

スクラッチを使って シューティングゲームを つくろう 2



<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。

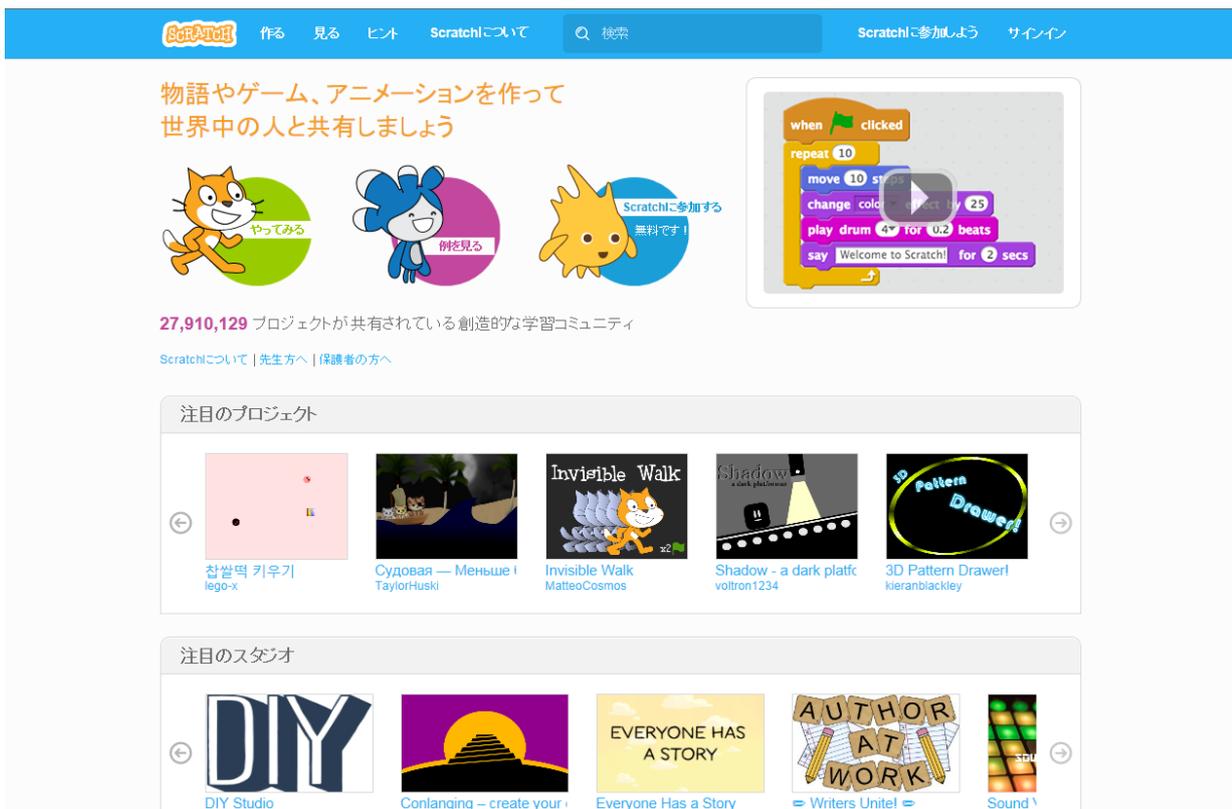
プログラミングができれば、人の役にものすごく役に立ちます。切符の券売機、インターネットオークション、スマホアプリ、Googleなどの検索画面など身近なものがほとんど、プログラミングで、できています。

