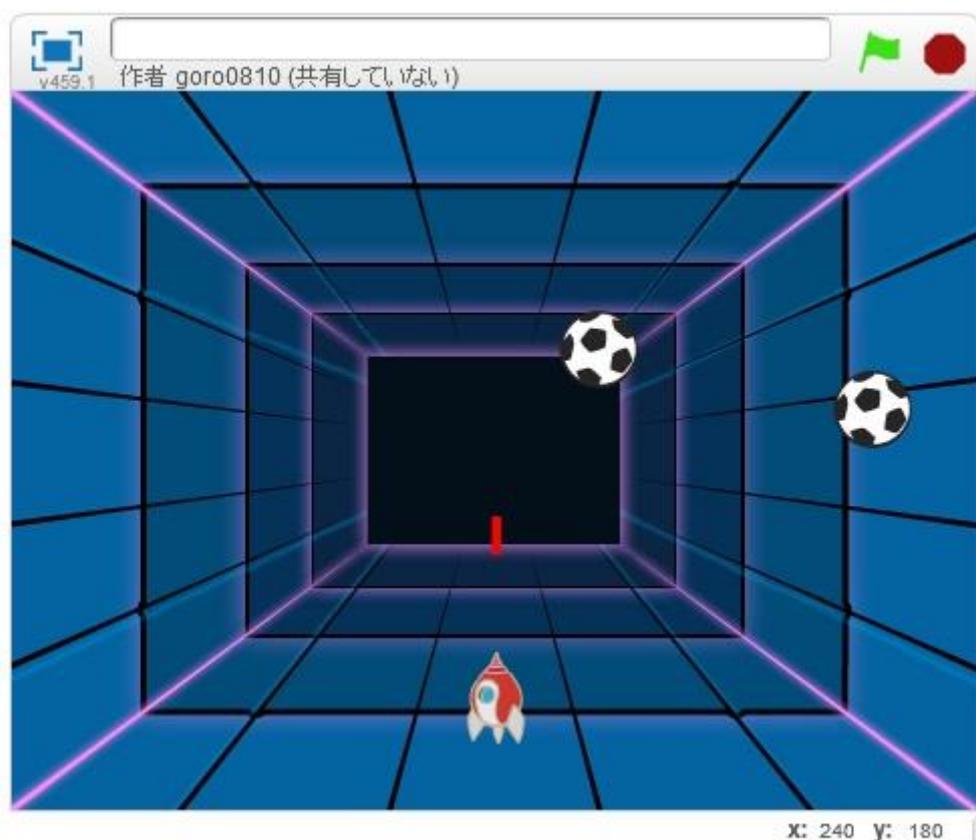


スクラッチを使って シューティングゲームを つくろう



<http://www.mathtext.info/>



数楽

プログラミングとは

プログラミングとは、決まったことをコンピュータやロボットにしてもらうために書く、マニュアル(作業手順書)のことです。

身近なもので例をあげると、自動販売機がそうです。お金を入れて、ボタンを押すと希望の飲み物が出てくる。おつりが必要であれば、適切な金額のおつりが出てきます。そのようにプログラムされていなければ、だれかほかの人が代わりにそれをしなければなりません。

コンピュータやロボットにこんなことしてほしいなと思ったら、そのようにしてもらうように、マニュアルをつくっていきます。そうすることで、コンピュータやロボットはそのマニュアル通り動くことになります。このようなマニュアルをつくることをプログラミングといいます。

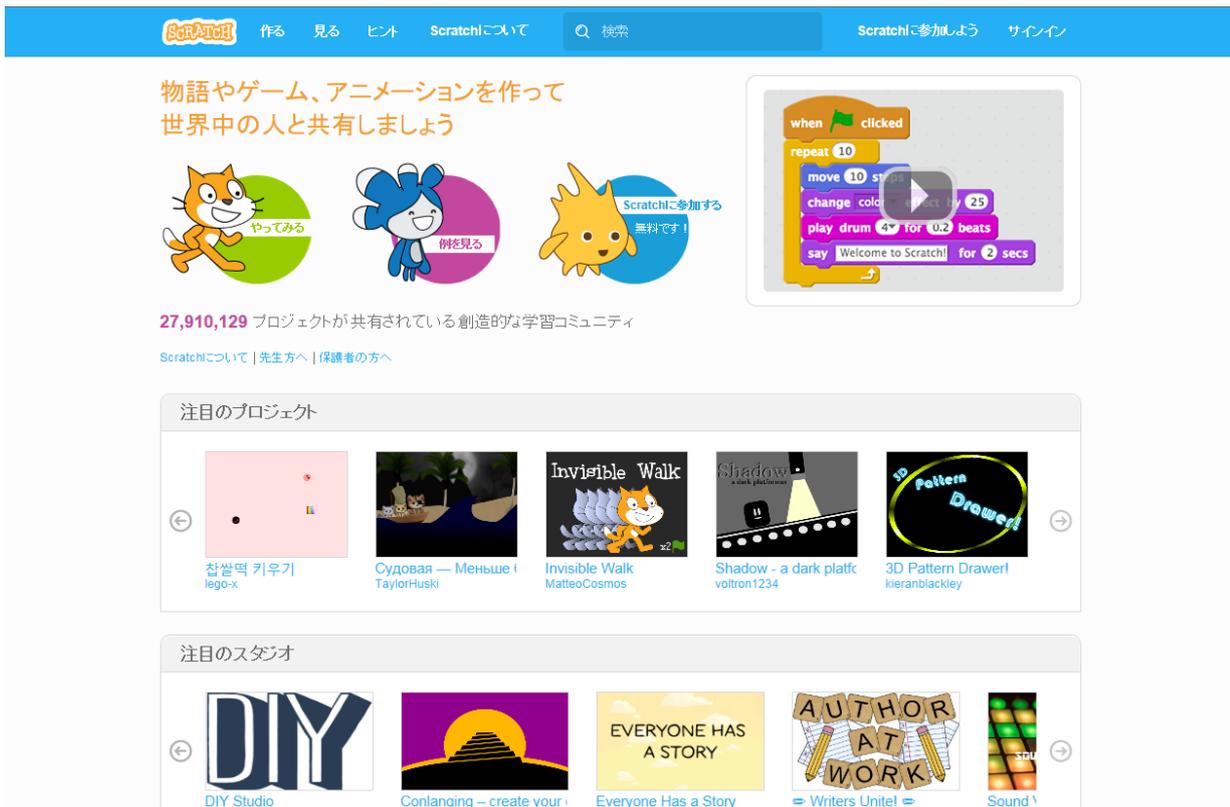
プログラミングができれば、人の役にものすごく役に立ちます。切符の券売機、インターネットオークション、スマホアプリ、Googleなどの検索画面など身近なものがほとんど、プログラミングで、できています。

プログラムが動く仕組み

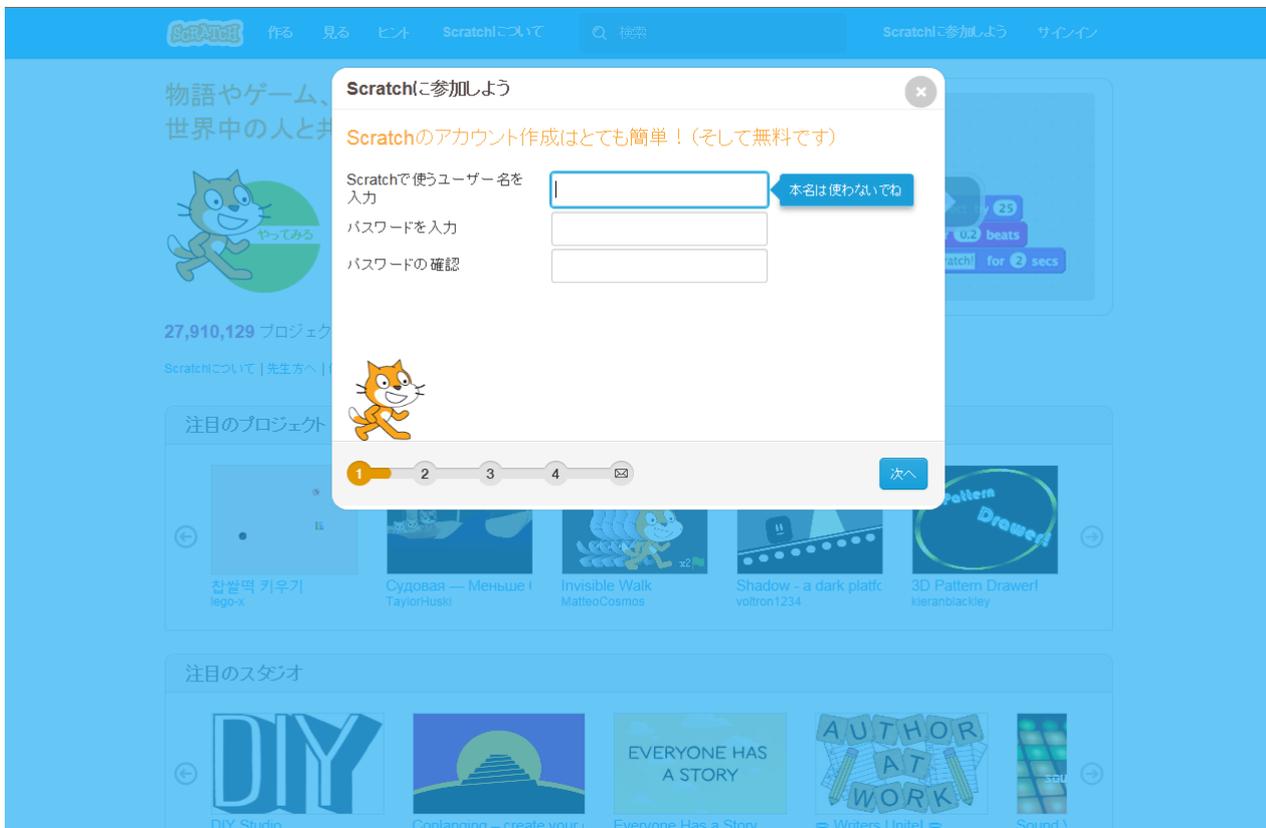
コンピュータなどの電子機械は信号があるかないかで動きます。専門的には2進数というもので動くことになります。どのように動くのか決めたのがマニュアルですが、コンピュータにはそのマニュアルは読めません。ですから、つくったマニュアルをコンピュータが読めるように電気信号に変換します。それがソフトウェアというものです。マニュアルをソフトウェアで電気信号に変えることで、コンピュータの頭脳であるCPUを動かして、期待する動作を得ることができます。

*プログラムが正確に書かれていないと、コンピュータは正常に動きません。場合によっては、ピクリともしません。そのあたりをわかっておかないと、プログラミングが楽しくなくなります。今回取り上げるスクラッチには、たくさんのサンプルプログラミングの本が出ておりますから、やってみたいと思った人は、本屋さんでご購入ください。

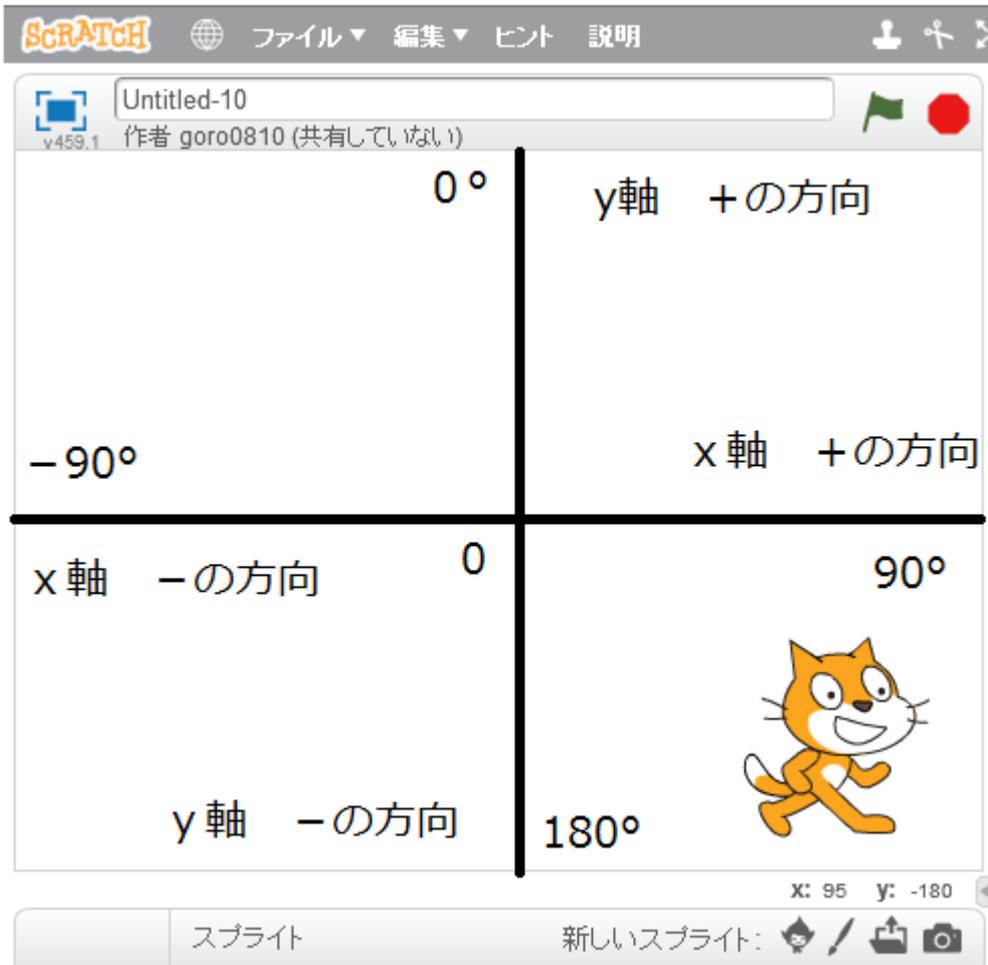
スクラッチを始めるには、スクラッチ MIT と検索をかけて、以下の画面に行きます。



