x`, and `0` for `y`.
- Sprite Area:** The stage shows a green fractal leaf. The sprite area displays the coordinates for the first sprite: `Sprite1: X 0.595594`, `Sprite1: Y 5.454323`, and `Sprite1: R 57`.

"Scratch3_barnsley_4clone.sb3"

12. 拡張機能「ペン」の高速化(8):フラクタル葉(7) マルチスレッド3

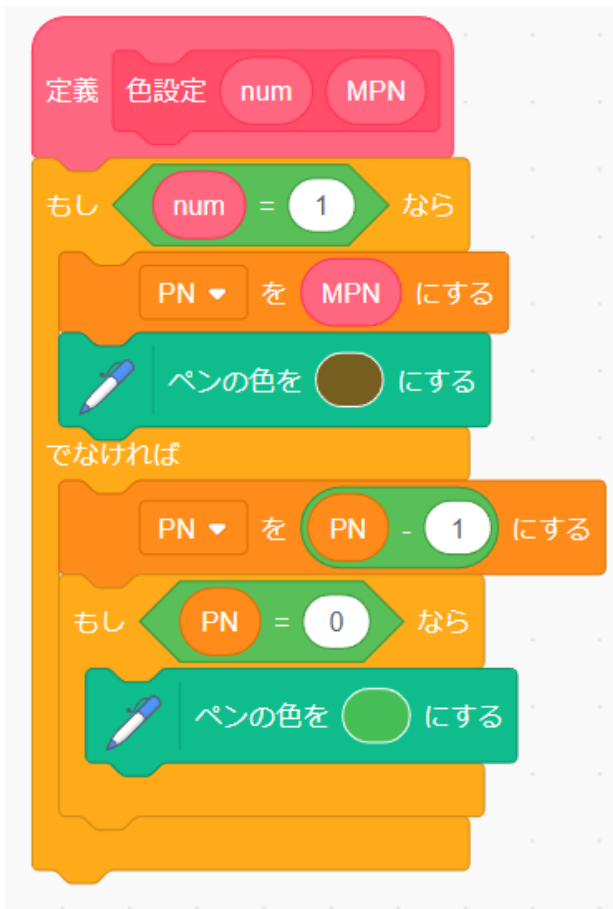


定義ブロック「WFxy」「Func」「Point」は、
"Scratch3_barnsley_single.sb3"と同じ。

"Scratch3_barnsley_4clone.sb3"

12. 拡張機能「ペン」の高速化(8):フラクタル葉(8) 枝葉の色分け

定義ブロック「Point」で枝を描画する際は茶色に着色するように色分けを行う。
定義ブロック「WFxy」を過去の指定回数MPN(設定例:30)の中で、一度でも「num=1」で実行した場合は、「枝」と認識してペンの色を茶色に設定し、それ以外は「葉」として従来通り緑色に設定する。このような機能をもたせた定義ブロック「色設定」を追加し、実装したのが、“Scratch3_barnsley2c_4clone.sb3”である。



12. 拡張機能「ペン」の高速化(8):フラクタル葉(9) 枝葉の色分け2

定義 Wfxy num x y

もし num = 1 なら

X を 0 にする

Y を $0.16 * y$ にする

でなければ

もし num = 2 なら

X を $x * 0.85 + y * 0.04$ にする

Y を $x * -0.04 + y * 0.85 + 1.6$ にする

でなければ

もし num = 3 なら

X を $x * 0.2 - y * 0.26$ にする

Y を $x * 0.23 + y * 0.22 + 1.6$ にする

でなければ

X を $x * -0.15 + y * 0.28$ にする

Y を $x * 0.26 + y * 0.24 + 0.44$ にする

色設定 num 30

Point X * 35 Y * 35

が押されたとき

大きさを 20 %にする

PN を 0 にする

全部消す

ペンの色を 緑 にする

ペンを上げる

自分自身 のクローンを作る

自分自身 のクローンを作る

自分自身 のクローンを作る

Func 50000 0 0

定義 色設定 num MPN

もし num = 1 なら

PN を MPN にする

ペンの色を 黒 にする

でなければ

PN を PN - 1 にする

もし PN = 0 なら

ペンの色を 緑 にする

スプライト1: X 1.370278

スプライト1: Y 1.733809

スプライト1: R 97

スプライト1: PN 26



スプライト

スプライト1

表示する

↔ x 48

↕ y -119

大きさ 20

向き 90

スプライト1

ステージ

背景 1

定義ブロック「Func」「Point」は、
”Scratch3_barnsley_4clone.sb3”と同じ。

64

12. 拡張機能「ペン」の高速化(9):フラクタル木(1)

フラクタル手法による自然画像生成の代表例である「フラクタルツリー(樹木)」を計算描画する処理を、次ページ”Scratch3_ftree_single.sb3”に示す。

P.13と同様に、描画処理は、シングルスレッドで実行させ、定義ブロック「DrawTree」を右クリックして、編集メニューで「画面を再描画せずに実行する」を設定している。

定義ブロック「DrawTree」は、座標(X,Y)に対して、指定角度R、 $R+dR$ 、 $R-dR$ の3方向に3本の枝を分岐・描画させる処理を、定義ブロック「DrawTree」を再帰的に3回呼ぶことにより深さN回(例:10)まで分岐実行させる方法をとっている。

3方向の枝を生成する際、中央の枝の長さは80%、左右の枝の長さは40～70%(乱数で決定)で減衰させる。中央の枝の方向は固定で、左右の枝の方向は各々+30～+70度と-30～-70度(乱数で決定)で枝分かれさせる。

12. 拡張機能「ペン」の高速化(9):フラクタル木(2)

The image displays a Scratch3 project titled "Scratch3_ftree_single.sb3". The script area on the left contains a recursive function `DrawTree` and a main execution block. The `DrawTree` function is defined with parameters `X`, `Y`, `R`, `L`, and `N`. It includes a loop that calls itself with updated parameters to draw a fractal tree. The main script starts with a "when green flag clicked" event, followed by "clear", "pen up", "set pen color to green", and a call to `DrawTree` with initial values: `0`, `0`, `90`, `100`, and `10`.

The right side of the image shows the stage with a green fractal tree. The tree is composed of many small green branches and leaves. A small orange cat sprite is visible at the base of the tree. The stage background is white.

Stage variables and sprite information:

- スプライト1: PX: 33.101098
- スプライト1: PY: -48.184128
- スプライト1: X: 0
- スプライト1: Y: -100
- スプライト: スプライト1
- 表示する: ☒ ☐
- 大きさ: 20
- 向き: 90
- ステージ: 背景 1

"Scratch3_ftree_single.sb3"

12. 拡張機能「ペン」の高速化(9):フラクタル木(3) プログラム部拡大

The image displays a Scratch script for drawing a fractal tree. The script begins with a 'when green flag is clicked' event block, followed by 'set size to 20%', 'clear', 'pen up', and 'set pen color to green'. The main action is a 'DrawTree' block with parameters 0, 0, 90, 100, and 10. A red arrow points to the final parameter '10' with the text '再帰処理の深さを指定' (Specify the depth of the recursive process). The 'DrawTree' block is defined as follows:

定義 直線描画 X Y R L Sx Sy Ox Oy

- X を $X + L * R$ の $\cos$ にする
- Y を $Y - L * R$ の $\sin$ にする
- x座標を $X * Sx + Ox$ 、y座標を $Y * Sy + Oy$ にする
- ペンを下ろす
- x座標を $X * Sx + Ox$ 、y座標を $Y * Sy + Oy$ にする
- ペンを上げる

12. 拡張機能「ペン」の高速化(9):フラクタル木(4) プログラム部拡大2

The image shows a Scratch script for a function named `DrawTree`. The function takes parameters `X`, `Y`, `R`, `L`, and `N`. It uses a loop to draw a fractal tree by recursively calling itself. The script includes blocks for updating coordinates, drawing lines, and calculating random values for branch length and angle.

```
定義 DrawTree X Y R L N
もし N > 1 なら
  X を X + L * R の cos にする
  Y を Y - L * R の sin にする
  DrawTree X Y R L * 0.8 N - 1
  PX を X + L * 0.5 * R の cos にする
  PY を Y - L * 0.5 * R の sin にする
  DrawTree PX PY R + 30 から 70 までの乱数 L * 0.4 から 0.7 までの乱数 N - 1
  PX を X + L * 0.5 * R の cos にする
  PY を Y - L * 0.5 * R の sin にする
  DrawTree PX PY R - 30 から 70 までの乱数 L * 0.4 から 0.7 までの乱数 N - 1
直線描画 X Y R L 0.8 -0.8 50 -170
```

”Scratch3_ftree_single.sb3”