プログラムが動く仕組み

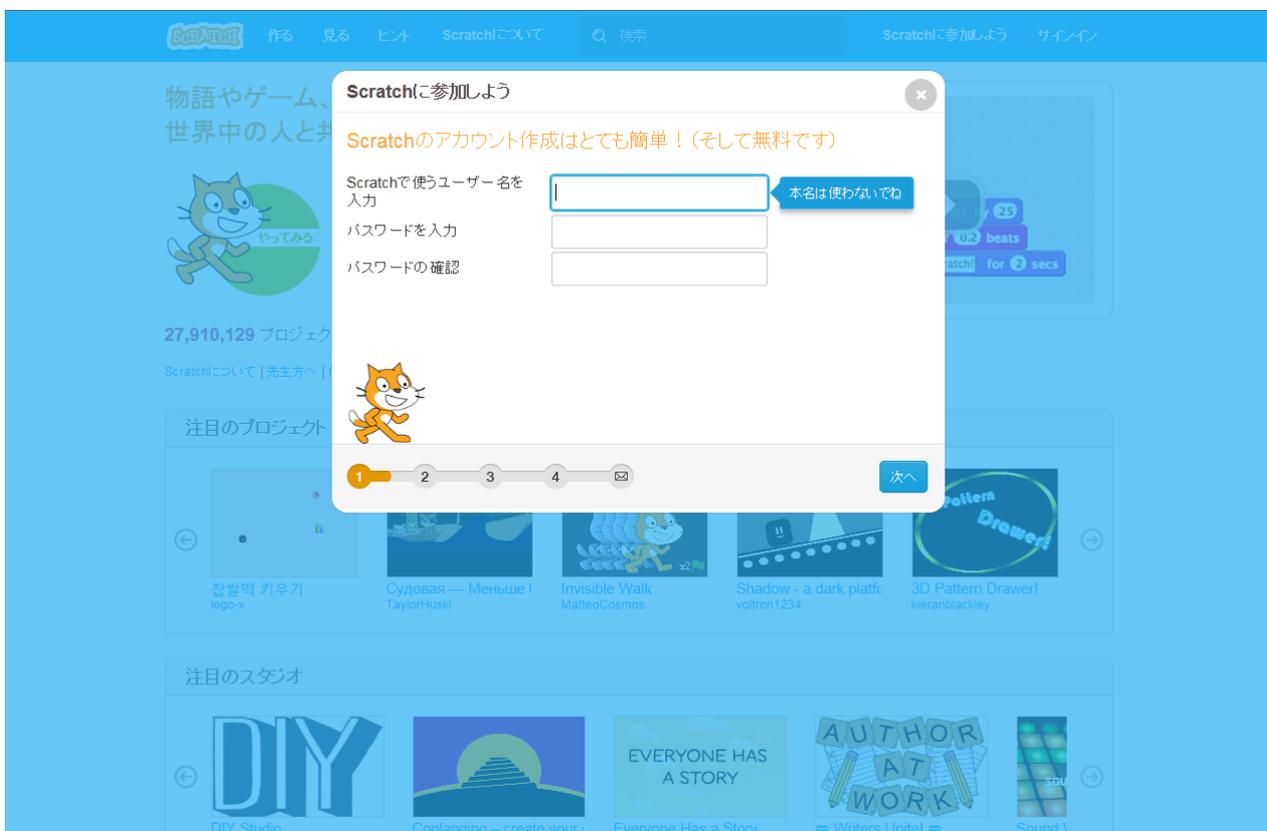
コンピュータなどの電子機械は信号があるかないかで動きます。専門的には2進数というもので動くことになります。どのように動くのか決めたのがマニュアルですが、コンピュータにはそのマニュアルは読めません。ですから、つくったマニュアルをコンピュータが読めるように電気信号に変換します。それがソフトウェアというものです。マニュアルをソフトウェアで電気信号に変えることで、コンピュータの頭脳であるCPUを動かして、期待する動作を得ることができます。

*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。

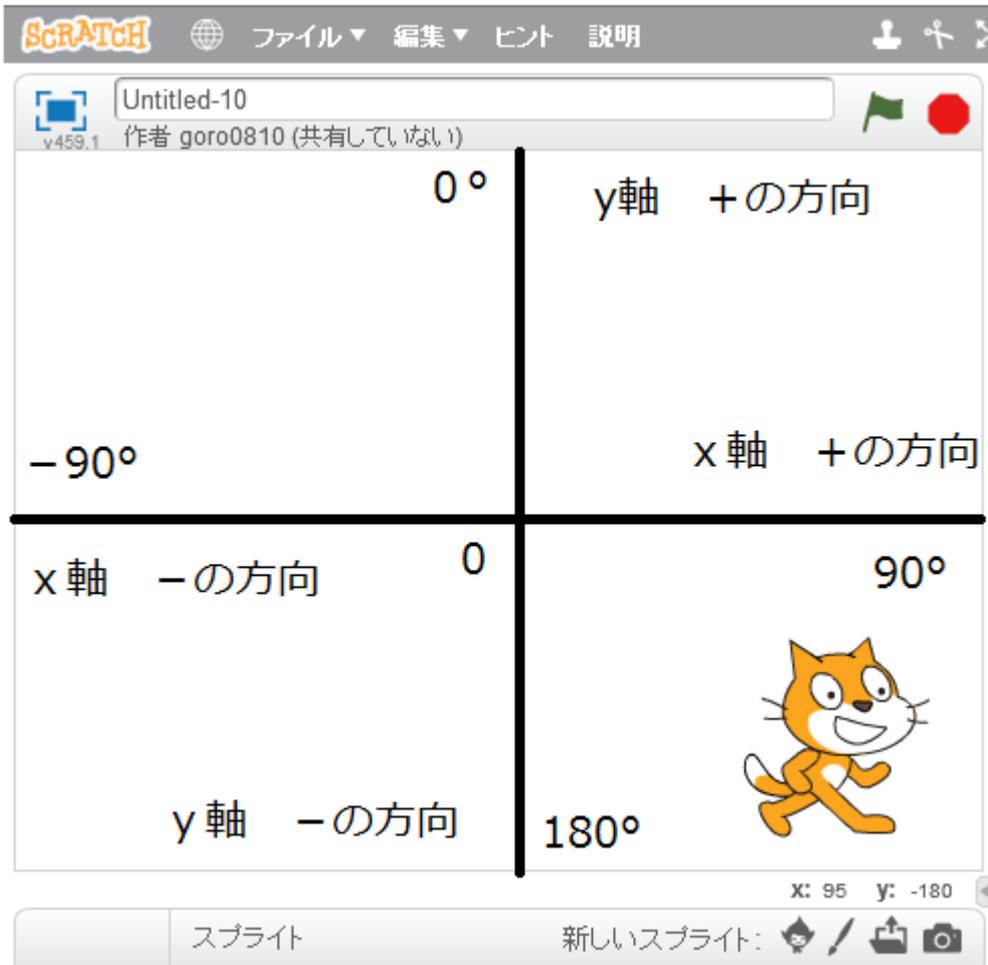
スクラッチを始めるには、スクラッチ MIT と検索をかけて、以下の画面に行きます。