サインインから、初期設定を行います。ただし、保護者の方の許可は得ましょう。各家にはそれぞれ家のルールがありますからね。



スクラッチでプログラミングしていくときは、画面が図のように設定されていますので、注意してつくっていきましょう。

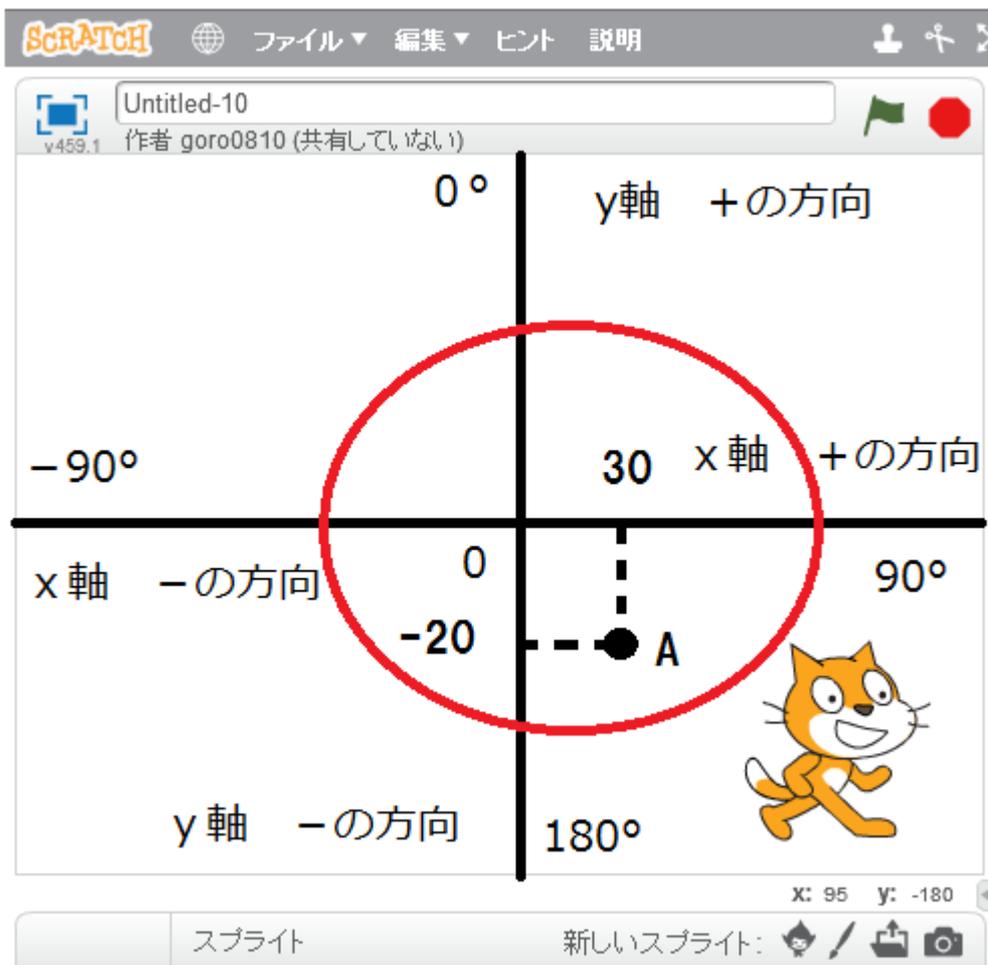


画面の中心が0というところです。0から横に伸びた線をx軸、0から縦に伸びた線をy軸といいます。

0はxもyも0のところなので、(x、y)の順番に(0、0)というように書きます。これを座標といいます。

スクラッチの画面は
横方向(x軸)は-240~+240の範囲
縦方向(y軸)は-180~+180の範囲
で構成されています。

例えば下の図で、Aはxが30、yが-20のところにありますから、Aの位置を(30、-20)という座標で表します。キャラクターをAに表示させれば、xが30、yが-20のところにキャラクターを表示するプログラムを書けば、表示することができます。キャラクターなどをコントロールするために、座標は必要になります。



キャラクターや背景を設定する場合に使うボタンを○で囲っておきます。



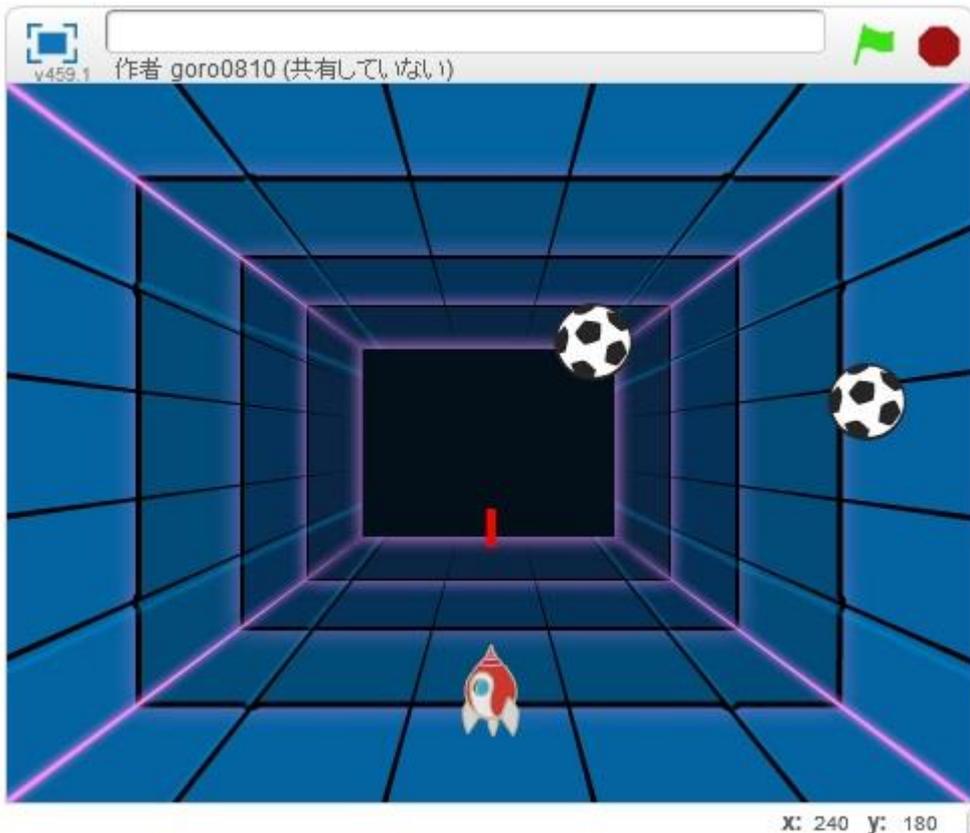
 このボタンは自分で作ったものをアップロードするときに使います。

 このボタンは自分でキャラクターや背景をつくっていくときに使います。

まずゲームをつくるのには設計図が必要です。ちょうど家を建てるのに設計図が必要なのに同じように。

ですから、今回は自機(自分が動かす機械)と敵キャラ(サッカーボール)があって、サッカーボールをミサイルで撃ち落とすというものをつくってみましょう。

途中で設計図(ゲームの内容)を変えてもいいですし、途中で失敗してもいいです。とにかく上達するには失敗をくり返し、その分、上達していくことです。失敗してすることが、プログラミングでは許されています。

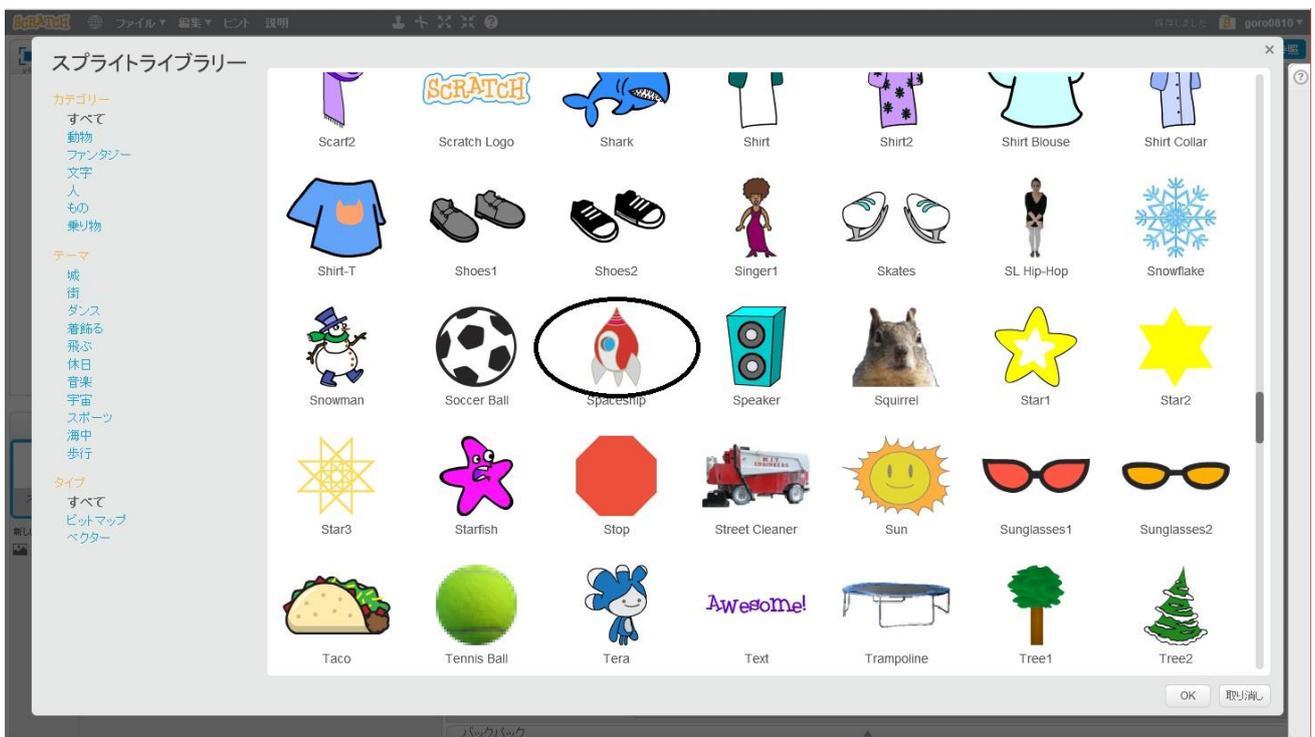
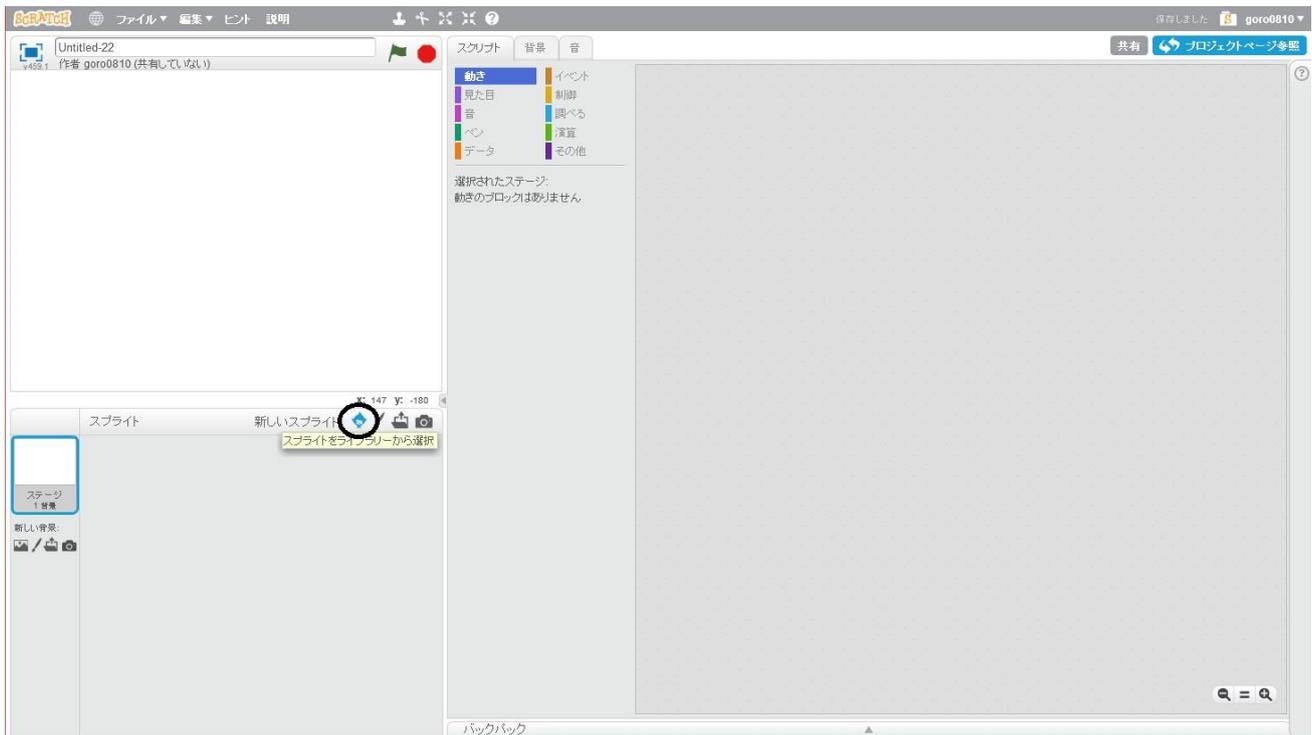