12. 拡張機能「ペン」の高速化(9):フラクタル木(5) 枝葉の色分け

定義ブロック「直線描画」で枝を描画する際は茶色に着色するように色分けを行う。
定義ブロック「DrawTree」で指定する深さNの値が、3を超える場合は、「枝」と認識してペンの色を茶色に設定し、それ以外は「葉」として従来通り緑色に設定する。
併せて、深さNの値により、枝の太さも変化させ、Nが3を超える場合は、ペンの太さをN/3画素に設定し、それ以下の場合はペンの太さを1画素に設定して描画する。
このような機能をもたせた定義ブロック「色設定」を追加し、実装したのが、“Scratch3_ftree2c_single.sb3”である。

The image displays the Scratch3_ftree2c_single.sb3 project interface. On the left, the code editor shows a series of nested blocks for drawing a fractal tree. The main script starts with a 'when green flag clicked' event, followed by a 'set size to 20%' block, a 'clear' block, and a 'pen up' block. The 'DrawTree' block is called with parameters (0, 0, 90, 100, 10). The 'DrawTree' block itself is a custom block that takes parameters X, Y, R, L, and N. It contains a conditional block 'if N > 1 then' which calls 'DrawTree' with parameters (X + L * R * cos, Y - L * R * sin, R, L * 0.8, N - 1). This is followed by a 'set PX to X + L * 0.5 * R * cos' and 'set PY to Y - L * 0.5 * R * sin' block, and then another 'DrawTree' call with parameters (PX, PY, R + 30 to 70 random, L * 0.4 to 0.7 random, N - 1). The 'DrawTree' block also includes a 'set color to N' block and a 'draw line' block with parameters X, Y, R, L, 0.8, -0.8, 50, -170. On the right, the stage shows a tree with a brown trunk and green foliage. The 'Sprites' panel on the right shows the tree's position: PX: 41.024877, PY: -56.657392, X: 0, Y: -100. The 'Stage' panel at the bottom shows the tree's position: x: 50, y: -90.

定義ブロック「直線描画」は、
”Scratch3_ftree_single.sb3”と同じ。

12. 拡張機能「ペン」の高速化(9):フラクタル木(6) プログラム部拡大

The image displays two Scratch code blocks for a fractal tree drawing function. The main block on the left is a 'Define' block for 'DrawTree' with parameters X, Y, R, L, and N. It contains a conditional block 'もし N > 1 なら' (If N > 1 then) which branches into two paths. The first path calculates new X and Y coordinates using trigonometric functions and recursive calls to 'DrawTree'. The second path calculates new X and Y coordinates using random numbers and recursive calls to 'DrawTree'. The second block on the right is a 'Define' block for '色設定' (Color Setting) with parameter N. It contains a conditional block 'もし N > 3 なら' (If N > 3 then) which branches into two paths. The first path sets the pen color to brown and the pen thickness to N / 3. The second path sets the pen color to green and the pen thickness to 1. Both blocks end with a '直線描画' (Draw Line) block.

```
定義 DrawTree X Y R L N
もし N > 1 なら
  X を X + L * R の cos にする
  Y を Y - L * R の sin にする
  DrawTree X Y R L * 0.8 N - 1
  PX を X + L * 0.5 * R の cos にする
  PY を Y - L * 0.5 * R の sin にする
  DrawTree PX PY R + 30 から 70 までの乱数 L * 0.4 から 0.7 までの乱数 N - 1
  PX を X + L * 0.5 * R の cos にする
  PY を Y - L * 0.5 * R の sin にする
  DrawTree PX PY R - 30 から 70 までの乱数 L * 0.4 から 0.7 までの乱数 N - 1
色設定 N
直線描画 X Y R L 0.8 -0.8 50 -170
```

定義 色設定 N

もし N > 3 なら

ペンの色を 茶色 にする

でなければ

ペンの色を 緑 にする

もし N > 3 なら

ペンの太さを N / 3 にする

でなければ

ペンの太さを 1 にする

直線描画 X Y R L 0.8 -0.8 50 -170

"Scratch3_ftree2c_single.sb3"

12. 拡張機能「ペン」の高速化(9):フラクタル木(7) マルチスレッド1

定義ブロック「DrawTree」は、座標(X,Y)に対して、指定角度R、R+dR、R-dRの3方向に3本の枝を分岐・描画させる処理を、定義ブロック「DrawTree」を再帰的に3回呼ぶことにより深さN回(例:10)まで分岐実行させる方法をとっている。

そこで、1段下の深さN-1を指定し、再帰的に呼ばれる3種の定義ブロック「DrawTree」をマルチスレッドで並列実行させると、理論上、描画処理を3倍に高速化できる。

具体的には、シングルスレッドN=10で、drawTree(0,0,90,100,10)を実行させると、drawTree(0,-100,90,80,9)、drawTree(0,-50,140,50,9)、drawTree(0,-50,40,50,9)の3種の定義ブロックがN=9で順次実行される。

そこで、自分自身のクローン(スプライト)を2種生成して、自分自身をクローンIDを0として、クローンID:0~2の3種のクローンの各々に、以下のパラメータ設定で定義ブロック「DrawTree」を並列実行させる。

クローン0: drawTree(0,-50,40,50,9)

クローン1: drawTree(0,-100,90,80,9)

クローン2: drawTree(0,-50,140,50,9)

本プログラムで使用するローカル変数として、IDを追加定義し、IDはローカル変数(このスプライトのみ)に変更設定する必要がある。

これらの設定で実装したのが、“Scratch3_ftree2c_3clone.sb3”である。

12. 拡張機能「ペン」の高速化(9):フラクタル木(8) マルチスレッド2

The image displays a Scratch script for drawing a fractal tree using multiple threads. The script is organized into three main sections: a definition block, a main execution block, and a cloning block.

Definition Block (Define DrawTree):

- Parameters: X, Y, R, L, N
- Condition: If $N > 1$ and $N < 10$, then:
- Calculate $PX = X + L \cdot R \cdot \cos$
- Calculate $PY = Y - L \cdot R \cdot \sin$
- Call `DrawTree PX PY R + 30 から 70 までの乱数 L * 0.4 から 0.7 までの乱数 N - 1`
- Calculate $PX = X + L \cdot 0.5 \cdot R \cdot \cos$
- Calculate $PY = Y - L \cdot 0.5 \cdot R \cdot \sin$
- Call `DrawTree PX PY R - 30 から 70 までの乱数 L * 0.4 から 0.7 までの乱数 N - 1`
- Set color to N
- Draw line from X, Y to PX, PY with thickness 0.8, color -0.8, and length 50.

Main Execution Block (When Green Flag Clicked):

- Set size to 20%.
- Clear all.
- Pen up.
- Set pen color to green.
- Set ID to 1.
- Clone self.
- Set ID to 2.
- Clone self.
- Call `DrawTree 0 -50 40 50 9`
- Call `DrawTree 0 0 90 100 10`

Cloning Block (When Cloned):

- Condition: If ID = 1, then:
- Call `DrawTree 0 -100 90 80 9`
- Else:
- Call `DrawTree 0 -50 140 50 9`

Stage Properties:

- Sprite 1: PX 30.567981, PY -47.680549, X 0, Y -100, ID 2
- Stage: Sprite 1, x 50, y -90

定義ブロック「直線描画」「色設定」は、
”Scratch3_ftree2c_single.sb3”と同じ。

”Scratch3_ftree2c_3clone.sb3”