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。

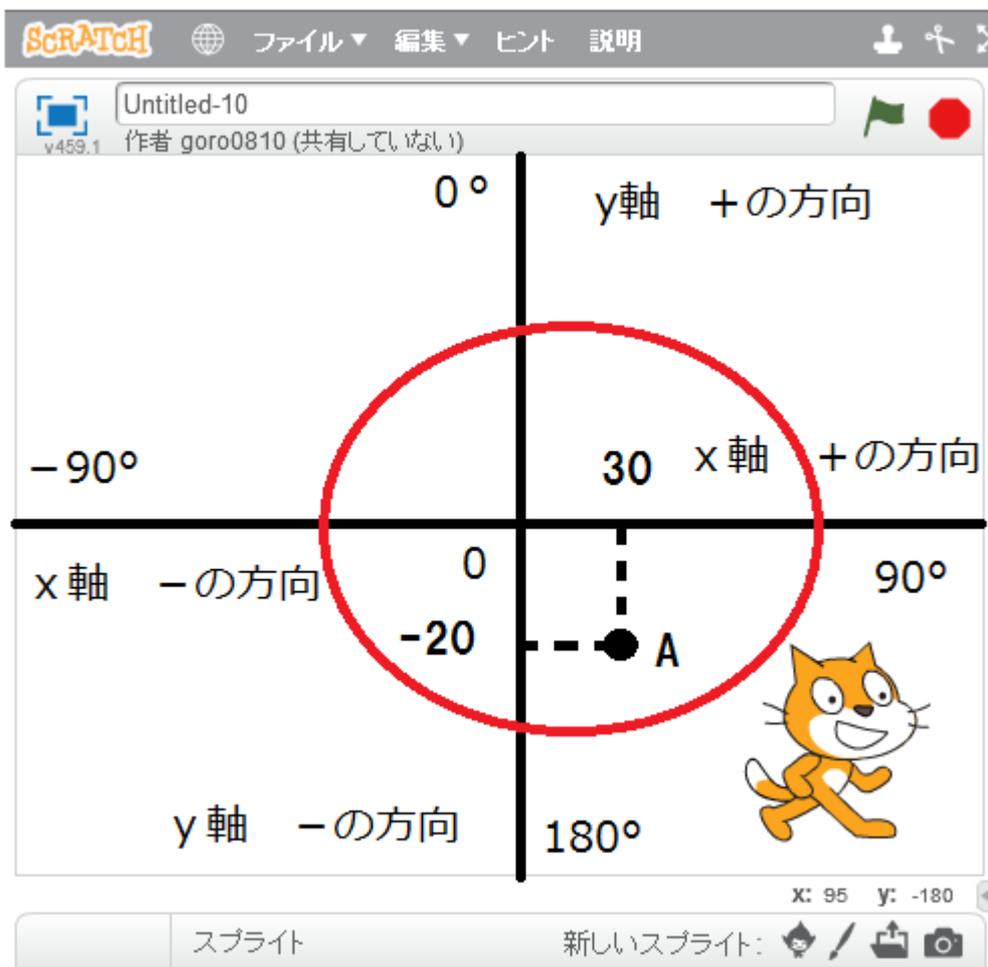


画面の中心が0というところです。0から横に伸びた線をx軸、0から縦に伸びた線をy軸といいます。

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に表示させたいければ、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。