完成イメージをつかんでおきましょう。

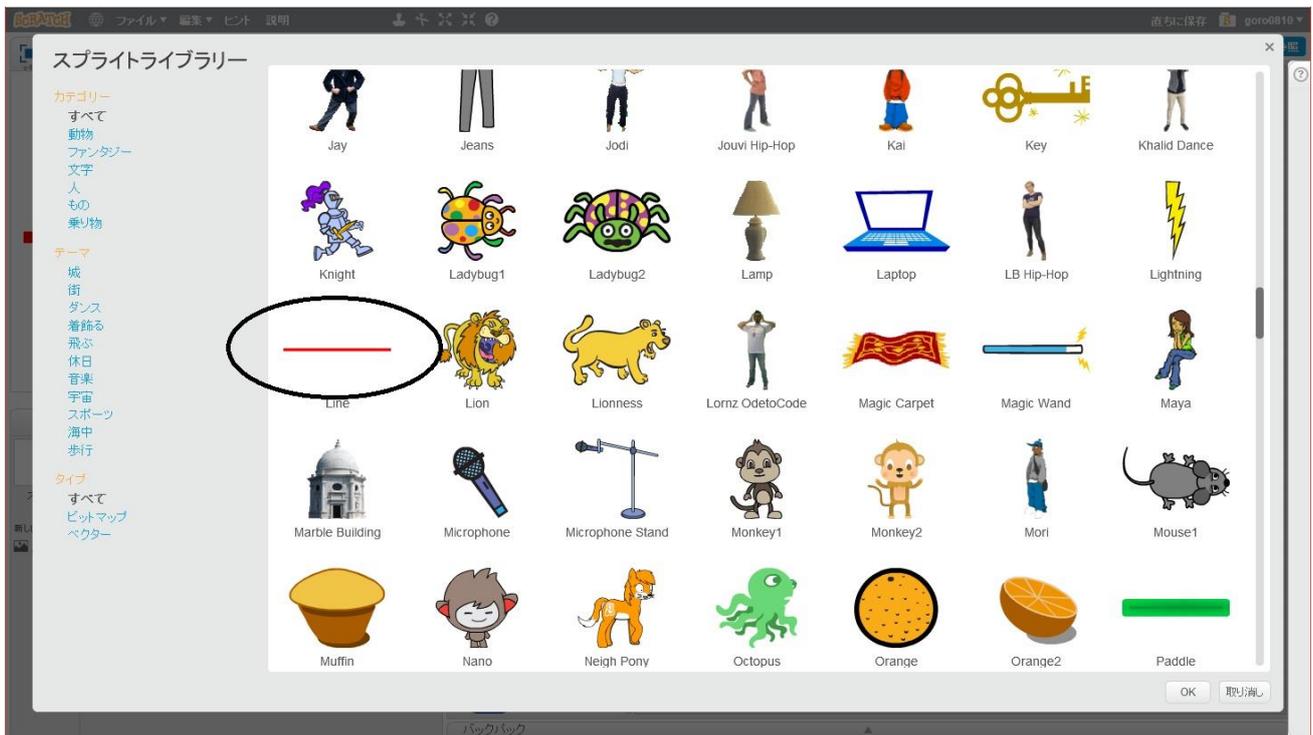
ではつくっていきましょう。

猫のキャラクターはいらないので、削除しましょう。

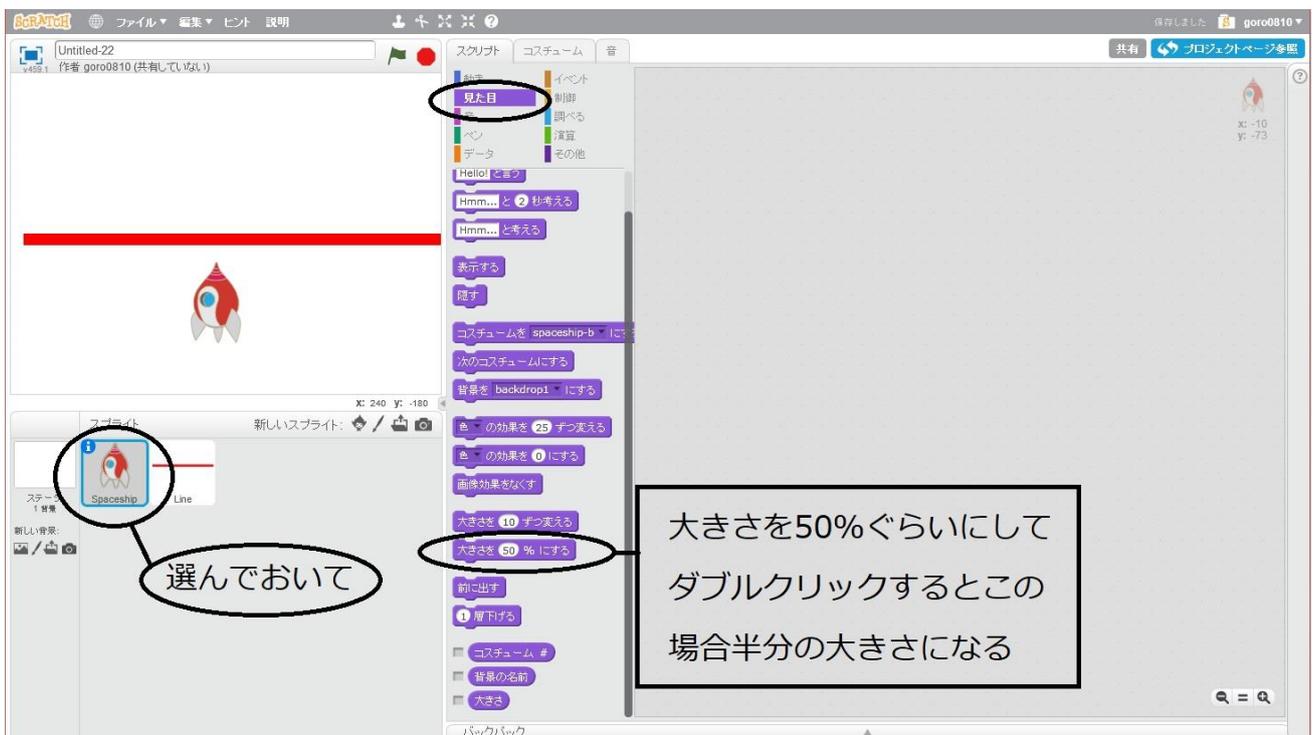
猫をクリックして、コスチュームで猫を選ぶと×印で削除していきましょう。消えないときは、猫の上で右クリック→削除でも消せることができます。