12. 拡張機能「ペン」の高速化(9):フラクタル木(9) プログラム部拡大

定義 DrawTree X Y R L N

もし $N > 1$ かつ $N < 10$ なら

X を $X + L \cdot R$ の $\cos$ にする

Y を $Y - L \cdot R$ の $\sin$ にする

DrawTree X Y R $L \cdot 0.8$ $N - 1$

PX を $X + L \cdot 0.5 \cdot R$ の $\cos$ にする

PY を $Y - L \cdot 0.5 \cdot R$ の $\sin$ にする

DrawTree PX PY $R + 30$ から 70 までの乱数 $L \cdot 0.4$ から 0.7 までの乱数 $N - 1$

PX を $X + L \cdot 0.5 \cdot R$ の $\cos$ にする

PY を $Y - L \cdot 0.5 \cdot R$ の $\sin$ にする

DrawTree PX PY $R - 30$ から 70 までの乱数 $L \cdot 0.4$ から 0.7 までの乱数 $N - 1$

色設定 N

直線描画 X Y R L 0.8 -0.8 50 -170

が押されたとき

大きさを 20 %にする

全部消す

ペンを上げる

ペンの色を 緑 にする

ID を 1 にする

自分自身 のクローンを作る

ID を 2 にする

自分自身 のクローンを作る

DrawTree 0 -50 40 50 9

DrawTree 0 0 90 100 10

クローンされたとき

もし ID = 1 なら

DrawTree 0 -100 90 80 9

でなければ

DrawTree 0 -50 140 50 9

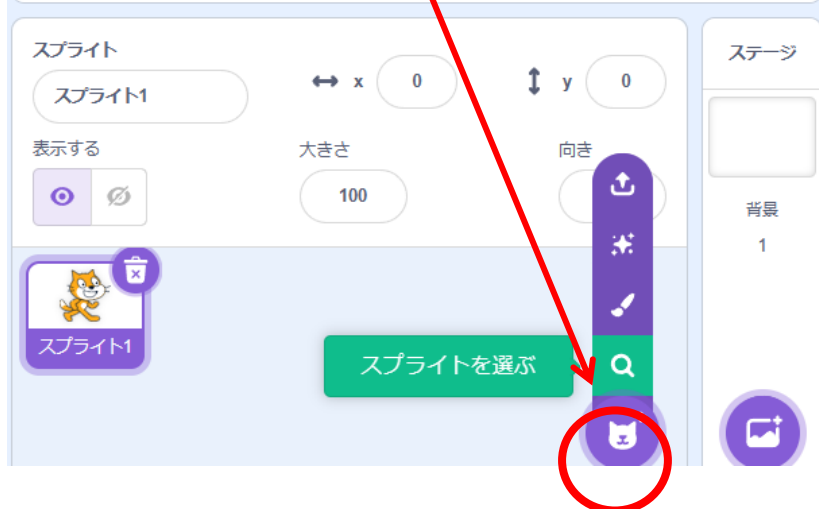
"Scratch3_ftree2c_single.sb3"

13. マルチスプライトの活用: 複数のキャラクター(スプライト)を操る方法(1)

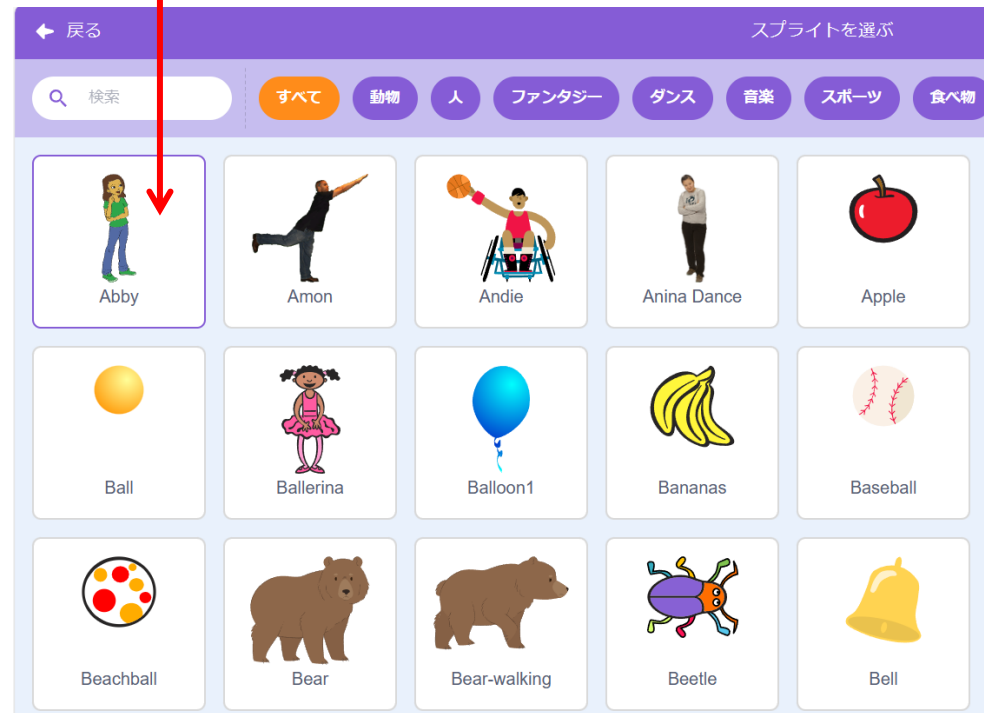
Scratchのマルチスレッド機能を用いることで、複数のキャラクター(スプライト)を操ることができる。はじめに、複数のスプライトを定義する。



スプライトの追加アイコンをクリックし、「スプライトを選ぶ」を選択



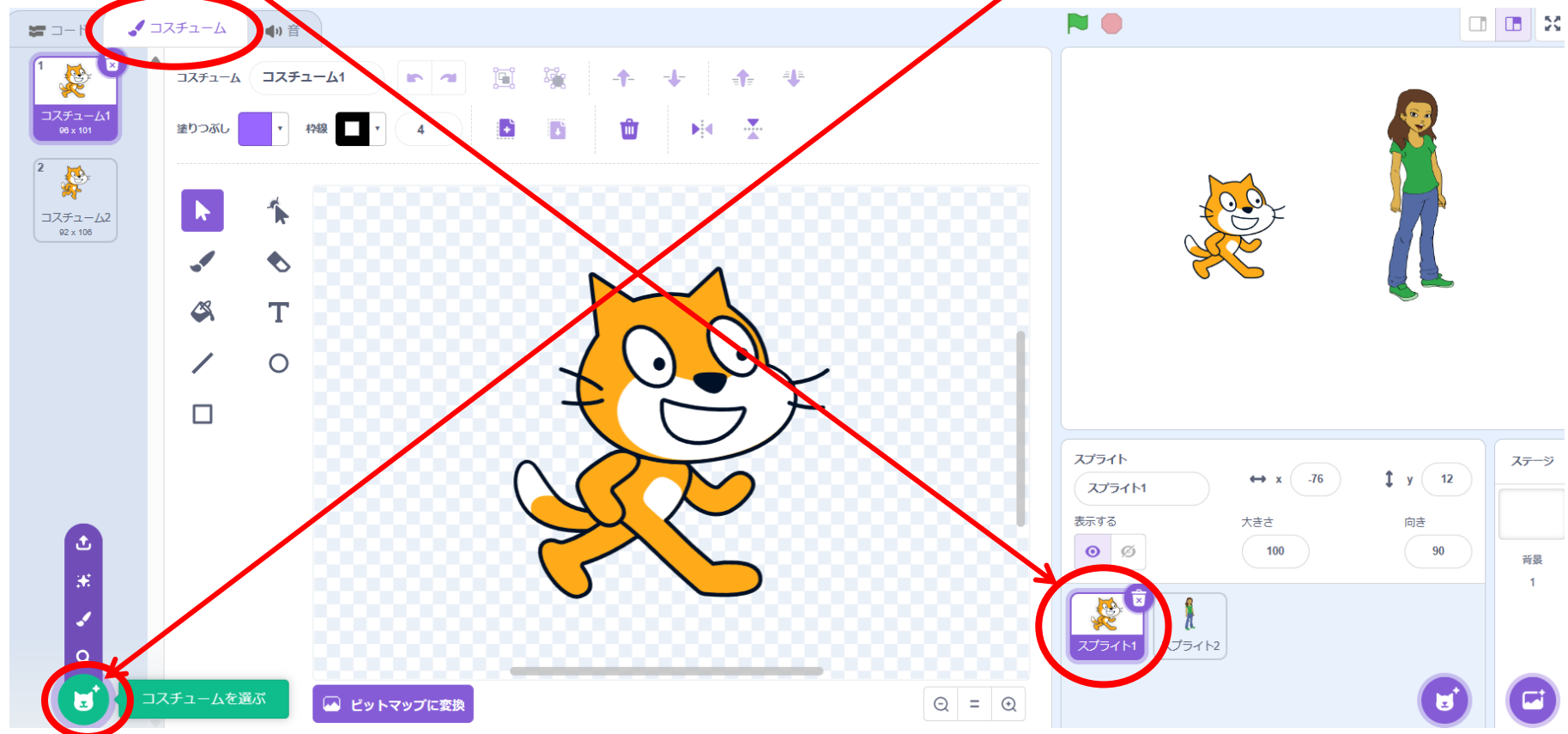
追加するスプライト"Abby"を選択



13. マルチスプライトの活用: 複数のキャラクター(スプライト)を操る方法(2)