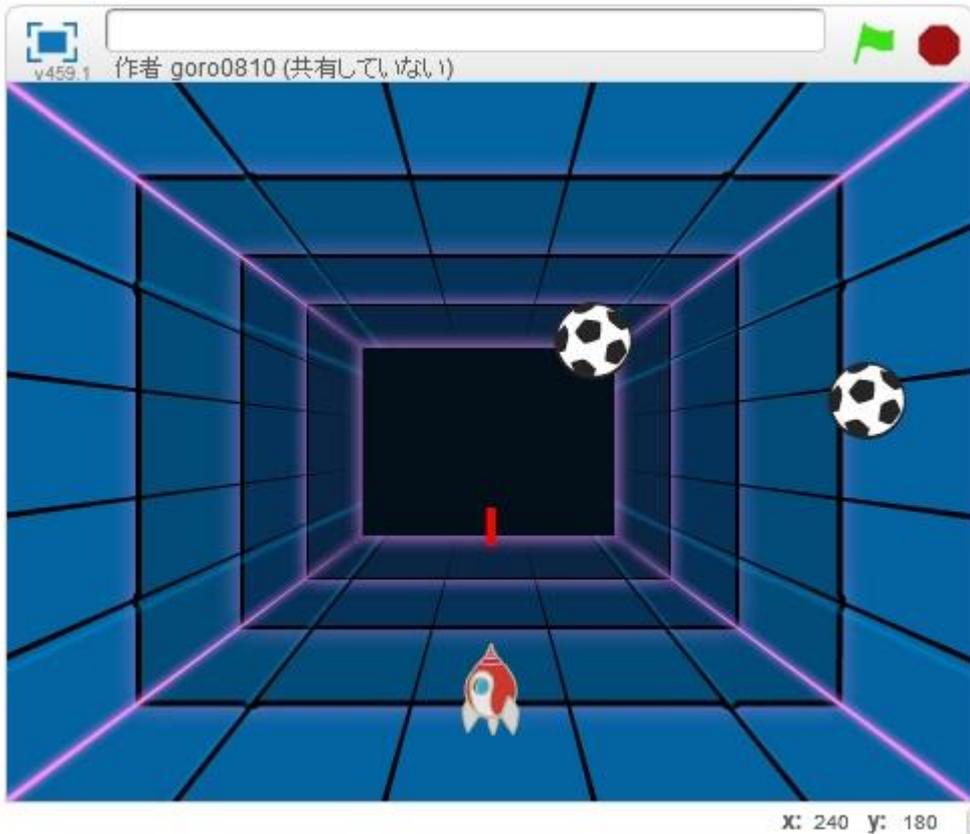
 このボタンは自分で作ったものをアップロードするときに使います。

 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。

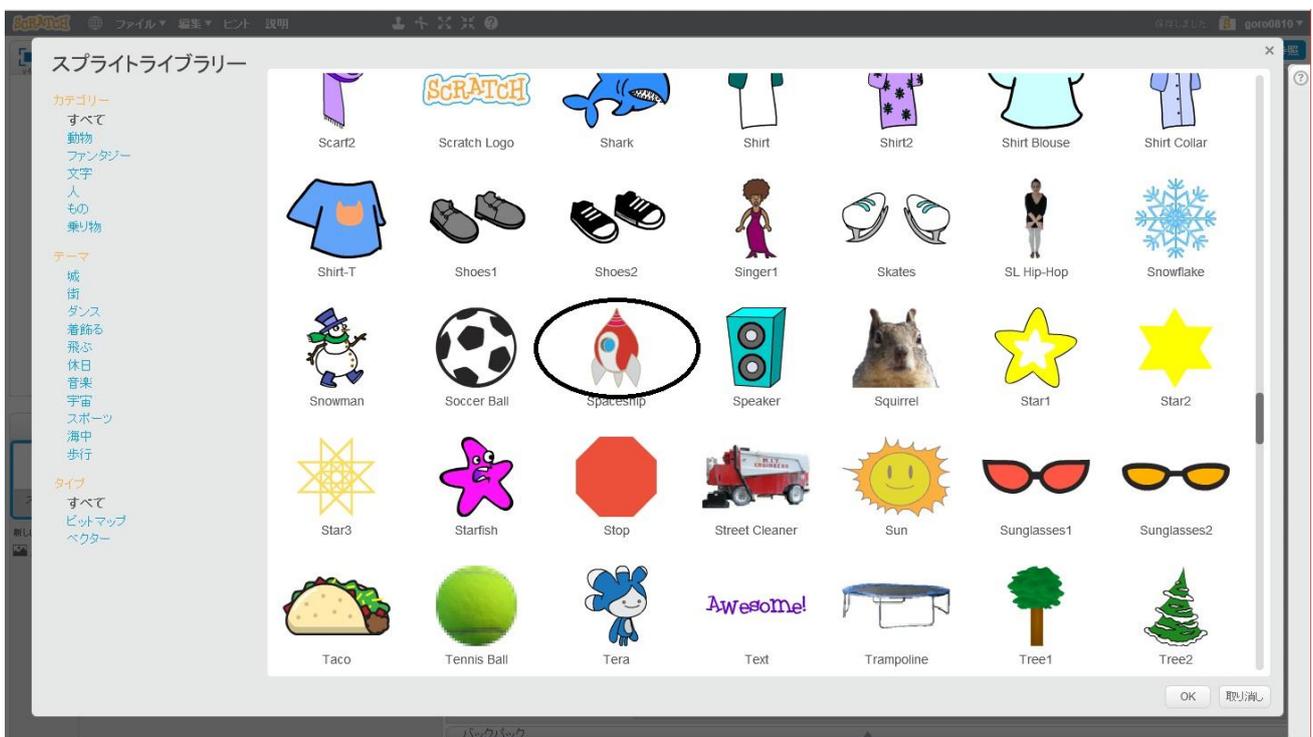