新しいスプライトの追加で Spaceship(自機)を選びます。

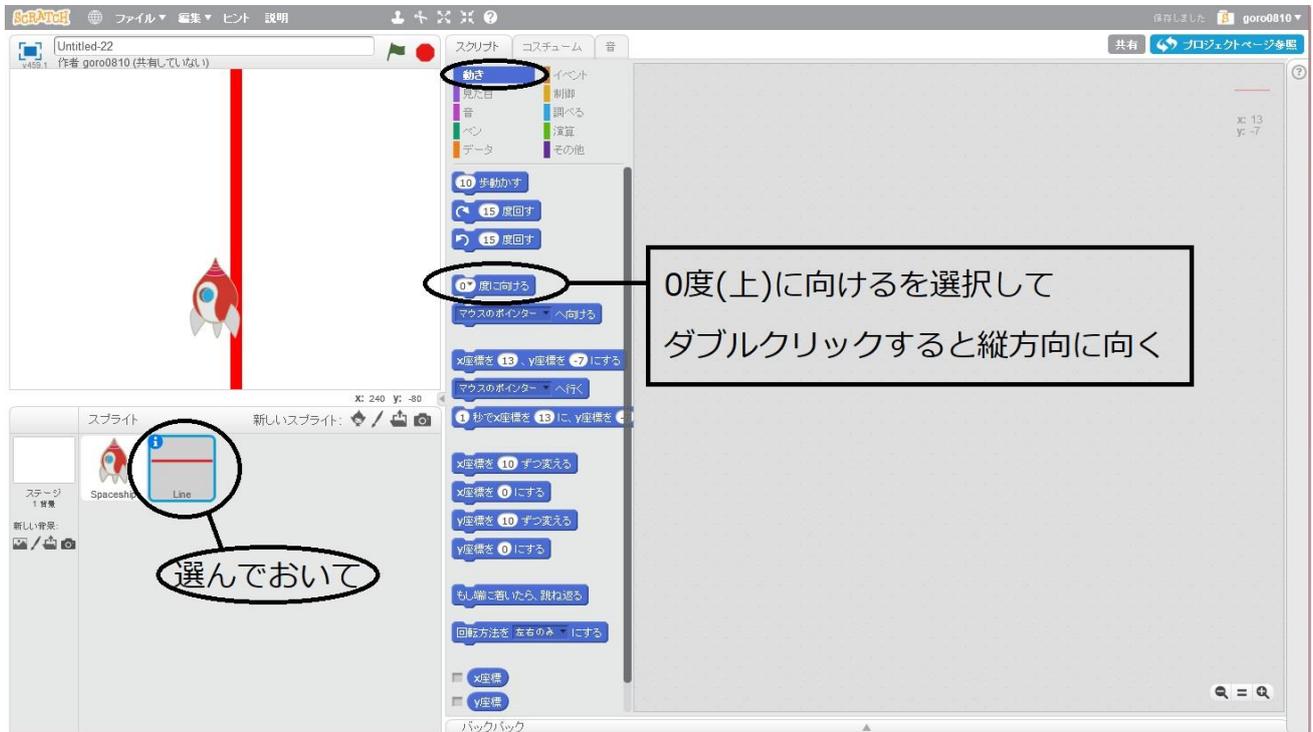


同じようにしてミサイルのもと Line を選択します。

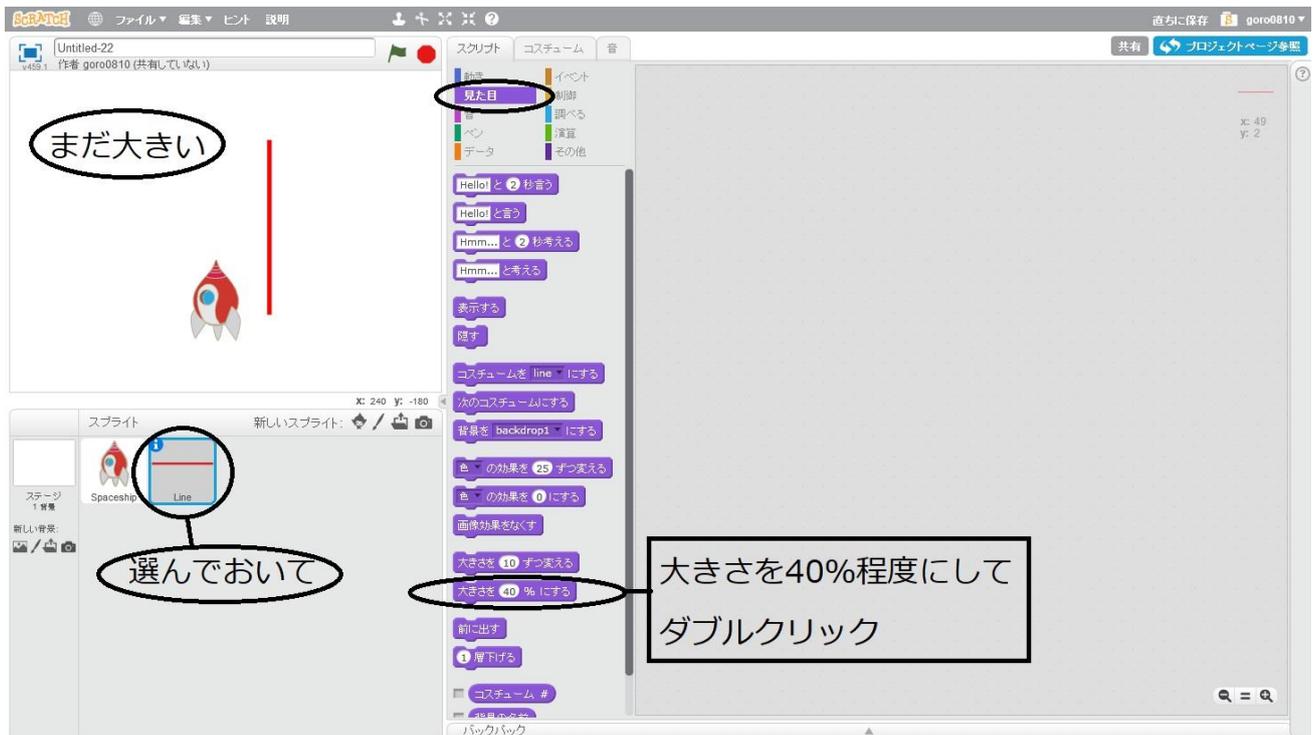


細かな調整はあとでするから、この作業は最後でいいよ。操作の方法を勉強するために、ここに載せています。

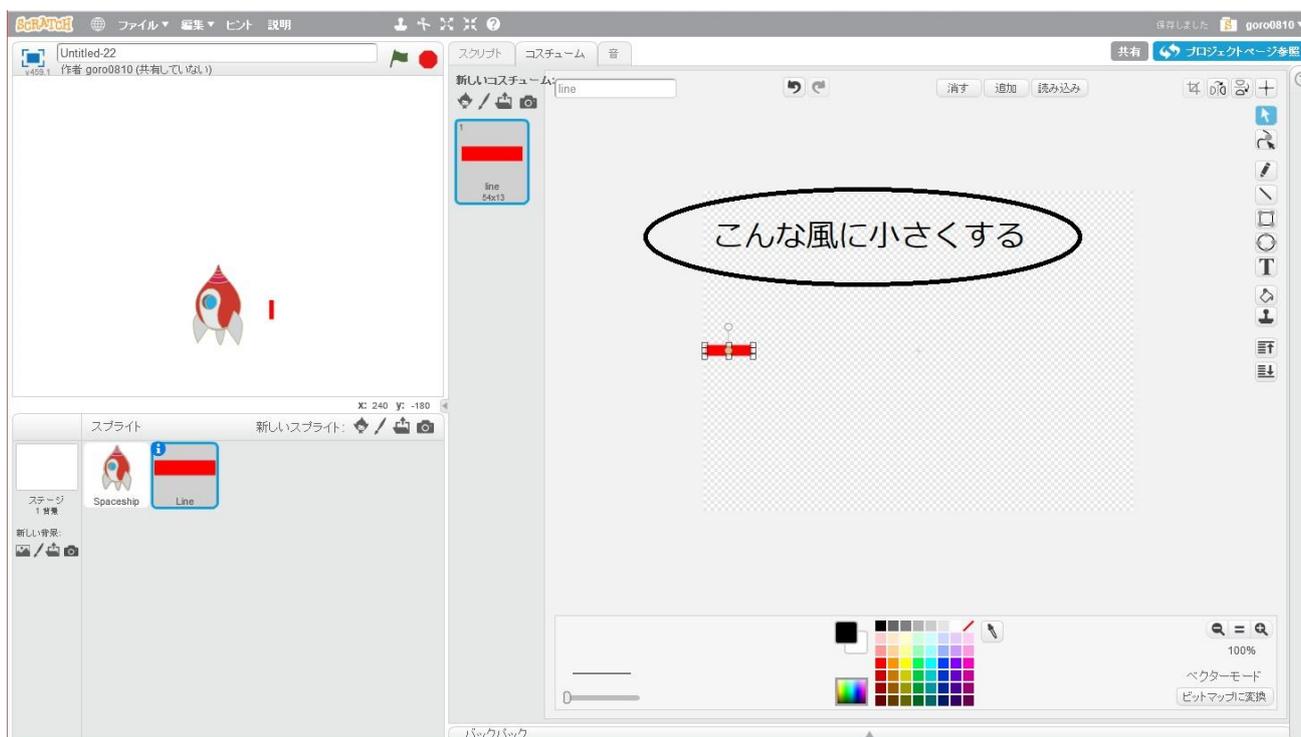
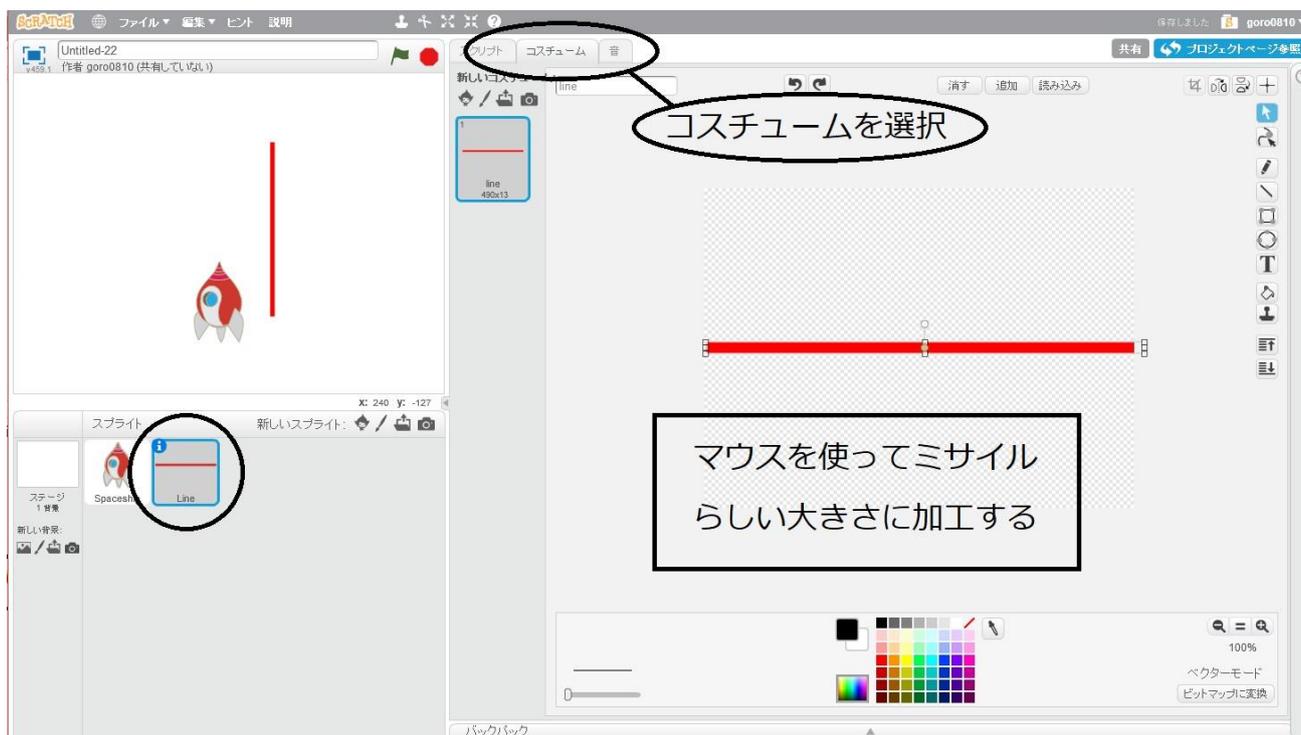
ミサイルが画面上で横になっているので、Line を選んでおいて、「動き」から 0 度(上)に向けるを選択する。そして青色の部分ダブルクリックすると Line が図のように縦に向きます。



ミサイルが大きいので調整しましたが、パーセントだとよくわからないので、うまくいかない。

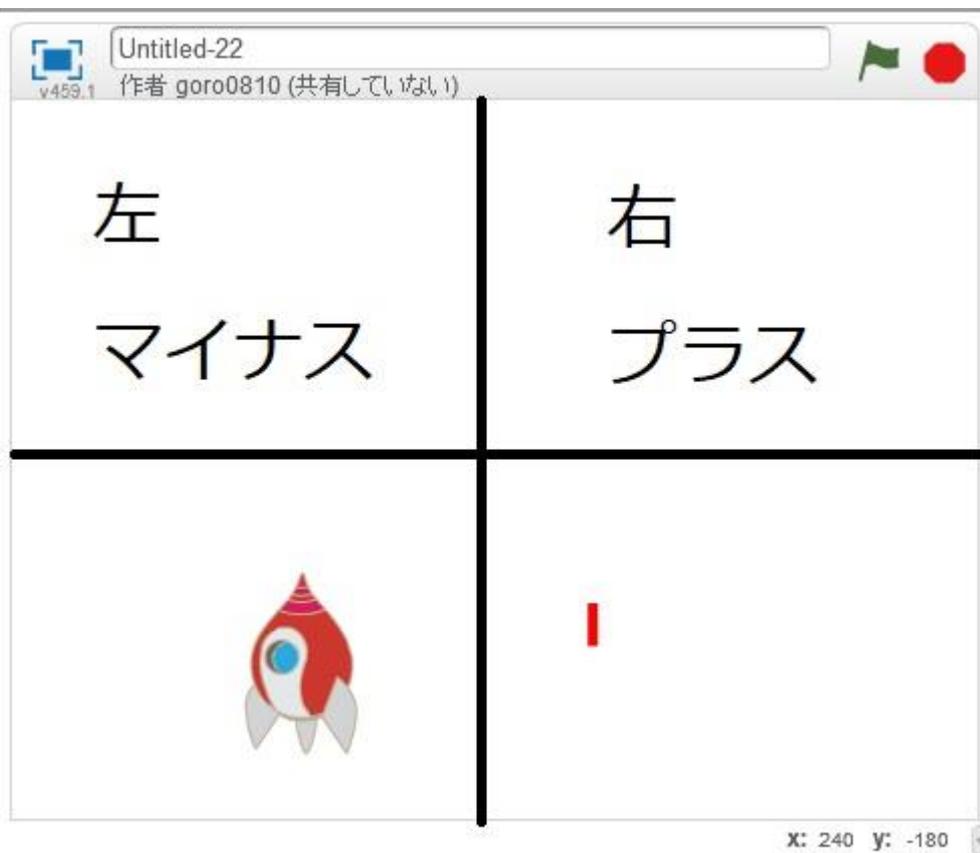


Line を選択して、コスチュームを選択。画面右側に Line の画像を加工できる場所があるから、大きさを換えよう。



出来ましたか？

続いてゲームの画面の説明。ゲーム画面を真ん中で縦に割って左側はマイナス、右側はプラスになっています。詳しくは3ページにあるから確認です。



それでは自機を動かすプログラミングを始めましょう。

自機(Spaceship)を選択し、スクリプトにプログラミングをしていきます。
イベントから「スペースキーが押されたとき」を選択し、右側へドラッグアンドドロップ、そして、スペースというところの▼を選択し、右向き矢印を選択する。右向き矢印では自機は右に動かしたいので、「動き」から 10 歩動かすを選んで図のようにつなぐ。これと同じようにして、左向き矢印を選択、-10 歩動かすと 10 のところをキーボードで-10 に変更します。これで自機は左右に動きます。