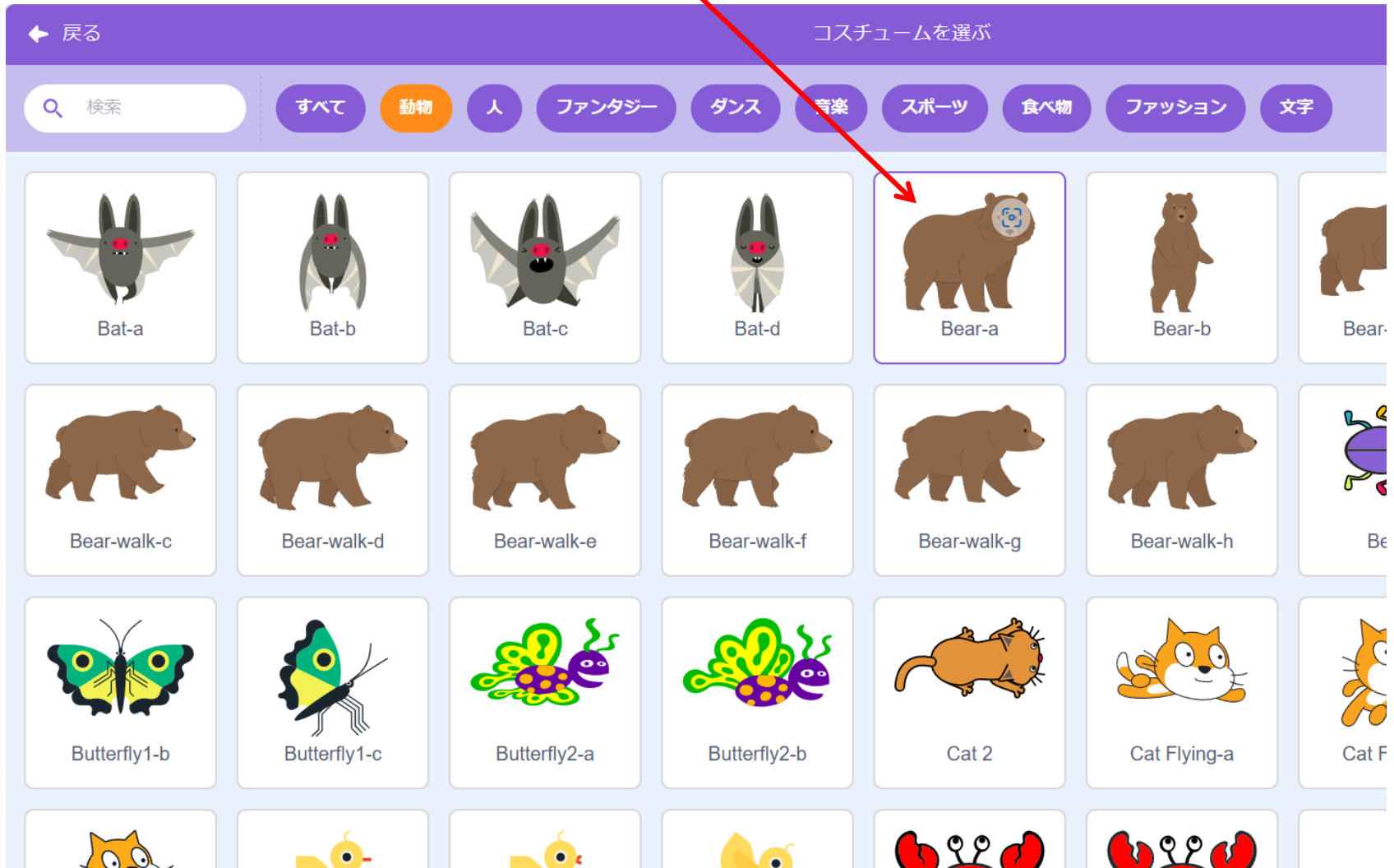
各スプライトに対して、「コスチューム」と「音」を選択する。本例題では「音」の選択は省略し、各スプライトにデフォルト設定されている「音」を使用する。

(1) 「スプライト1」を選び、(2) 「コスチューム」タブを選び、(3) 「コスチュームを選ぶ」ボタンをクリックすると、コスチューム一覧が表示される。



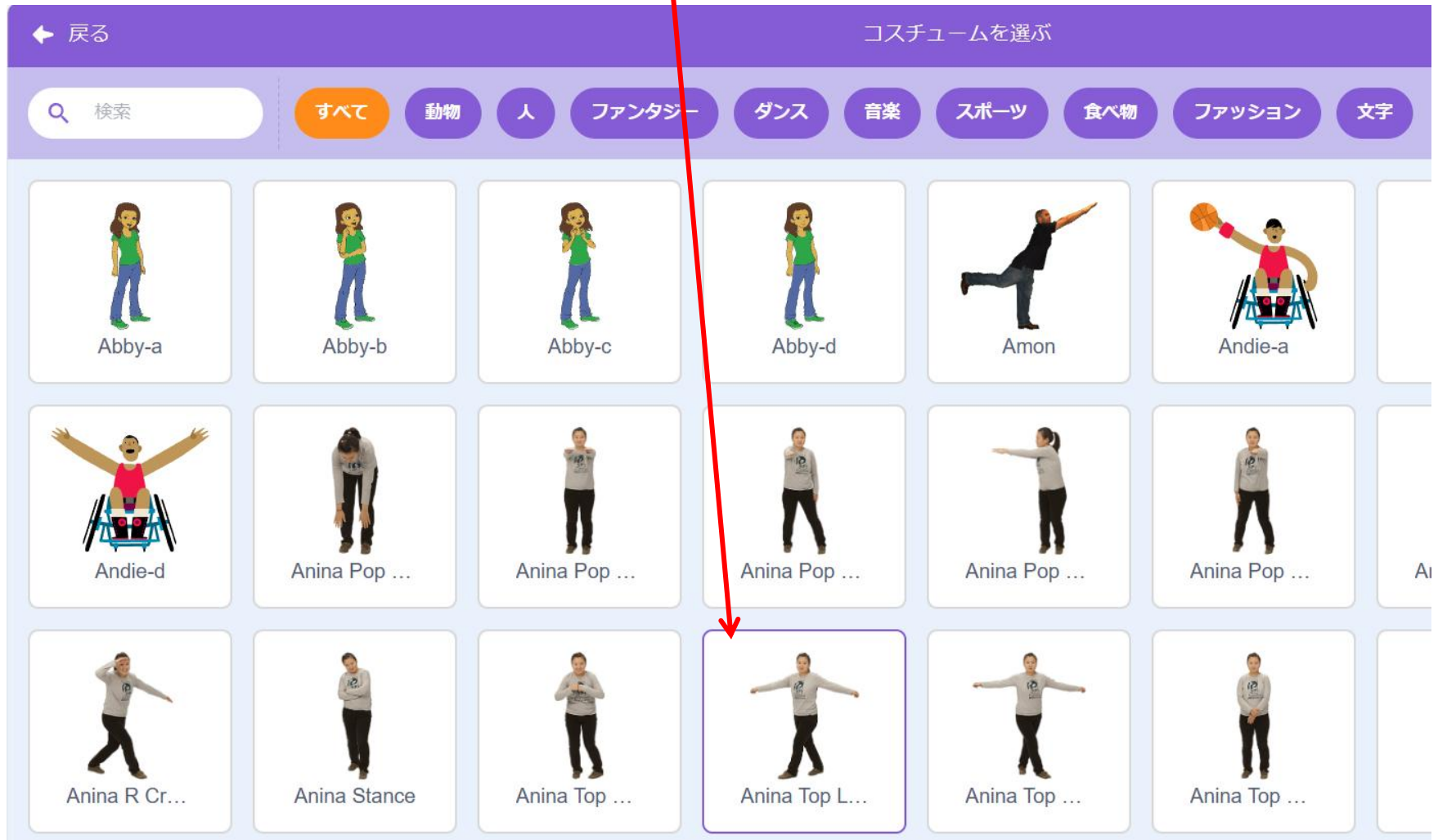
13. マルチスプライトの活用: 複数のキャラクター(スプライト)を操る方法(3)

例えば、スプライト1に対して、「Bear-a」を選択する。



13. マルチスプライトの活用: 複数のキャラクター(スプライト)を操る方法(4)

続いて、スプライト2に対して、「Anina Top L Step」を選択する。



13. マルチスプライトの活用: 複数のキャラクター(スプライト)を操る方法(5)

[スプライト1のプログラム例] "Scratch3_multi_sprite.sb3"



The screenshot shows the Scratch3 Multi-Sprite Editor interface. On the left, the 'Scripts' area contains a program for 'Sprite 1' starting with a 'When green flag clicked' event, followed by 'Set costume to Costume 1', 'Say Sprite 1 for 2 seconds', 'Play Meow sound until finished', 'Send Message 1', and 'Set costume to Bear-a'. The main stage displays two sprites: a cat (Sprite 1) and a girl (Sprite 2). The bottom right shows the 'Sprites' panel with 'Sprite 1' (cat) and 'Sprite 2' (girl) listed. The 'Stage' panel shows 'Background 1'.