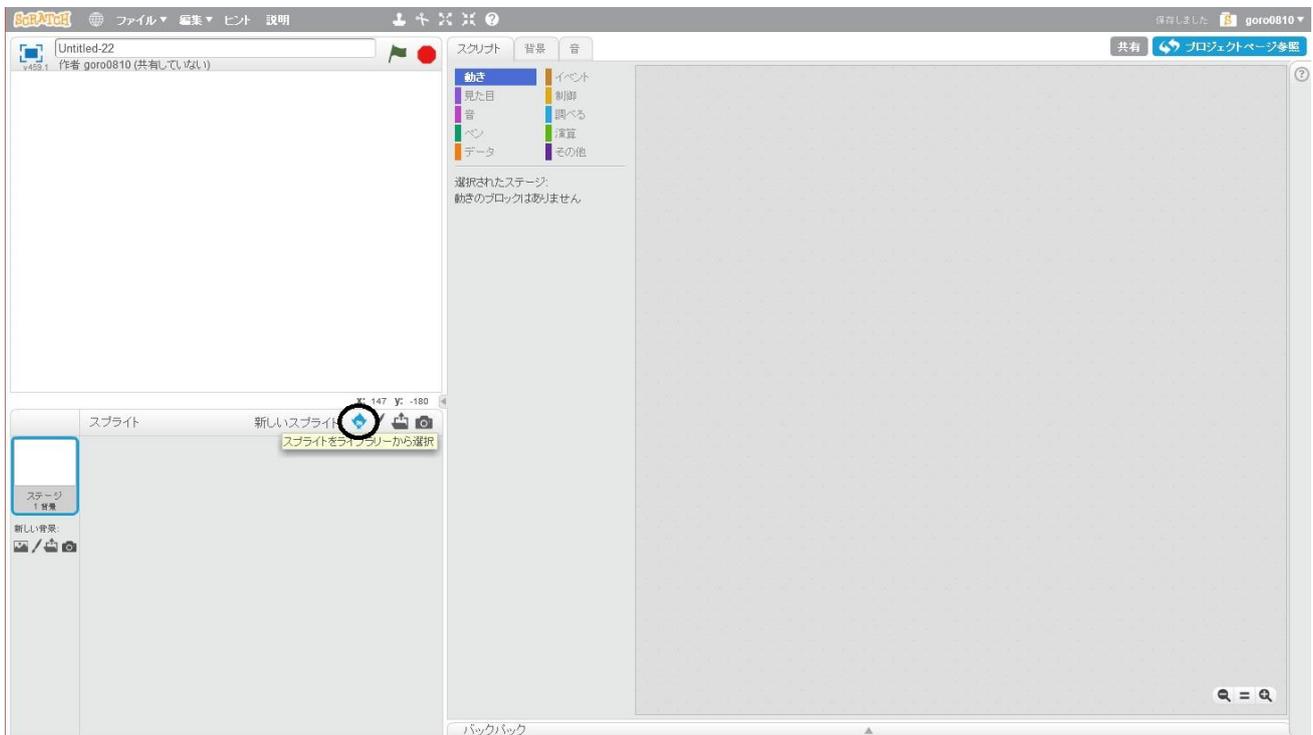
途中で設計図(ゲームの内容)を変えてもいいですし、途中で失敗してもいいです。とにかく上達するには失敗をくり返し、その分、上達していくことです。失敗してすることが、プログラミングでは許されています。