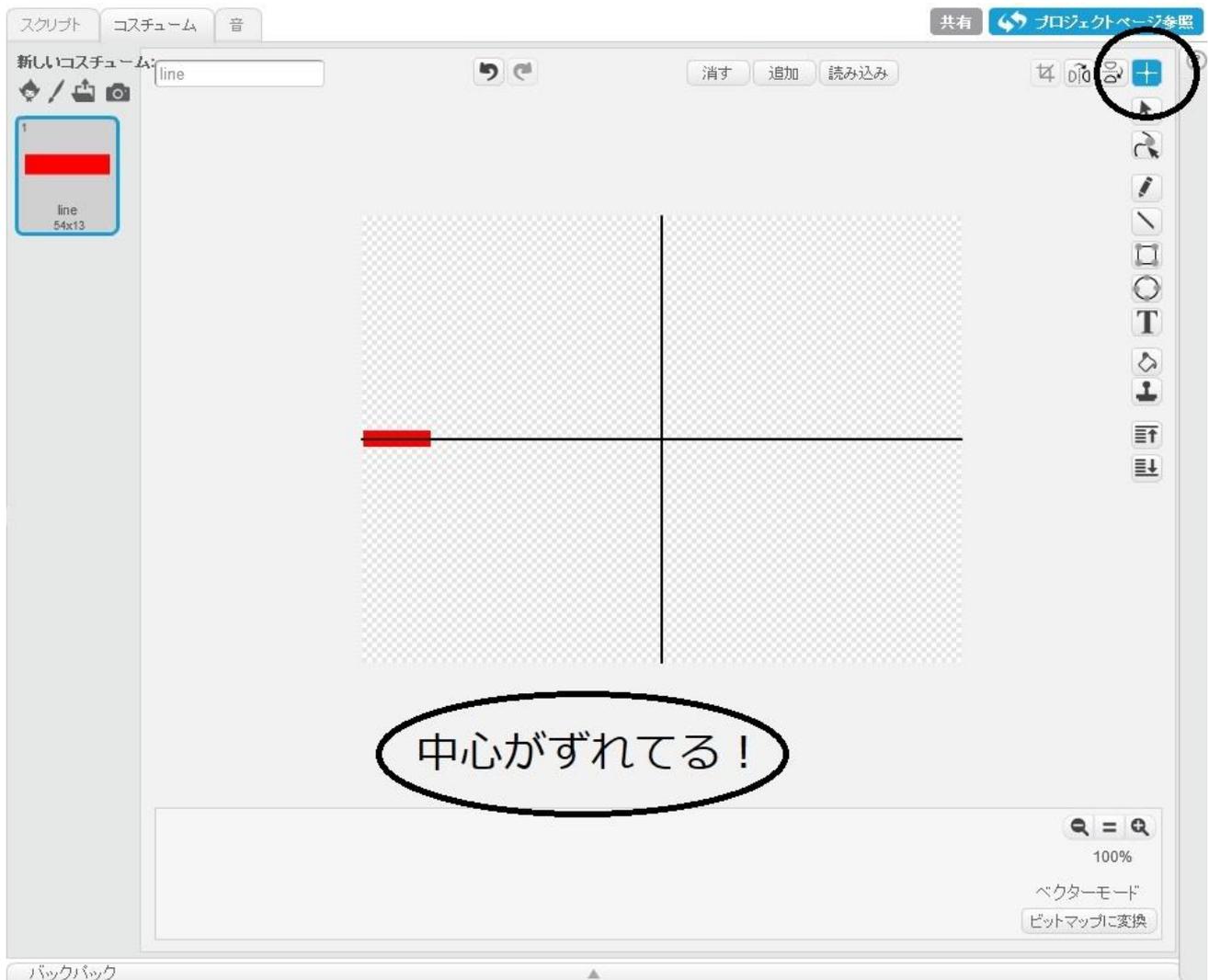
The screenshot shows the Scratch script editor with a spaceship sprite selected. The script contains two event-driven blocks: "When the space key is pressed" (右向き矢印) followed by "Move 10 steps right" (10 歩動かす), and "When the space key is pressed" (左向き矢印) followed by "Move -10 steps left" (-10 歩動かす). Handwritten annotations include: "イベントから選択" (Selected from event) pointing to the event blocks, "動きから選択" (Selected from movement) pointing to the movement blocks, and a red-bordered box containing the text: "-10歩動かすという意味は 左へ10歩動かすということ" (The meaning of -10 steps is moving 10 steps to the left). The left sidebar shows various block categories like "動き" (Movement) and "イベント" (Event). The top right shows "共有" (Share) and "プロジェクトページ参照" (View project page).

自機を高速で動かしたい場合は、10、-10 という数字を 20、-20 とすれば早くなります。もちろん 100、-100 ならもっと速く、反対に 3、-3 なら遅くなります。

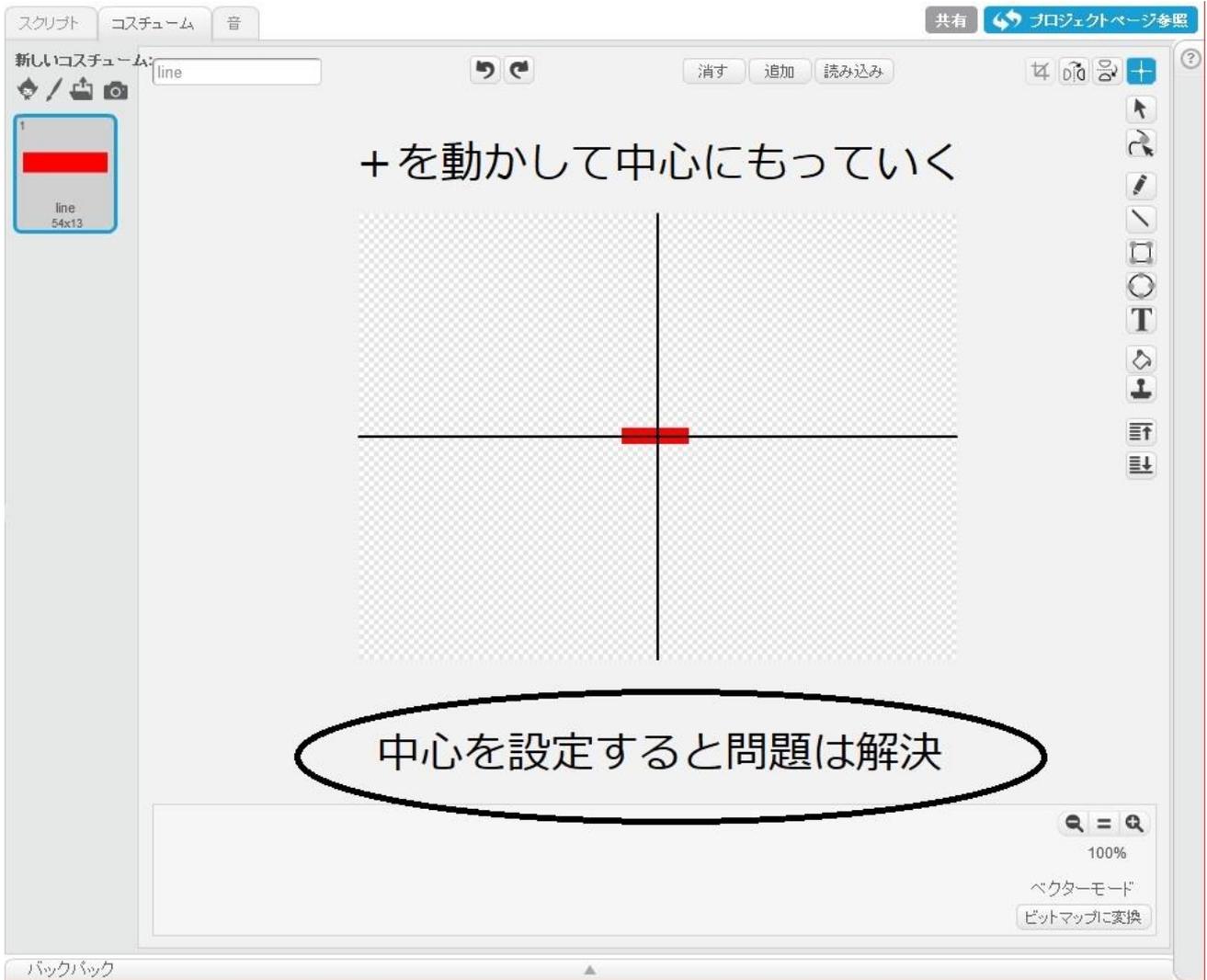
次にミサイルですが、ミサイルの発射ボタンを決めましょう。
今回はミサイルの発射ボタンはスペースキーということにします。
このキーの設定は後でも変更可能です。しかし、次のようにプログラムを組んで、スペースキーを押すと、ミサイルが自機の下にくるという問題が発覚しました。



ミサイルのコスチュームの内容を調べるために、コスチュームを選択し、右上の+を選ぶと、+の中心にミサイルがないことがわかりました。これが自機とミサイルがずれる原因でした。早速+を動かして調整しましょう。



+を動かして。





ずれは解消されました。

次はミサイルを飛ばすようにしましょう。

ミサイルは画面の端まで飛ばしたいので、次のようにプログラミングします。
これで、スペースキーが押されたら、壁に触れるまで上方向に10ずつ進んでいきます。
ただ、これでは壁に触れてもミサイルが消えることはありません。



そこで、壁に触れたあと、ミサイルを消すために「隠す」を追加します。
ただ、「隠す」だけを追加すると、今度はミサイルが隠れたままになって現れません。
そこで、「表示する」を「スペースキーが押されたとき」のすぐあとに追加します。



これでうまくいきました。

次に敵キャラの追加です。新しいスプライトの追加で、サッカーボールを追加します。なぜサッカーボールなのか・・・特に意味はありません。好きなキャラクターを選択してください。



スプライトのところから、サッカーボールを選択し、スクリプトを選択し、以下のプログラムを組んでいきます。この内容は、緑の旗(スタートボタン)がクリックされたら、サッカーボールを $x=0$ 、 $y=145$ に置いて、表示させなさい。そして、端に触れるまで、5 ずつ下に進めるといふものです。そして端に触れたら消す(隠す)。





スタートが押されたら

$x = 0$ 、 $y = 145$ のところに表示

端に触れるまで

y 座標を 5 減らしていく

端に触れたら隠す

先ほどのものでは、1度しかサッカーボールは出てきません。それではゲームになりません。そこで、何でも出現させるように以下のようにします。



The image shows a Scratch script for a soccer ball character. The script is divided into two main sections. The first section, triggered by a green flag click, sets the x-coordinate to 0 and the y-coordinate to 145, displays the character, and enters a loop that moves the y-coordinate down by 5 units until it reaches the bottom edge. The second section, triggered by receiving a message, sets the x-coordinate to a random value between -240 and 240, and the y-coordinate to 145, displays the character, and enters a loop that moves the y-coordinate down by 5 units until it reaches the bottom edge. A search bar in the bottom right corner shows 'x: 0' and 'y: 33'.