が押されたとき

コスチュームを コスチューム1 にする

スプライト1 と 2 秒言う

終わるまで Meow の音を鳴らす

メッセージ1 を送って待つ

コスチュームを Bear-a にする

スプライト

スプライト1

表示する

大きさ 100

向き 90

ステージ

背景 1

スプライト1

スプライト2

スプライト1のプログラムを起動すると、コスチュームを"コスチューム1(Cat1)"にし、「スプライト1」と吹き出し表示し、「Meow」の音を鳴らす。スプライト2にメッセージ1を送り、スプライト2のプログラムを起動する。スプライト2のプログラム終了後に、コスチュームを"Bera-a"に変更する。

13. マルチスプライトの活用: 複数のキャラクター(スプライト)を操る方法(6)

[スプライト2のプログラム例]

"Scratch3_multi_sprite.sb3"



The image shows the Scratch3 interface for a project named "Scratch3_multi_sprite.sb3". On the left, the script editor for Sprite 2 is visible, showing a sequence of five blocks: a yellow "When I receive message 1" block, a purple "Costume to abby-a" block, a purple "Say Sprite 2 for 2 seconds" block, a pink "Play pop sound until finished" block, and a purple "Costume to Anina Top L Step" block. On the right, the stage view shows a large brown bear sprite (Sprite 1) and a person in a grey sweatshirt and black pants (Sprite 2) standing next to it. Below the stage, the Sprite and Stage panels are visible. The Sprite panel shows two sprites: a bear (Sprite 1) and a person (Sprite 2). The Stage panel shows a single background (Background 1).

メッセージ1 を受け取ったとき

コスチュームを abby-a にする

スプライト2 と 2 秒言う

終わるまで pop の音を鳴らす

コスチュームを Anina Top L Step にする

スプライト1よりメッセージ1を受け取ると、スプライト2のプログラムが起動される。コスチュームを"abby-a"にし、「スプライト2」と吹き出し表示し、「pop」の音を鳴らす。

その後、コスチュームを"Anina Top L Step"に変更する。

13. マルチスプライトの活用例(1): 複数の波形描画(1)

第9章で紹介した「波形合成・うなり」"Scratch3_beetw.sb3"において、3本の波形を描画する処理につき、3種のスプライト(スプライト1(Cat1), Dog1, ButterFly1)を定義して並列処理で描画できるように書き直したものが、次ページに示す"Scratch3_beetw_3sprite.sb3"である。

本プログラムで使用する変数として、幅、振幅、X、Y、X2の5点を定義しているが、「波形表示」、「波形設定」、「枠表示」ブロック内で使用する変数Xと変数Yおよび、「波形表示」ブロック内で使用する変数X2は、マルチスレッドで動いている場合、各スレッド(スプライト)ごとに異なる値になるため、変数を定義する際、ローカル変数(このスプライトのみ)に変更設定する必要がある。他の3つの変数はプログラム内で共通して使用するため、デフォルトのグローバル変数(すべてのスプライト用)にする。

13. マルチスプライトの活用例(1):複数の波形描画(2):スプライト1コード

The image shows a Scratch3 project titled "Scratch3_beetw_3sprite.sb3". The project features a stage with a black background and a white coordinate system. Three waveforms are plotted: a green sine wave, a blue sine wave, and a red sine wave. The waves are drawn by three different sprites: "Sprite 1" (a cat), "Dog1" (a dog), and "Butterfly 1" (a butterfly). The code editor on the left shows the logic for these sprites.

Code Editor:

- Sprite 1 (Cat):**
 - When green flag clicked, set all to 0.
 - Set size to 50%.
 - Set width to 200.
 - Set amplitude to 140.
 - Pen up.
 - Set x coordinate to X, y coordinate to Y.
 - Pen down.
- Sprite 2 (Dog1):**
 - When green flag clicked, set all to 0.
 - Set size to 50%.
 - Set width to 200.
 - Set amplitude to 140.
 - Pen up.
 - Set x coordinate to X, y coordinate to Y.
 - Pen down.
- Sprite 3 (Butterfly 1):**
 - When green flag clicked, set all to 0.
 - Set size to 50%.
 - Set width to 200.
 - Set amplitude to 140.
 - Pen up.
 - Set x coordinate to X, y coordinate to Y.
 - Pen down.

Stage:

- Width: 200, Height: 45.
- Size: 50, Direction: 90.
- Background: 1.

Wave Drawing Logic:

- Define Wave Setting:** Set pen width to 1, set x to 0 - width, set y to 0.
- Define Wave Display:** Set pen width to 2, set pen color to black, set x to 0 - width, set y to 0.
- Define Line Drawing:** Set x1 to 0, y1 to 0, set x2 to 0 - width, y2 to 0.

Wave Drawing Process:

- Set x to width, y to amplitude * sin(x / width * 2 * pi).
- Set x to x + width, y to y + amplitude * sin(x / width * 2 * pi).
- Set x to x + 1.

"Scratch3_beetw_3sprite.sb3"

13. マルチスプライトの活用例(1):複数の波形描画(3):Dog1コード

The image displays a Scratch3 project titled "Dog1コード" (Dog1 Code). The left panel shows the code blocks, and the right panel shows the stage view.

Code Blocks:

- Message:** "メッセージ1" (Message 1) を受け取ったとき (When I receive Message 1).
- Pen Color:** ペンの色を (Pen color) にする (Set to).
- Waveform Display:** 波形表示 (Waveform display) 10 0.
- Definition:** 定義 (Definition) 波形設定 (Waveform settings).
- Pen Thickness:** ペンの太さを (Pen thickness) 1 にする (Set to).
- Coordinates:** X を (X) 0 - 幅 (width) にする (Set to).
- Pen Up:** ペンを上げる (Lift pen).
- Coordinates:** x座標を (x coordinate) X 、y座標を (y coordinate) Y にする (Set to).
- Pen Down:** ペンを下ろす (Put pen down).
- Waveform Settings:** 定義 (Definition) 波形表示 (Waveform display) 倍率 (Scale) 倍率2 (Scale 2).
- Waveform Settings:** 波形設定 (Waveform settings) X > 幅 (width) まで繰り返す (Repeat until).
- Waveform Settings:** X2 を (X2) X + 幅 (width) * 倍率 (Scale) にする (Set to).
- Waveform Settings:** Y を (Y) 振幅 (Amplitude) * X2 の sin (sin) にする (Set to).
- Waveform Settings:** もし (If) 倍率2 (Scale 2) > 0 なら (then) X2 を (X2) X + 幅 (width) * 倍率2 (Scale 2) にする (Set to).
- Waveform Settings:** Y を (Y) Y + 振幅 (Amplitude) * X2 の sin (sin) にする (Set to).
- Coordinates:** x座標を (x coordinate) X 、y座標を (y coordinate) Y にする (Set to).
- Coordinates:** X を (X) X + 1 にする (Set to).

Stage View:

- Sprite 1:** スプライト1: X 201, スプライト1: Y 44.995133, スプライト1: X2 3600.
- Waveform:** 幅 (width) 200, 振幅 (Amplitude) 70.
- Sprite:** スプライト (Sprite) Dog1, 表示する (Display) 30, 向き (Direction) 90.
- Stage:** ステージ (Stage) 背景 (Background) 1.

"Scratch3_beetw_3sprite.sb3"

13. マルチスプライトの活用例(1):複数の波形描画(4):ButterFly1コード

The image displays the Scratch3 code editor with a script for drawing multiple sine waves. The script is as follows:

```
メッセージ1 を受け取ったとき
  ペンの色を 青 にする
  波形表示 9 0
  定義 波形設定
  ペンの太さを 1 にする
  X を 0 - 幅 にする
  Y を 0 にする
  ペンを上げる
  x座標を X 、y座標を Y にする
  ペンを下ろす
  定義 波形表示 倍率 倍率2
  波形設定
  X > 幅 まで繰り返す
    X2 を X + 幅 * 倍率 にする
    Y を 振幅 * X2 の sin にする
  もし 倍率2 > 0 なら
    X2 を X + 幅 * 倍率2 にする
    Y を Y + 振幅 * X2 の sin にする
  x座標を X 、y座標を Y にする
  X を X + 1 にする
```

The right side of the image shows the stage with a coordinate system. The butterfly sprite is positioned at the end of the drawn waves. The stage settings are as follows:

スプライト	スプライト1: X	スプライト1: Y	スプライト1: X2
Butterfly 1	201	44.995133	3600

Below the stage, the sprite settings for Butterfly 1 are shown:

スプライト	表示する	大きさ	向き
Butterfly 1	☒ 表示 ☐ 非表示	30	90

The stage background is set to 1.