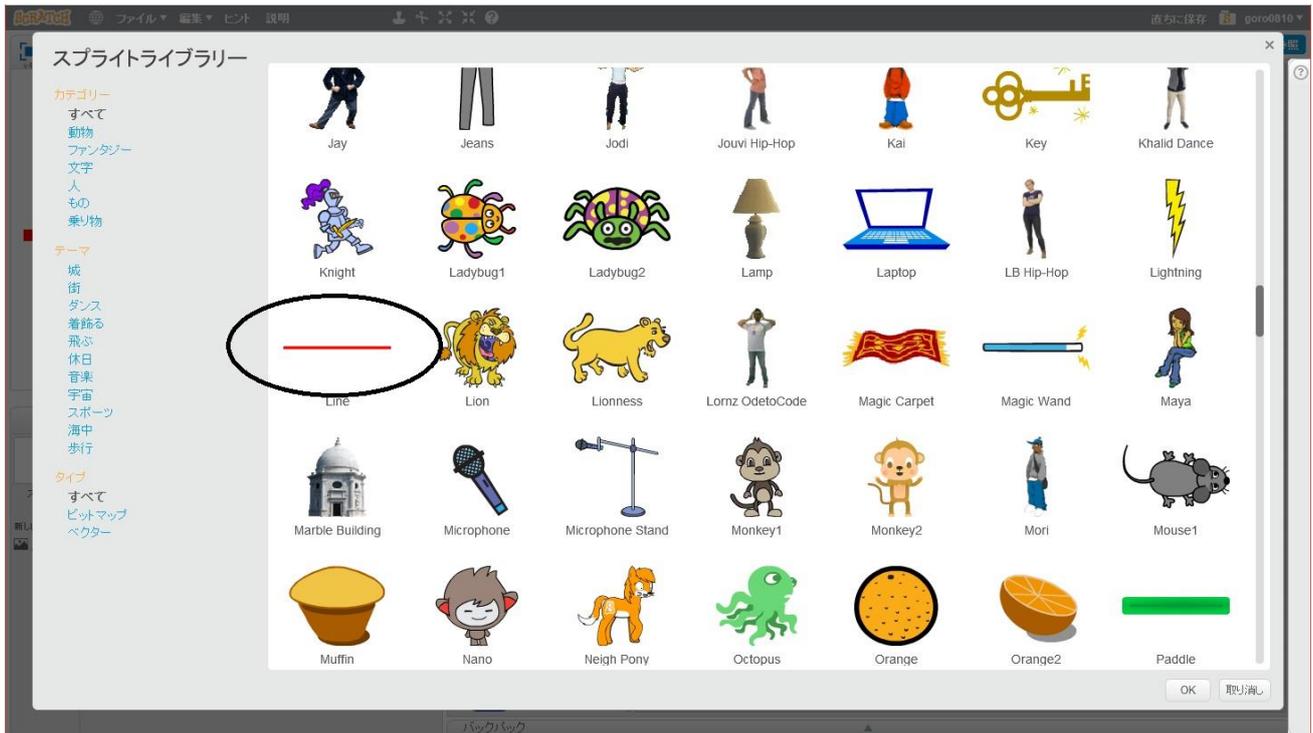
完成イメージをつかんでおきましょう。

ではつくっていきましょう。

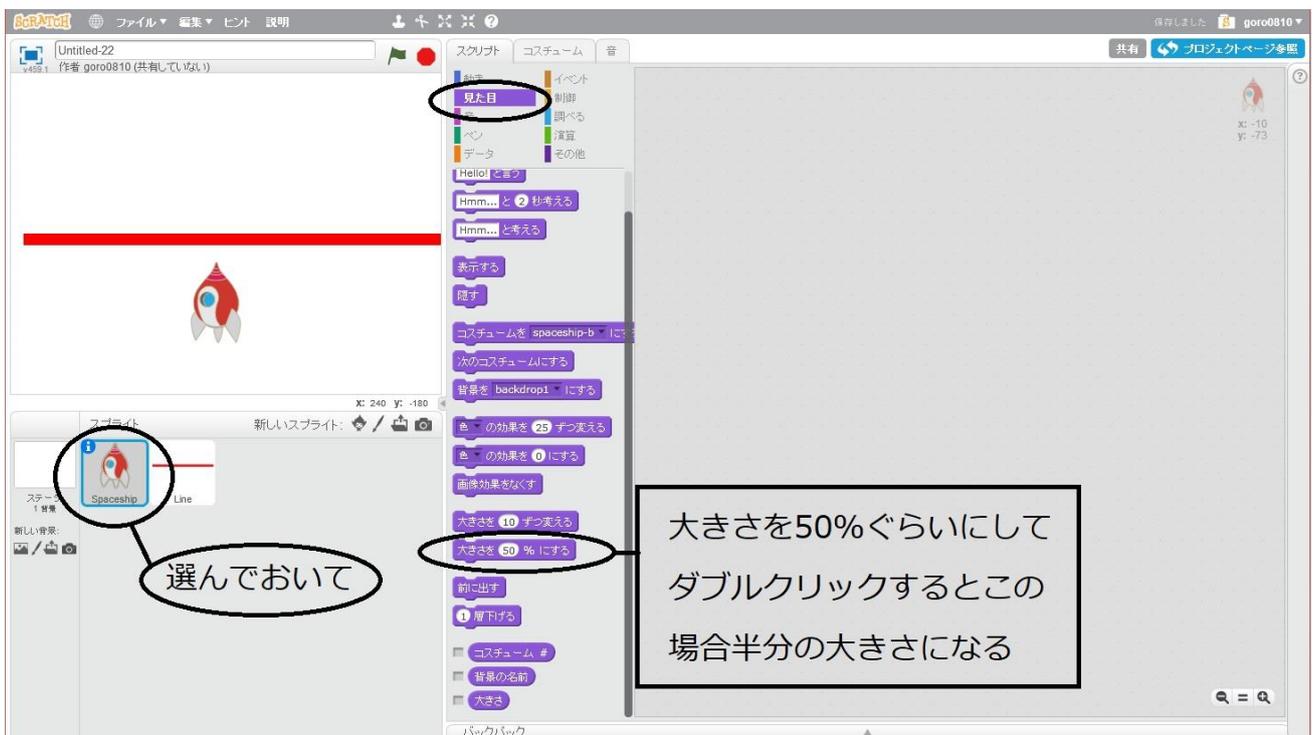
猫のキャラクターはいらないので、削除しましょう。
 猫をクリックして、コスチュームで猫を選ぶと×印で削除していきましょう。消えないときは、猫の上で右クリック→削除でも消せることができます。