端に触れたらメッセージ1を送る

メッセージ1を受け取ったら、敵キャラの出現位置をxが-240~240の間で適当に出現させるようにプログラムする。

そして端に触れたらメッセージ1を送るようになる。

ここで少し修正。



右側のスクリプトにあった

「ずっと」というのはいらない。

端に触れるまでというのがあり、端に触れたら、メッセージ1を送るようにすることで繰り返しているから。

結果的にずっと繰り返すようになる。

では次は、ミサイルが敵キャラに当たったという処理を追加しましょう。ミサイルは赤色なので、赤に触れたら消す(隠す)ようにプログラムを追加します。

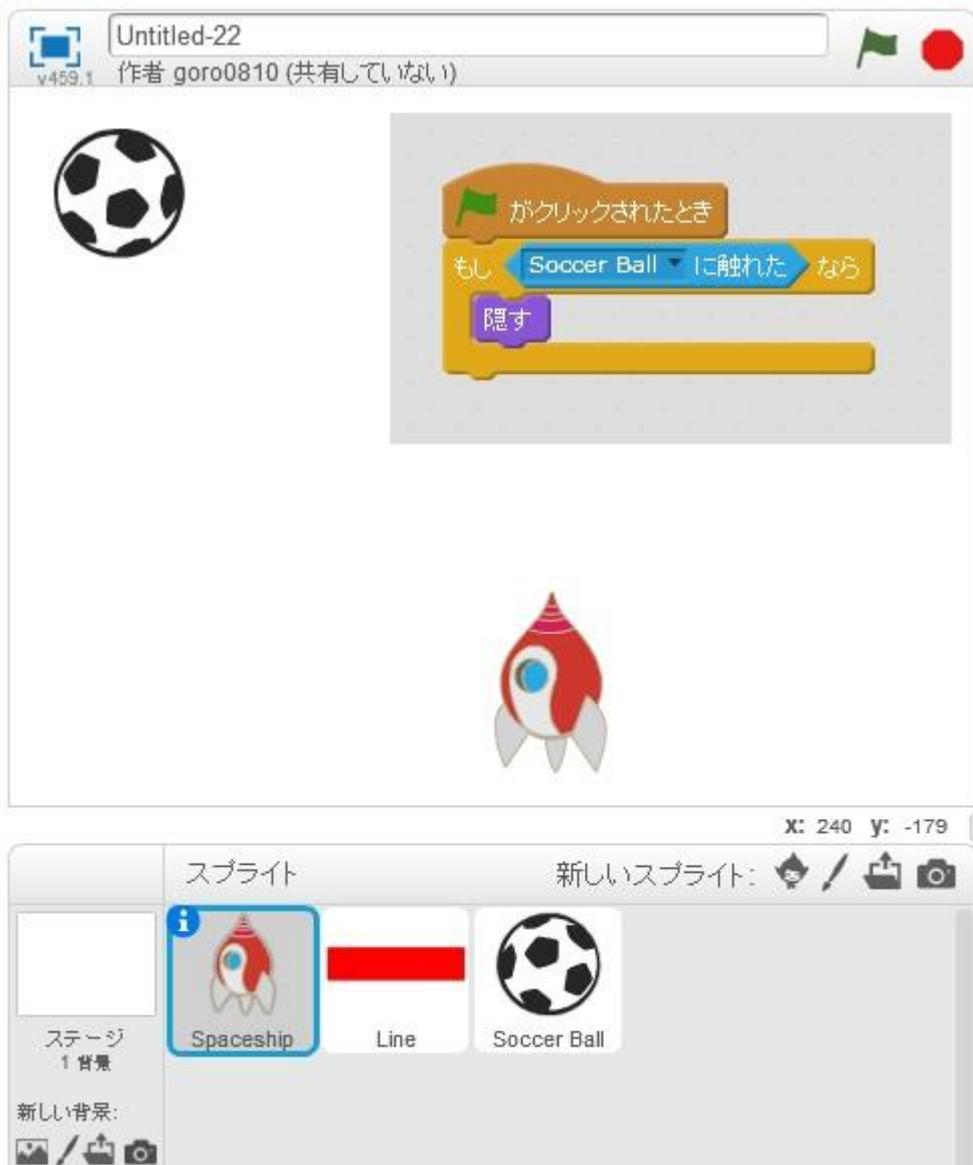


The image shows two Scratch code blocks. The left block is triggered by 'when clicked' and contains: 'show', 'set x to 0, y to 145', a 'while loop' with 'when reached edge' condition, 'decrease y by 5', and 'if red touched then hide'. The right block is triggered by 'when message 1 received' and contains: 'set x to random from -240 to 240, y to 145', 'show', a 'while loop' with 'when reached edge' condition, 'decrease y by 5', and 'if red touched then hide'. A soccer ball icon in the top right corner has coordinates x: 4, y: -47.

敵キャラがミサイルに当たったら隠す(消す)ようにします。その処理は端に触れるまでにしなくてはいけないので、上のように赤(ミサイル)に触れたら隠す(消す)とプログラミングします。

これでミサイルに当たったら、敵キャラ(サッカーボール)は消えます。

次は敵キャラ(サッカーボール)に触れたら自機を消す(隠す)というプログラムを追加します。ただ、下のプログラムは不完全で、これではスタートが押されたときしか有効になりません。



これではスタートが押されたときにサッカーボールに当たっていないと自機は消えません。

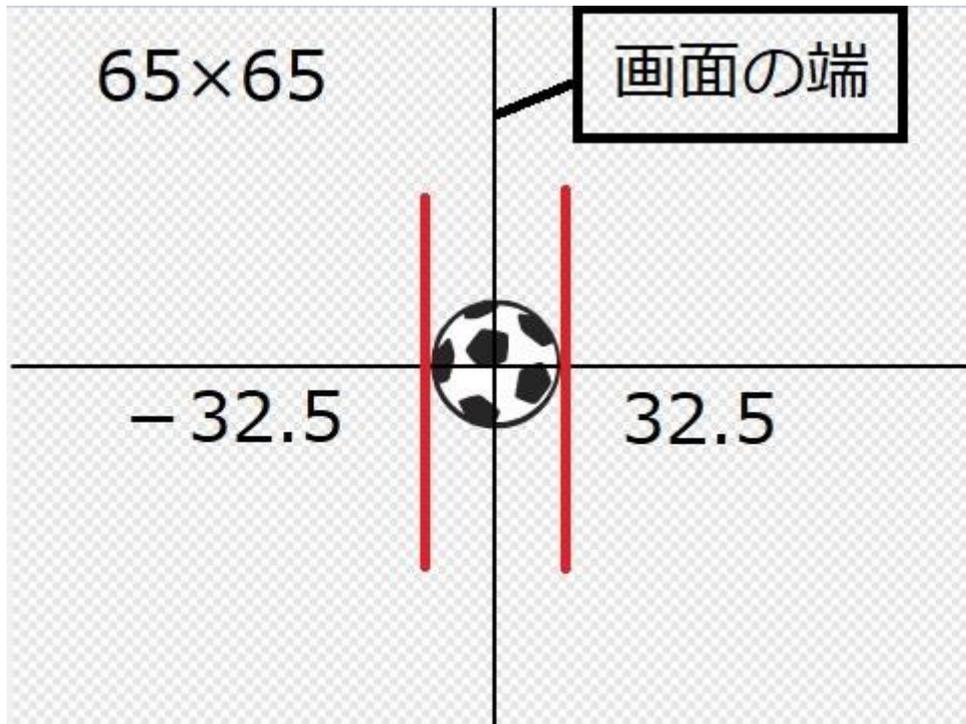
次のように修正しましょう。



サッカーボールに触れたかどうかずっと調べるようにします。そして、触れたら隠すようにします。そのとき、スタートボタンが押されたら自機を表示するのを忘れないように。

これでよいはずのプログラムを、スタートボタンを押して始めると、バグ？に出くわすでしょう。それは初めの方に、 -240 から 240 の範囲の乱数というのがありますが、このとき、乱数の数字が 240 だとすると、表示された時点で壁に触れてしまいます。それは 240 という数字はボールの中心を 240 にしなさいという命令のため、ボール画像幅の半分が画面からはみでることになってしまうからです。これではうまくいきません。

このプログラムも修正対象です。



ボールの縦のラインが画面の端にくると
画面の端から32.5はみ出てしまう

ボールの幅より少し大きい数字分、左右に余裕を持たせましょう。

プログラムを修正しました。



左ではサッカーボールが
±240のところにくると下に行かず消える。



ボールの幅の半分より少し
大きい37を左右に余裕を
持たせて、問題を解消しま
した。

ちなみに画面上の方は $y=180$ が一番上ですが、
ボールの幅約33を180から引くと $y=147$ となり、こ
こでは145となっているので、上にはみ出すことはない

乱数を使って、ゲームの開始時もサッカーボールがどこに現れるか予測不能としましょう。



ゲーム開始のときに
サッカーボールが現れる
位置もランダムにかえましょ
う。

自機や敵キャラの大きさの調整をしましょう。

Untitled-22
v459.1 作者 goro0810 (共有していません)

スクリプト コスチューム 音

動き イベント
見た目 制御
音 調べる
ペン 演算
データ その他

Hello! と 2 秒言う
Hello! と言う
Hmm... と 2 秒考える
Hmm... と考える
表示する
隠す
コスチュームを spaceship-b に設定
次のコスチュームにする
背景を backdrop1 にする
色 の効果を 25 ずつ変える
色 の効果を 0 にする
画像効果をなくす
大きさを 10 ずつ変える
大きさを 25 % にする
前に出す
1 層下げる
コスチューム #
名前

スプライト 新しい

ステージ 背景
新しい背景:
Sprites: Spaceship Line Soccer Ball

スプライトの大きさを調整しよう。

これで基本は出来上がりです。

ここからは、複製などを用いて敵キャラの数を増やしたり、背景の画像の交換などをしていきましょう。



これでサッカーボールが2つできました。

これによって、はじめにあったサッカーボールがサッカーボール1、複製したのがサッカーボール2というように区別されます。

このままでは、サッカーボール1とサッカーボール2が同じタイミングで出現しますから、プログラミングで、出現に時間差を与えます。またメッセージもメッセージ2に変更しておきましょう。図中の乱数というのは、上の場合、1.5から2までの間で適当(ランダム)な数字を発生させるというものです。下の場合、1.5から3までの間の適当(ランダム)な数字を発生させるということになります。この数字は予測できないので、面白くなるんです。

The image shows a Scratch script for two soccer balls. The script is divided into two main sections, one for each ball. Each section starts with a 'when clicked' event, followed by setting the x-coordinate to a random number between -203 and 203, and the y-coordinate to 145. The first ball has a 'wait 1.5 to 2 seconds' block, while the second ball has a 'wait 1.5 to 3 seconds' block. Both balls then have a 'show' block, followed by a 'when edge is reached' loop that moves the y-coordinate down by 5 units and checks for a red color collision. If a collision occurs, the ball is hidden. The second ball's script is triggered by a 'message 2 received' event, which is sent by the first ball's script. Annotations include a red box pointing to the wait blocks with the text 'ボールの出現を遅らせる' (Delay ball appearance) and a black box pointing to the message blocks with the text 'メッセージは2に変えておきます。そうではないともう1つのボールのメッセージ1に影響を受けます' (Change message to 2. It will be affected by message 1 of the other ball).

がクリックされたとき

x座標を -203 から 203 までの乱数、y座標を 145 にする

1.5 から 2 までの乱数 秒待つ

表示する

端に離れたまで繰り返す

y座標を -5 ずつ変える

もし 色に離れたなら

隠す

隠す

メッセージ2 を送る

メッセージ2 を受け取ったとき

x座標を -203 から 203 までの乱数、y座標を 145 にする

1.5 から 3 までの乱数 秒待つ

表示する

端に離れたまで繰り返す

y座標を -5 ずつ変える

もし 色に離れたなら

隠す

メッセージ2 を送る

ボールの出現を遅らせる

メッセージは2に変えておきます。そうではないともう1つのボールのメッセージ1に影響を受けます

ただ、このプログラムでは、ボールが現れて動き始めるまでの少しの間、止まって表示されます。これが面白いという方は、このプログラムを特に直す必要はありませんが、いやだという方は次のように修正していきましょう。



一旦隠しておいてから、時間がたてば表示するというように、プログラミングします。



The image shows two Scratch code blocks. The top block is triggered by a click event and contains the following steps: hide, set x and y coordinates to random values, wait 1.5 to 2 seconds, show, and then enter a loop where the ball moves down and hides if it hits the bottom. The bottom block is triggered by a message received and contains: hide, set x and y coordinates to random values, wait 1.5 to 3 seconds, show, and then enter a loop where the ball moves down and hides if it hits the bottom. A text box on the right explains the logic: the ball is hidden for a few seconds, and then shown after the time has passed.

まずボールを数秒間
隠して(待機させて)おい
て、その時間が過ぎた
ら、表示させるようにし
ました。

自機の方のスクリプトも変更します。
サッカーボール2に触れたら、隠すように追加します。



ここは、補足です。今までで気づいた方がいらっしゃるかもしれませんが、サッカーボールとミサイルの当たり判定で赤色に触れたなら、としていましたが、もちろん Line(ミサイル)に触れたなら、としても問題ありません。