"Scratch3_beetw_3sprite.sb3"

13. マルチスプライトの活用例(1):複数の波形描画(5):コードのコピー方法

「スプライト1」のコードによりメッセージ1を送信すると、「Dog1」および「ButterFly1」のスプライトの「波形表示」ブロックが起動され、「スプライト1」を含む3種の「波形表示」ブロックが並列実行される。

即ち、「スプライト1」のコードの「波形表示」、「波形設定」ブロックは、「Dog1」および「ButterFly1」のスプライトでも使用するため、これらのコードを「Dog1」および「ButterFly1」のスプライトに次ページのようにDrag&Dropによりコピーする。

ローカル変数X、Y、X2は、本来各スプライトごとに定義する必要があるが、コードのコピー操作により「スプライト1」のローカル変数も一緒にコピーされ、「Dog1」および「ButterFly1」のスプライトでもローカル変数として使用できる。

13. マルチスプライトの活用例(1):複数の波形描画(6):コードのコピー方法(2)

13. マルチスプライトの活用例(1):複数の波形描画(7):クローンによる実装

本プログラムは前述の複数のクローンを生成する方法でも実装できる。自分自身のクローン(スプライト)を3種生成して、自分自身をクローンIDを2として、クローンID:0~2の3種のクローンの各々に以下の設定で定義ブロック「波形表示」を並列実行させる。

クローン0: 周期倍率9倍の単一のサインカーブ

クローン1: 周期倍率10倍の単一のサインカーブ

クローン2: 上記2種のサインカーブの合成波形

これらの設定を行ったのが、次ページに示す "Scratch3_beetw_3clone.sb3"である。

13. マルチスプライトの活用例(1):複数の波形描画(8):クローンによる実装(2)

The image displays a Scratch3 project titled "Scratch3_beetw_3clone.sb3". The script is designed to draw multiple overlapping waveforms using clones.

Script Details:

- When Green Flag Clicked:**
 - 全部消す (Erase All)
 - 大きさを 50 %にする (Set size to 50%)
 - 幅 を 200 にする (Set width to 200)
 - 振幅 を 140 にする (Set amplitude to 140)
 - 枠表示 (Show bounding box)
 - 振幅 を 70 にする (Set amplitude to 70)
 - ID を 0 にする (Set ID to 0)
 - 自分自身 のクローンを作る (Create clone of self)
 - ID を 1 にする (Set ID to 1)
 - 自分自身 のクローンを作る (Create clone of self)
 - ペンの色を 緑 にする (Set pen color to green)
 - 波形表示 10 9 (Show waveform 10 9)
- クローンされたとき (When cloned):**
 - もし ID = 0 なら (If ID is 0):
 - ペンの色を 紫 にする (Set pen color to purple)
 - 波形表示 10 0 (Show waveform 10 0)
 - でなければ (Otherwise):
 - ペンの色を 青 にする (Set pen color to blue)
 - 波形表示 9 0 (Show waveform 9 0)
 - このクローンを削除する (Delete this clone)

Visual Output:

The stage shows a coordinate system with three overlapping waveforms: a green one (ID 1), a purple one (ID 0), and a blue one (ID 0). The waveforms are drawn with a width of 200 and an amplitude of 70. The Scratch cat is visible on the right side of the stage.

Stage Properties:

- スプライト: スプライト1
- 表示する: 表示 (Show)
- 大きさ: 50
- 向き: 90
- 背景: 1

"Scratch3_beetw_3clone.sb3"

13. マルチスプライトの活用例(2):ピアノ鍵盤(1):三和音の打鍵

第10章で紹介した「ピアノ鍵盤」"Scratch3_piano.sb3"において、C3～B3のいずれかのキーを打鍵すると、そのキーをルートとして、長三和音または短三和音を構成する2種の音を自動的に打鍵して和音演奏できるようにする処理につき、3種のスプライト(スプライト1(Cat1), Dog2, ButterFly1)を定義して並列処理で演奏できるように書き直したものが、次ページに示す"Scratch3_piano_3sprite.sb3"である。

"Scratch3_piano.sb3"においては鍵盤サイズが1オクターブ(13鍵)であったが、三和音を打鍵できるように、21鍵に拡張した。

本プログラムで使用する変数として、WW, WH, BW, BH, NN, WW2, C, mode, X, Y, X2, Y2, 移調の13点を定義しているが、「演奏」、「座標計算」ブロック内で使用する変数Cは、マルチスレッドで動いている場合、各スレッド(スプライト)ごとに異なる値になるため、変数を定義する際、ローカル変数(このスプライトのみ)に変更設定する必要がある。他の12個の変数はプログラム内で共通して使用するため、デフォルトのグローバル変数(すべてのスプライト用)にする。

プログラムを起動すると、演奏モードをきいてくるので、0～2のいずれかを入力する。

(1:打鍵音を単音演奏、1:打鍵音をルートとする長三和音演奏、2:打鍵音をルートとする短三和音演奏)

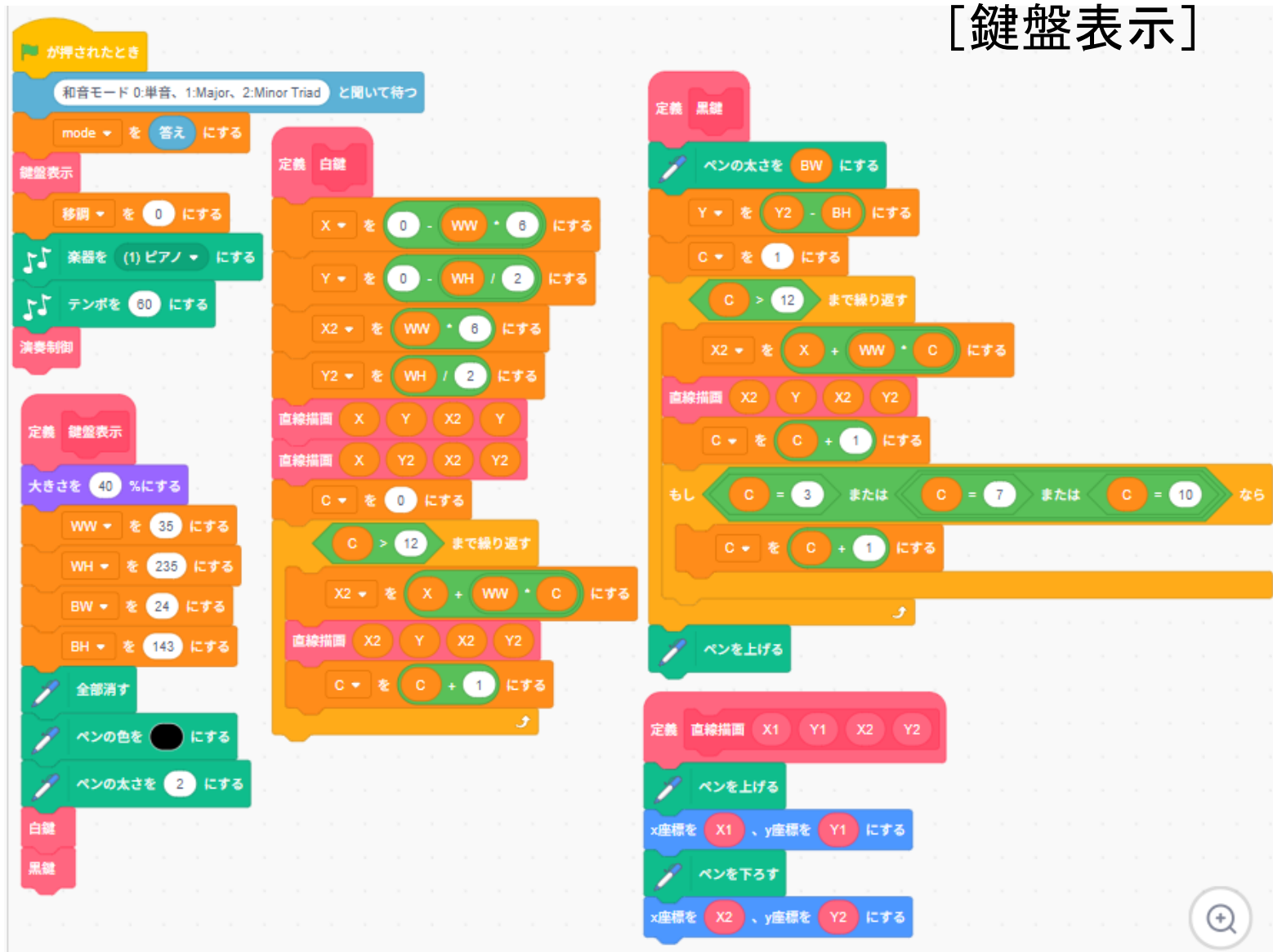
mode=1の場合、C3～B3のいずれかのキーを打鍵すると、メッセージ1が送信される。スプライト1では打鍵した音を認識して、打鍵位置を表示しそのまま演奏する。Dog2では、打鍵した音より4半音高い音を表示しその音を演奏する。ButterFly1では、打鍵した音より7半音高い音を表示しその音を演奏する。

13. マルチスプライトの活用例(2):ピアノ鍵盤(2):三和音の打鍵(2)

[illegible]

13. マルチスプライトの活用例(2):ピアノ鍵盤(3):スプライト1のコード(1)

[鍵盤表示]



The image displays a Scratch script for a piano keyboard visualization, organized into several functional sections:

- Initialization (when clicked):** Sets the harmony mode to 0 (Major/Minor Triad), initializes the mode variable, sets the instrument to piano and tempo to 60, and defines the keyboard display function.
- Keyboard Display Function:** A custom block that sets drawing parameters (size 40%, WW=35, WH=235, BW=24, BH=143), clears the stage, sets pen color to black and thickness to 2, and then calls the white and black key drawing functions.
- White Key Drawing Function:** A custom block that calculates the position and size of white keys based on the width (WW) and height (WH) of the keyboard, and uses the 'draw line' block to create the key shape.
- Black Key Drawing Function:** A custom block that calculates the position and size of black keys, using the 'draw line' block to create the key shape.