新しいスプライトの追加で Spaceship(自機)を選びます。

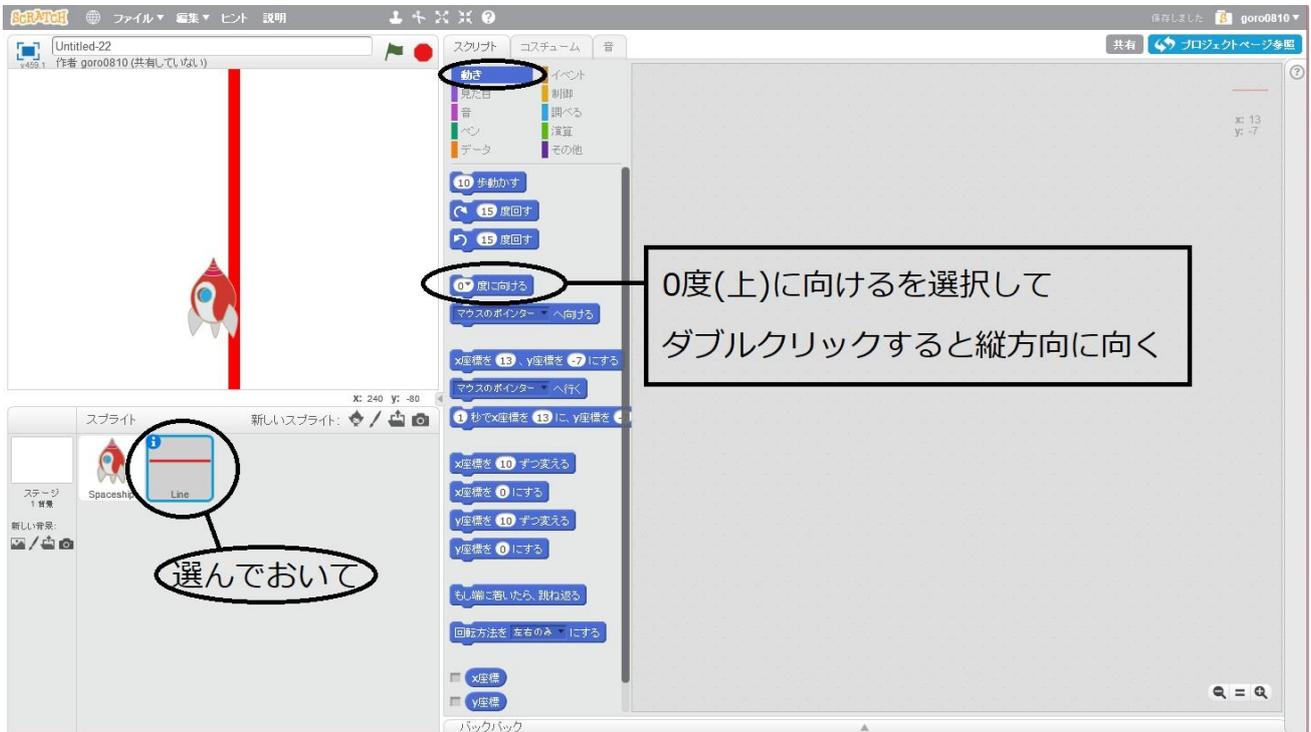


同じようにしてミサイルのもと Line を選択します。



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。

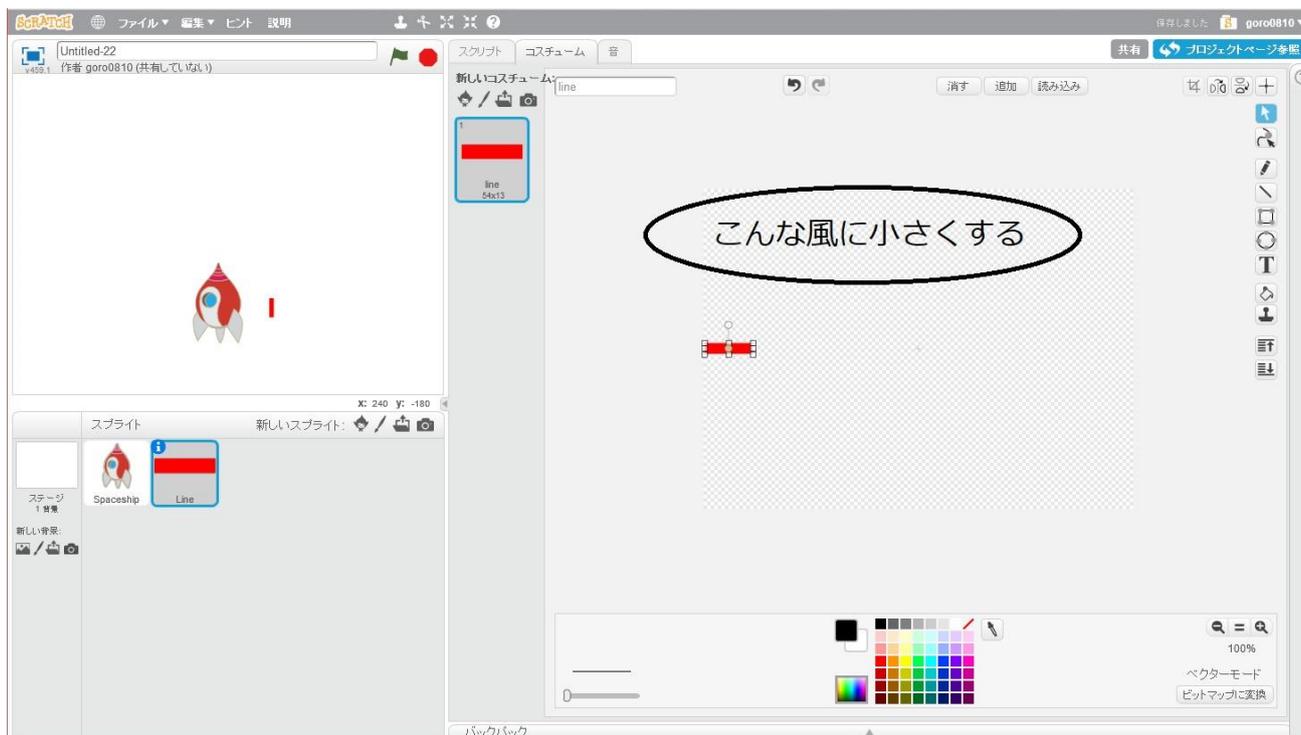