赤色に触れたら隠すのではなく、
ミサイルに触れたら隠すに変更

ゲームの雰囲気をもつたために背景を交換しましょう。



宇宙らしくなりましたかね？

サッカーボールが気になる方は、何か別のキャラクターでやってみてください(笑)。

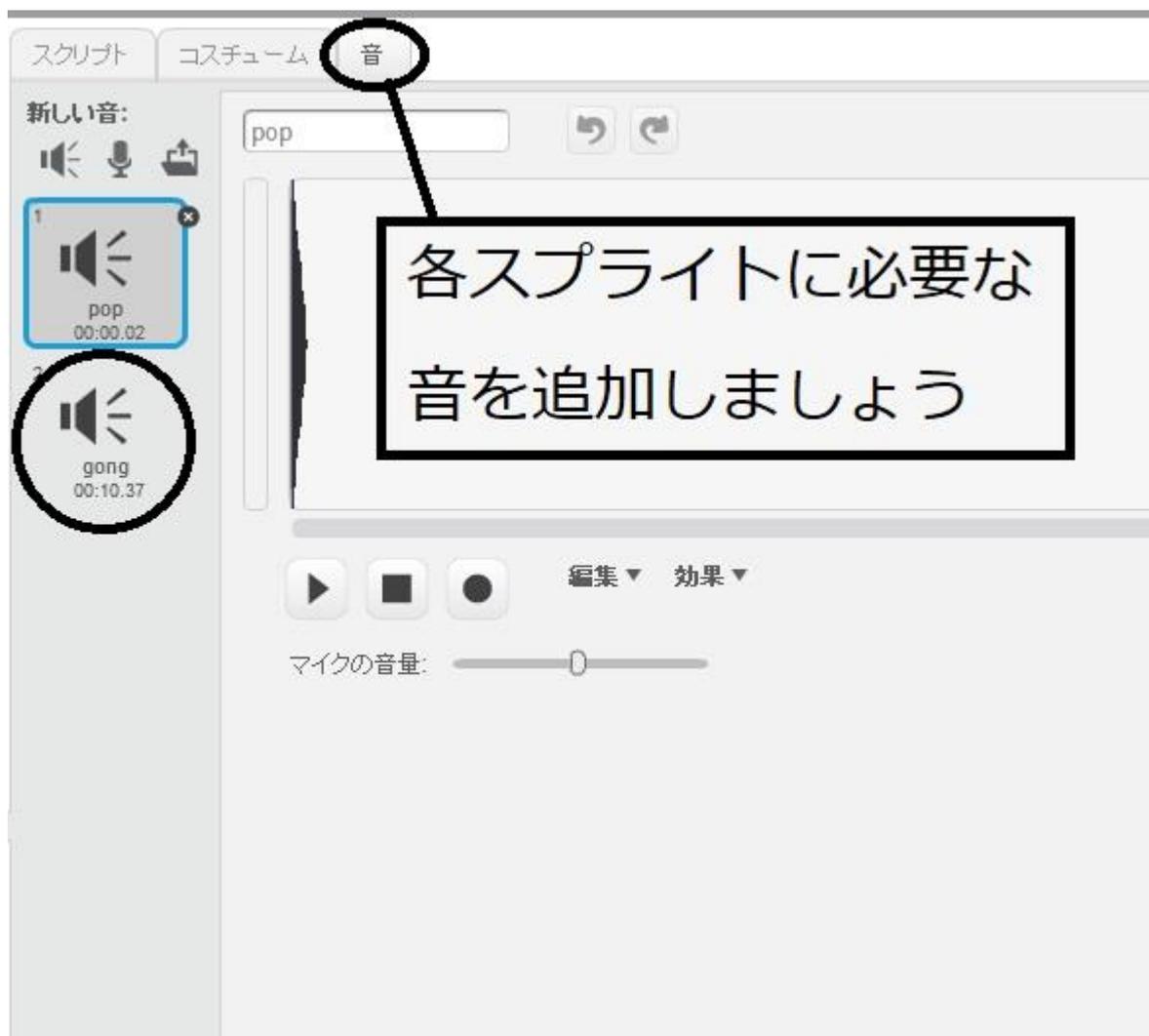
さらにゲームの雰囲気を楽しむために、音を追加しましょう。
今回は以下の設定でいきます。

crash cymbal	cricket	crickets	cymbal crash	cymbal echo	cymbal	dance around	c
lance snare beat	dance space	dog1	dog2	door creak	drip drop	drive around	
drum machine	drum satellite	drum set1	drum set2	drum	duck	eggs	ek
finger snap	flam snare	footsteps-1	garden	gong	goose	guitar chords1	
hi tun tabla	high conga	high hat	high tom	hihat beatbox	hihat cymbal	hip hop	
kick drum	large cowbell	laser1	laser2	laugh-female	laugh-male1	laugh-male2	

自機の消滅音

ミサイルの音

ボールの消滅音



そして、サッカーボールに触れたなら、音が出るようにプログラムしましょう。



Line(ミサイル)はスペースが押されたら音が出るようにプログラムしましょう。



サッカーボール 1 は Line(ミサイル)に触れたら音を出しましょう。



The image shows a Scratch script for a character named "Soccer Ball". The script is divided into two main sections: "when clicked" and "when message 1 received".

when clicked:

- Set x coordinate to a random number between -203 and 203, and y coordinate to 145.
- Show the character.
- Repeat until the "end" condition is met:
 - Decrease the y coordinate by 5.
 - If it touches the "Line" object:
 - Play the "laser2" sound.
 - Hide the character.
- Hide the character.
- Send message 1.

when message 1 received:

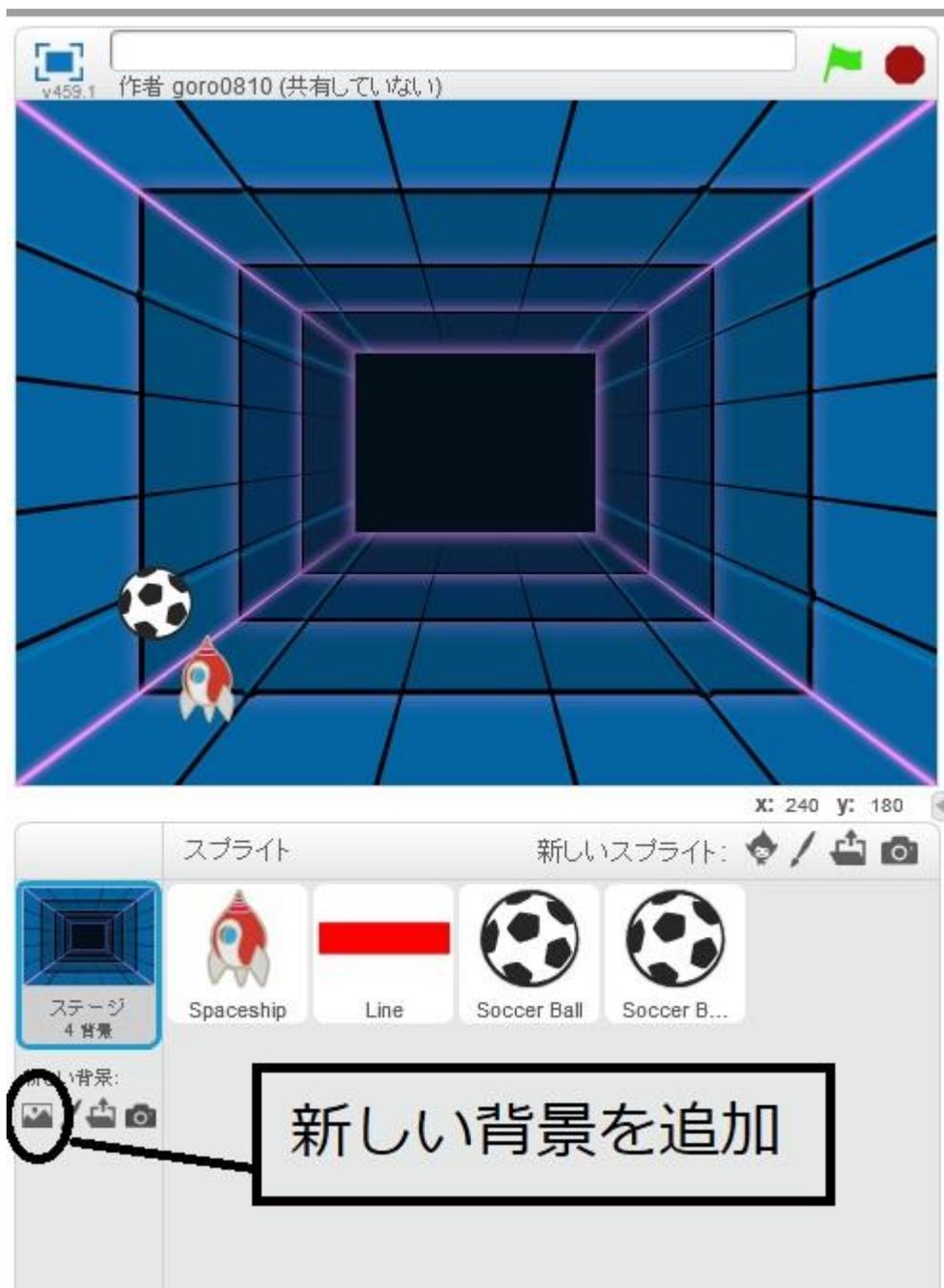
- Set x coordinate to a random number between -203 and 203, and y coordinate to 145.
- Show the character.
- Repeat until the "end" condition is met:
 - Decrease the y coordinate by 5.
 - If it touches the "Line" object:
 - Play the "laser2" sound.
 - Hide the character.
- Send message 1.

In the top right corner, there is a "Soccer Ball" character icon with a soccer ball pattern.

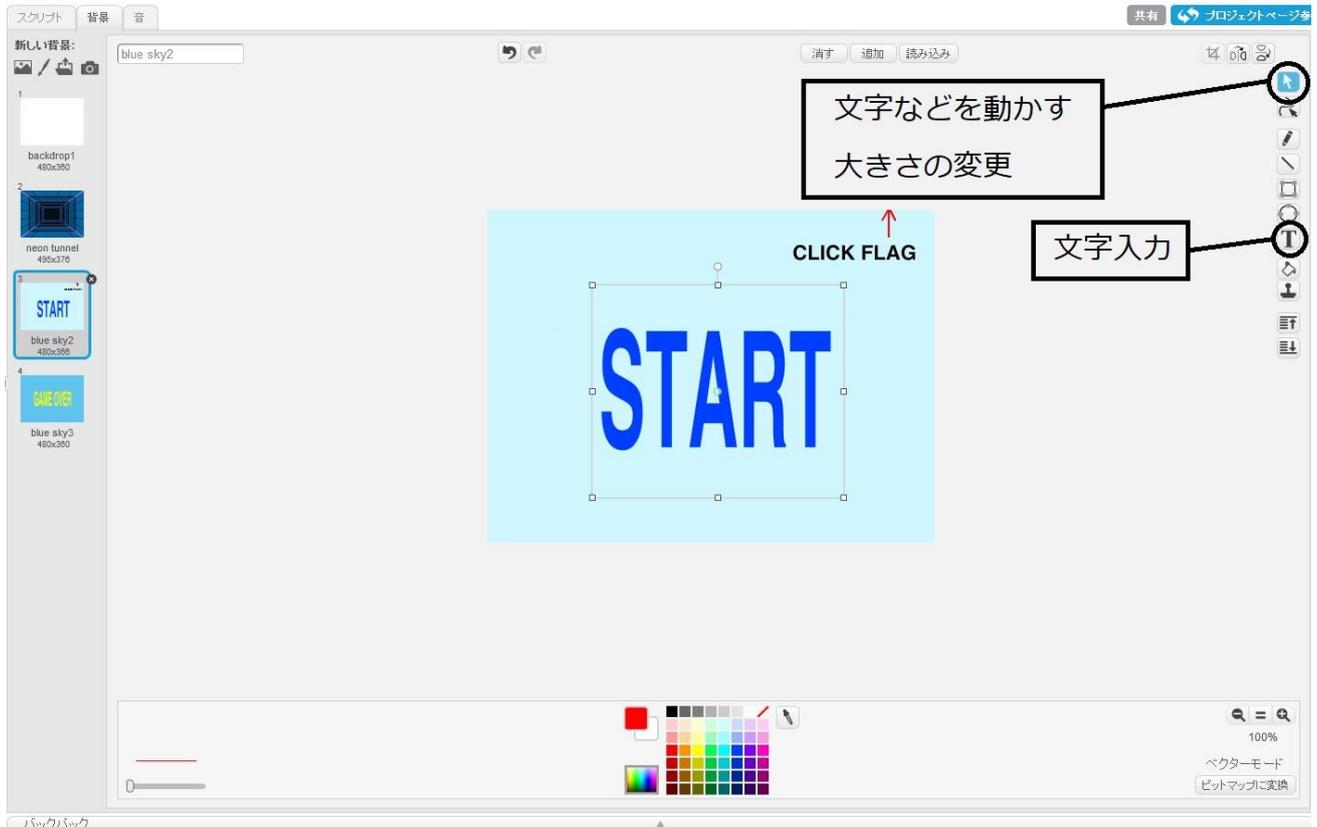
サッカーボール 2 も同じように、Line(ミサイル)に触れたら音を出しましょう。

Scratch script for a soccer ball character named "Soccer B...". The script is triggered by a click event and contains two identical loops. Each loop starts with a "hide" block, followed by a "set x coordinate to random number between -203 and 203, y coordinate to 145" block, and a "wait 1.5 to 2 seconds" block. Then, it enters a "show" block with a "loop until end reached" block. Inside the loop, there is a "set y coordinate to -5" block, followed by an "if Line touched" block. If true, it plays the "laser2" sound and then hides. The loop ends with a "send message 2" block. The second loop is triggered by a "receive message 2" event and follows the same sequence of actions.