The script uses various Scratch blocks including 'when clicked', 'set harmony mode to', 'set mode to', 'set instrument to', 'set tempo to', 'define keyboard display', 'set size to', 'set WW to', 'set WH to', 'set BW to', 'set BH to', 'clear stage', 'set pen color to', 'set pen thickness to', 'define white key', 'define black key', and 'draw line'.

"Scratch3_piano_3sprite.sb3"

13. マルチスプライトの活用例(2):ピアノ鍵盤(4):スプライト1のコード(2)

[打鍵音認識]

The image displays three Scratch code snippets for piano key recognition, each within a custom block definition.

定義 演奏制御

- スペース キーが押された まで繰り返す
- もし マウスが押された なら
 - x座標を マウスのx座標、y座標を マウスのy座標 にする
 - X2 を マウスのx座標 - X にする
 - もし マウスのy座標 > Y なら
 - 黒鍵算出
 - でなければ
 - 白鍵算出
 - NN を C + 移調 にする
 - メッセージ1 を送る
 - マウスが押された ではない まで待つ

定義 白鍵算出

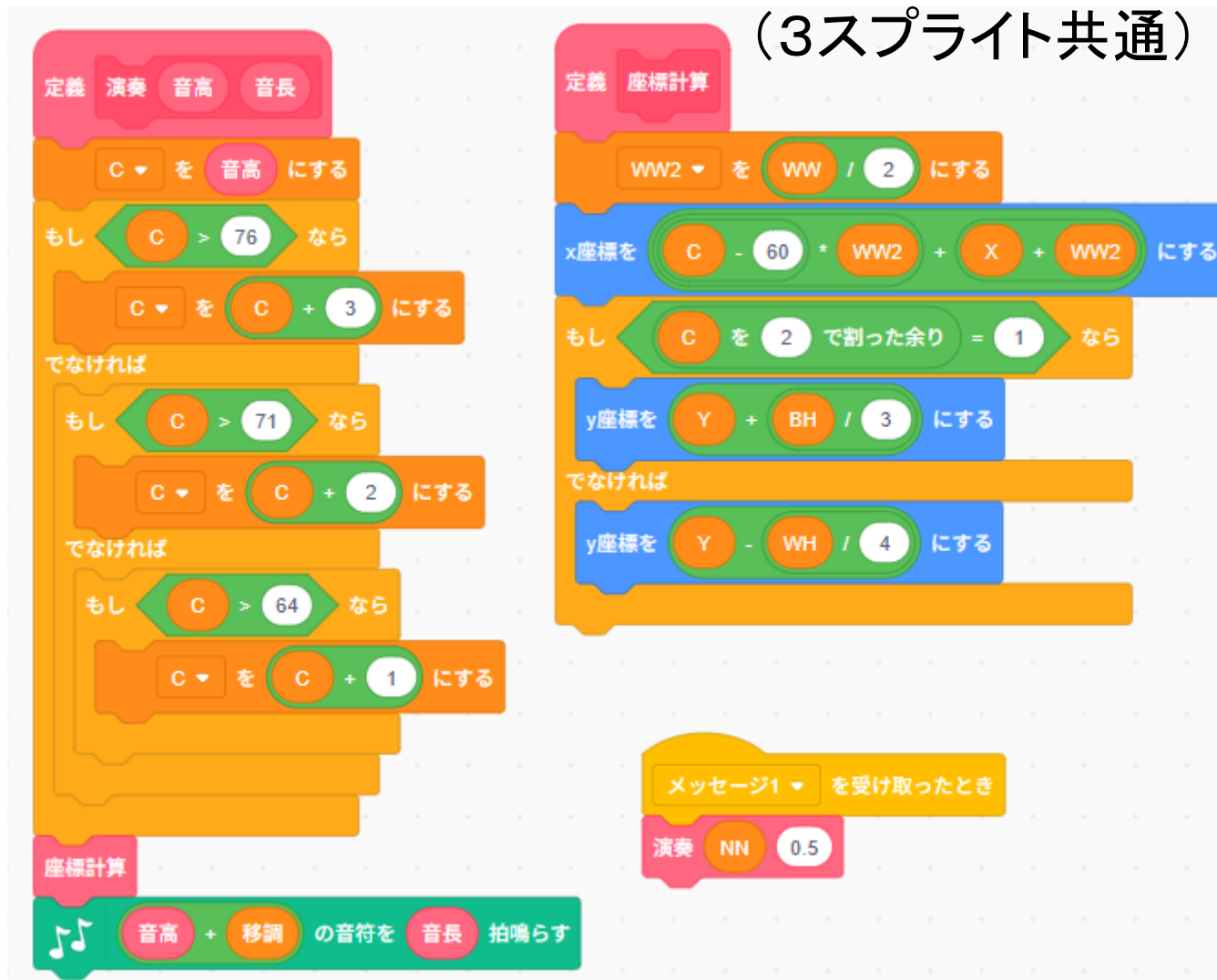
- C を $X2 / WW$ にする
- C を C の 切り下げ にする
- もし $C < 3$ なら
 - C を $60 + C \cdot 2$ にする
- でなければ
- もし $C < 7$ なら
 - C を $59 + C \cdot 2$ にする
- でなければ
- もし $C < 10$ なら
 - C を $58 + C \cdot 2$ にする
- でなければ
- C を $57 + C \cdot 2$ にする

定義 黒鍵算出

- C を $(X2 - WW / 2) / WW$ にする
- C を C の 切り下げ にする
- もし $C < 3$ なら
 - C を $61 + C \cdot 2$ にする
- でなければ
- もし $C < 7$ なら
 - C を $60 + C \cdot 2$ にする
- でなければ
- もし $C < 10$ なら
 - C を $59 + C \cdot 2$ にする
- でなければ
- C を $58 + C \cdot 2$ にする

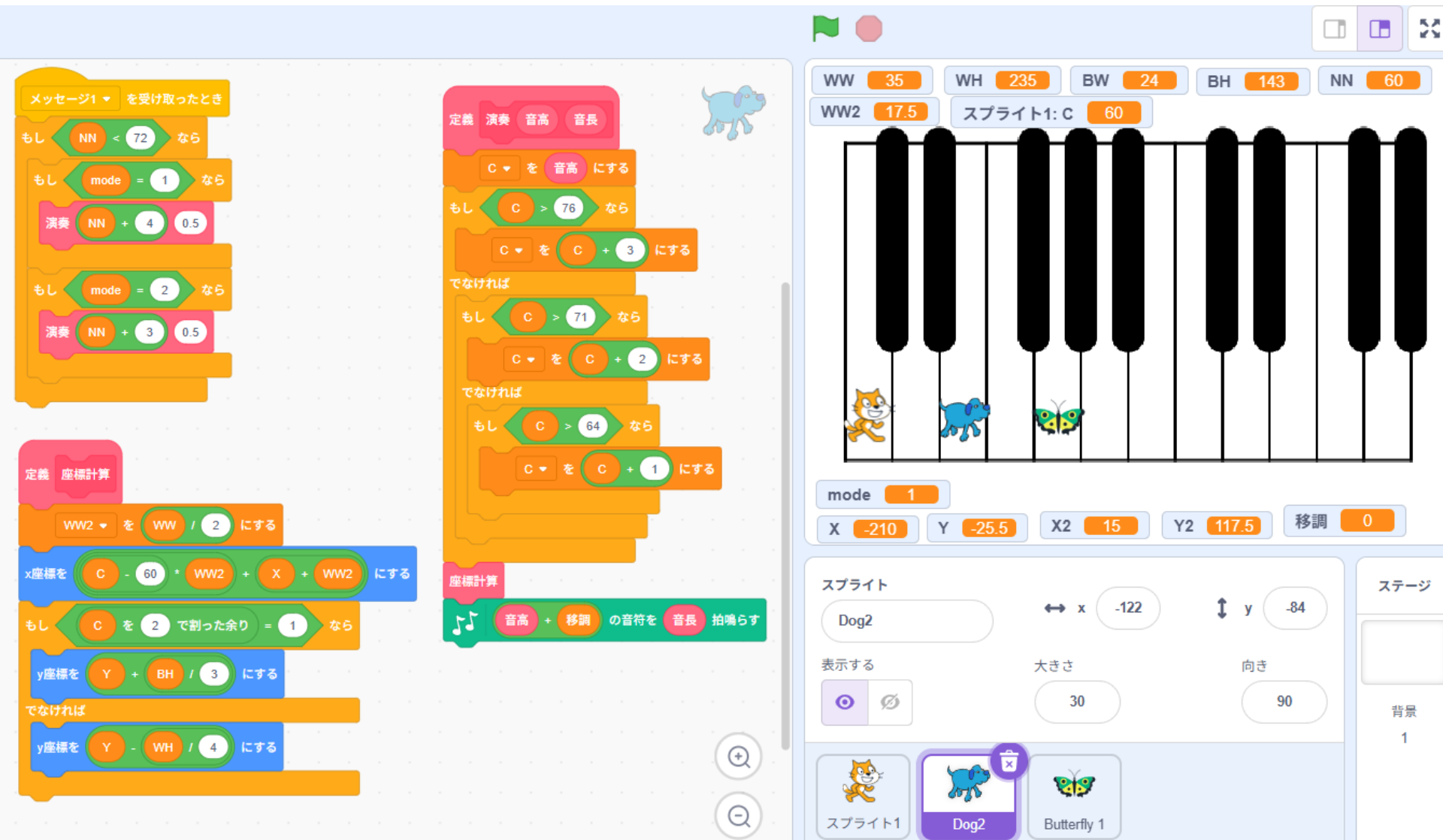
13. マルチスプライトの活用例(2):ピアノ鍵盤(5):スプライト1のコード(3)