ゲーム開始の画面と、ゲーム終了の画面をつくりましょう。



新しい背景を選んで、T(テキスト)を選んで、START とかきましょう。
開始がわかれば入力の文字は何でもいいですよ。下のように開始と終了をつくりましょう。



開始



終了



そして、以下のように画面の切り替えの設定、音の切れ目の間隔の調整などしていきます。以下はその説明です。

また、サッカーボールに当たったらゲームオーバーなので、ゲームオーバーになったというフラグ(合図)を立てます。それが図の中の「ゲーム終わりを送る」です。



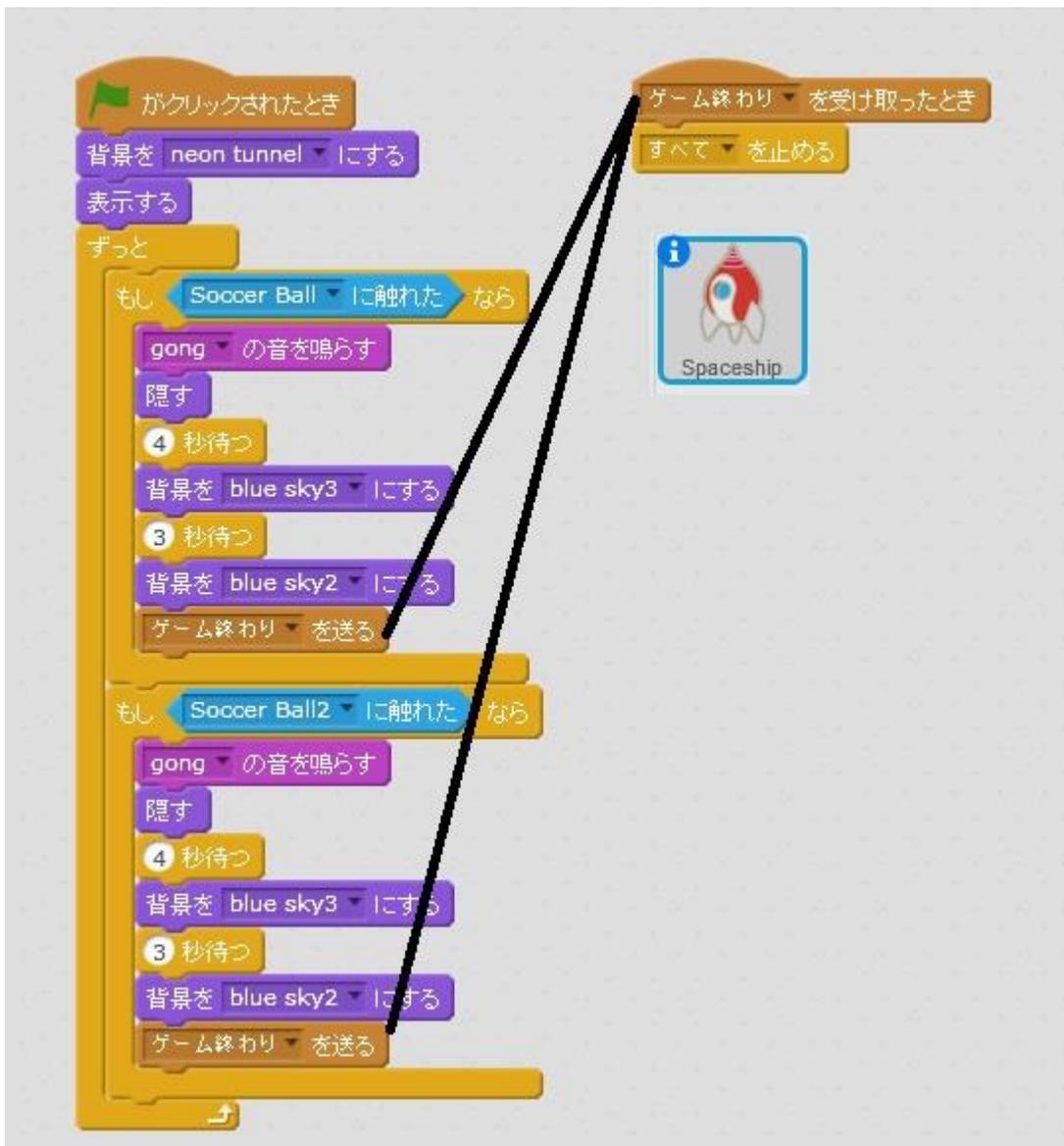
The image shows a Scratch script for a game. At the top, there is a 'SpaceShip' sprite. The script starts with a 'when clicked' event, followed by 'set background to neon tunnel', 'show', and a 'forever' loop. Inside the loop, there are two 'if Soccer Ball touches' conditions. Each condition triggers a 'gong sound', 'hide', a '4 second wait', 'set background to blue sky3', a '3 second wait', 'set background to blue sky2', and 'send game over message'. The script ends with a 'send game over message' block.

Annotations in the image include:

- A black box pointing to the 'set background to neon tunnel' block with the text: **ゲーム開始時は黒い画面にする**
- A red box pointing to the 'set background to blue sky2' blocks with the text: **GAME OVER画面** and **START画面**
- A black box pointing to the '4 second wait' and '3 second wait' blocks with the text: **時間を挟んで、音の切れ目や画面の切り替えを調整**

最後にゲームが終了すると、すべてのスクリプトを停止させます。これを「ゲーム終わりを送る」を合図として行います。

次のように「ゲーム終わりを受け取る」とすべてを停止させるようにします。



サッカーボールはゲームが終わると隠します。そのため次のように、「ゲーム終わり」を受け取ると隠すようにプログラムします。1つ1つのサッカーボールをそのようにプログラムします。



これでゲームの完成です。動かして楽しんでみてくださいね。

各プライトのスク립トをまとめておきます。
自機

The image shows a Scratch script for a character named 'Spaceship'. The script is organized into several sections:

- Right Arrow Key Press:** When the right arrow key is pressed, the character moves 20 steps to the right.
- Left Arrow Key Press:** When the left arrow key is pressed, the character moves 20 steps to the left.
- Click Event:** When the green flag is clicked, the background is set to 'neon tunnel', the character is shown, and a loop begins.
- Collision with Soccer Ball:** Inside the loop, if the character touches 'Soccer Ball', it plays a 'gong' sound, mutes it, waits 4 seconds, sets the background to 'blue sky3', waits 3 seconds, sets the background to 'blue sky2', and sends a 'game over' message.
- Collision with Soccer Ball2:** Similarly, if the character touches 'Soccer Ball2', it plays a 'gong' sound, mutes it, waits 4 seconds, sets the background to 'blue sky3', waits 3 seconds, sets the background to 'blue sky2', and sends a 'game over' message.

Additional elements include a 'Game Over' message block (when received, stop everything) and a 'Spaceship' character icon.

ミサイル



サッカーボール 1

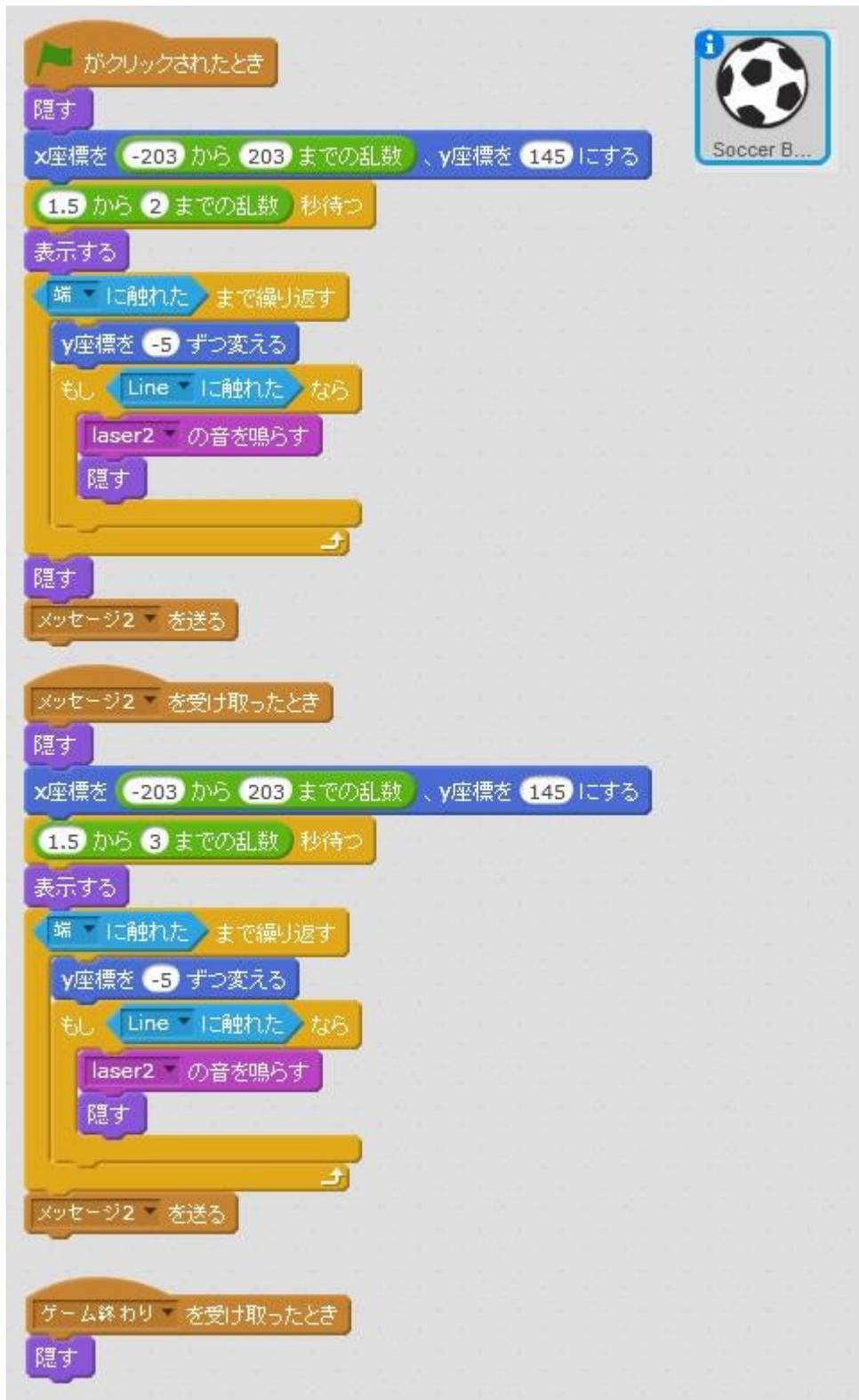


The image shows a Scratch script for a soccer ball character. The script is organized into three main sections, each starting with a 'when triggered' event block.

- When clicked:** This section starts with a 'when clicked' event block. It then sets the x-coordinate to a random number between -203 and 203, and the y-coordinate to 145. The ball is then shown. A 'repeat until' loop follows, with the condition 'edge reached'. Inside the loop, the y-coordinate is decreased by 5. A 'when line reached' block is used as a condition; if it is reached, a 'play sound' block for 'laser2' is executed, followed by a 'hide' block. After the loop, the ball is hidden and a message 'message1' is sent.
- When message1 received:** This section starts with a 'when message1 received' event block. It repeats the same sequence of actions as the 'when clicked' section: setting random coordinates, showing the ball, moving it down until it hits a line, playing a laser sound, and then hiding the ball and sending a message.
- When game over received:** This section starts with a 'when game over received' event block, followed by a 'hide' block to remove the ball from the stage.

In the top right corner, there is a 'Soccer Ball' character icon with an information symbol.

サッカーボール 2



The image shows a Scratch script for a soccer ball game. The script is organized into three event-driven sections: a click event, a message received event, and a game over event. Each section includes logic for hiding the ball, setting its position to a random x-coordinate between -203 and 203 and a fixed y-coordinate of 145, and a 1.5 to 2-second delay. A loop block handles collisions with the top and bottom boundaries, adjusting the y-coordinate by -5 and playing a laser sound when a collision occurs. The script also includes a 'hide' block and a 'send message' block at the end of each event sequence. A soccer ball icon labeled 'Soccer B...' is visible in the top right corner.

```
Scratch Script: Soccer Ball 2

Event: がクリックされたとき
  隠す
  x座標を -203 から 203 までの乱数、y座標を 145 にする
  1.5 から 2 までの乱数 秒待つ
  表示する
  端に離れたまで繰り返す
    y座標を -5 ずつ変える
    もし Lineに離れたなら
      laser2の音を鳴らす
      隠す
  隠す
  メッセージ2を送る

Event: メッセージ2を受け取ったとき
  隠す
  x座標を -203 から 203 までの乱数、y座標を 145 にする
  1.5 から 3 までの乱数 秒待つ
  表示する
  端に離れたまで繰り返す
    y座標を -5 ずつ変える
    もし Lineに離れたなら
      laser2の音を鳴らす
      隠す
  隠す
  メッセージ2を送る

Event: ゲーム終わりを受け取ったとき
  隠す
```