[打鍵音の位置表示と演奏]
(3スプライト共通)



"Scratch3_piano_3sprite.sb3"

13. マルチスプライトの活用例(2):ピアノ鍵盤(6):Dog2のコード



The image displays the Scratch3 piano_3sprite.sb3 project, showing the code for the Dog2 sprite and the stage layout.

Code Snippets:

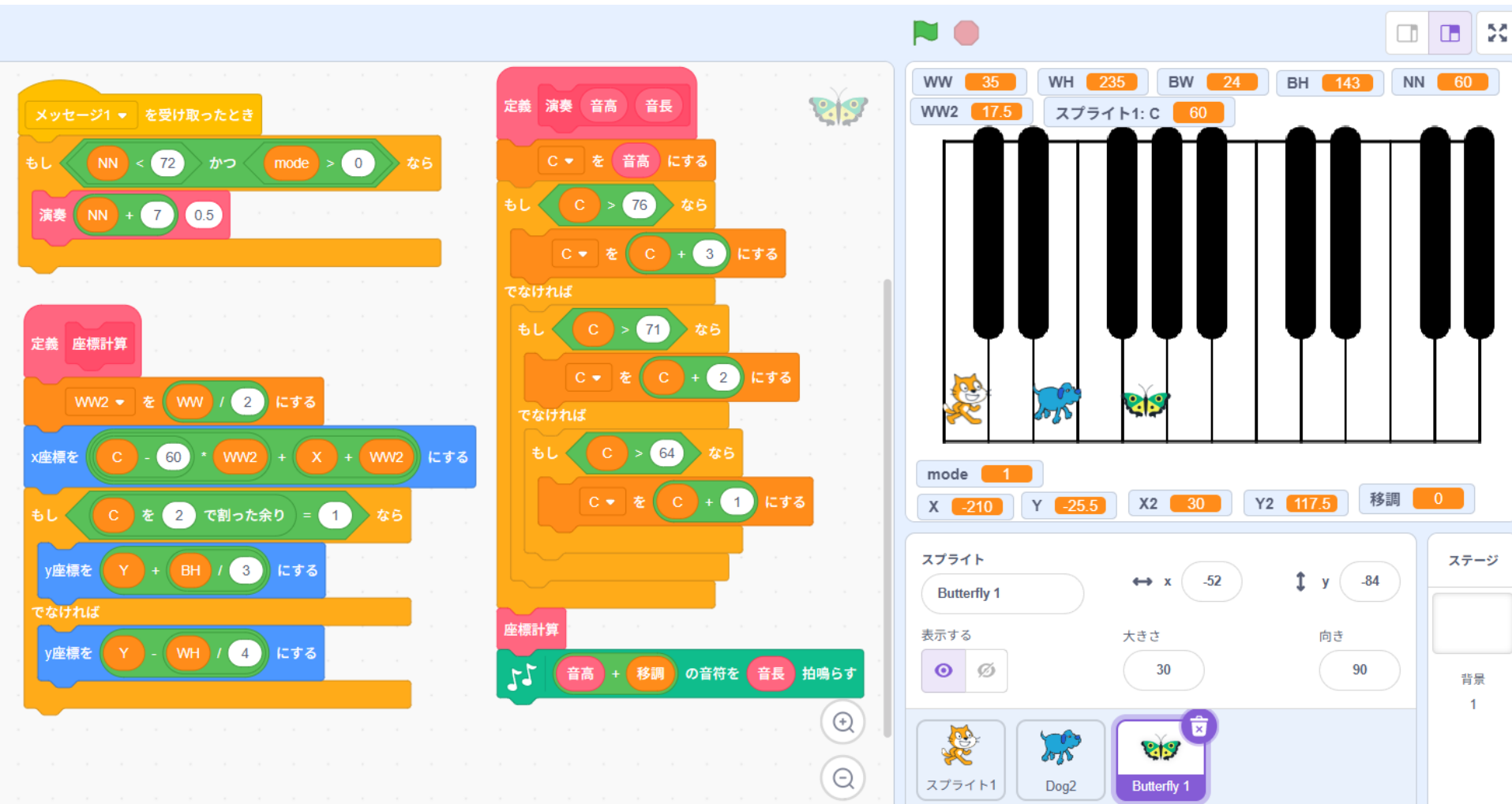
- Message 1: When received**
 - もし $NN < 72$ なら
 - もし $mode = 1$ なら
 - 演奏 $NN + 4$ 0.5
 - もし $mode = 2$ なら
 - 演奏 $NN + 3$ 0.5
- Define: Coordinate Calculation**
 - $WW2$ を $WW / 2$ にする
 - x座標を $C - 60 * WW2 + X + WW2$ にする
 - もし C を 2 で割った余り = 1 なら
 - y座標を $Y + BH / 3$ にする
 - でなければ
 - y座標を $Y - WH / 4$ にする
- Define: Performance, Pitch, Duration**
 - C を 音高 にする
 - もし $C > 76$ なら
 - C を $C + 3$ にする
 - でなければ
 - もし $C > 71$ なら
 - C を $C + 2$ にする
 - でなければ
 - もし $C > 64$ なら
 - C を $C + 1$ にする
 - 座標計算
 - 音高 + 移調 の音符を 音長 拍鳴らす

Stage View:

- WW** 35, **WH** 235, **BW** 24, **BH** 143, **NN** 60
- WW2** 17.5, **スプライト1: C** 60
- mode** 1
- X** -210, **Y** -25.5, **X2** 15, **Y2** 117.5, **移調** 0
- スプライト**
 - Dog2: x -122, y -84, 表示する (checked), 向き 90
- ステージ**
 - 背景 1

"Scratch3_piano_3sprite.sb3"

13. マルチスプライトの活用例(2):ピアノ鍵盤(7):ButterFly1のコード



The image displays the Scratch3 piano_3sprite.sb3 project, showing the code for the Butterfly 1 sprite and the stage view.

Code for Butterfly 1:

- Message 1: When received**
 - IF $NN < 72$ AND $mode > 0$ THEN
 - Play $NN + 7$ for 0.5 seconds
- Define: Calculate coordinates**
 - Set $WW2$ to $WW / 2$
 - Set x coordinate to $C - 60 * WW2 + X + WW2$
 - IF C divided by 2 has a remainder of 1 THEN
 - Set y coordinate to $Y + BH / 3$
 - ELSE
 - Set y coordinate to $Y - WH / 4$
- Define: Play note**
 - Set C to $C + 3$ (Pitch)
 - IF $C > 76$ THEN
 - Set C to $C + 3$
 - ELSE IF $C > 71$ THEN
 - Set C to $C + 2$
 - ELSE IF $C > 64$ THEN
 - Set C to $C + 1$
 - Calculate coordinates
 - Play note with pitch C and duration NN (Pitch and Duration)

Stage View:

- Variables:** WW 35, WH 235, BW 24, BH 143, NN 60, WW2 17.5, スプライト1: C 60
- Costume:** Butterfly 1
- Position:** X -210, Y -25.5, X2 30, Y2 117.5, 移調 0
- Mode:** 1
- Sprite List:** スプライト1, Dog2, Butterfly 1

"Scratch3_piano_3sprite.sb3"