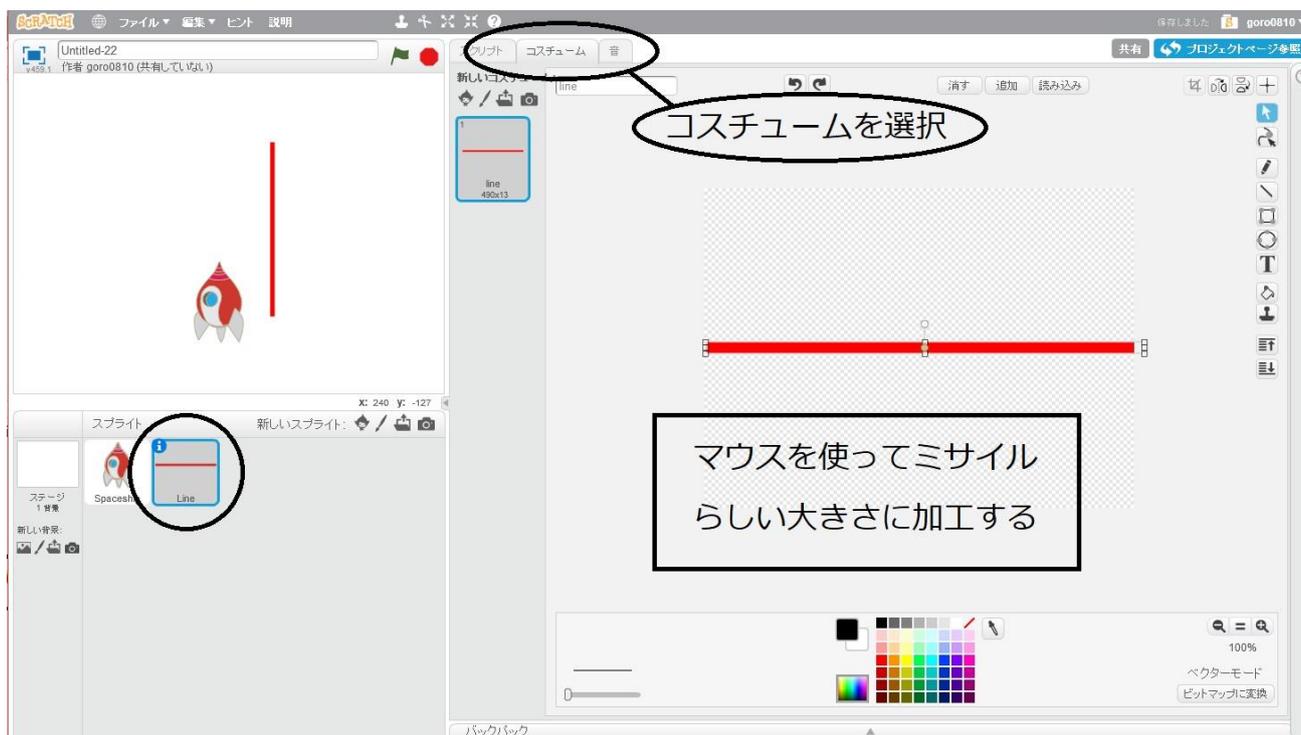
ミサイルが画面上で横になっているので、Line を選んでおいて、「動き」から 0 度(上)に向けるを選択する。そして青色の部分ダブルクリックすると Line が図のように縦に向きます。



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。



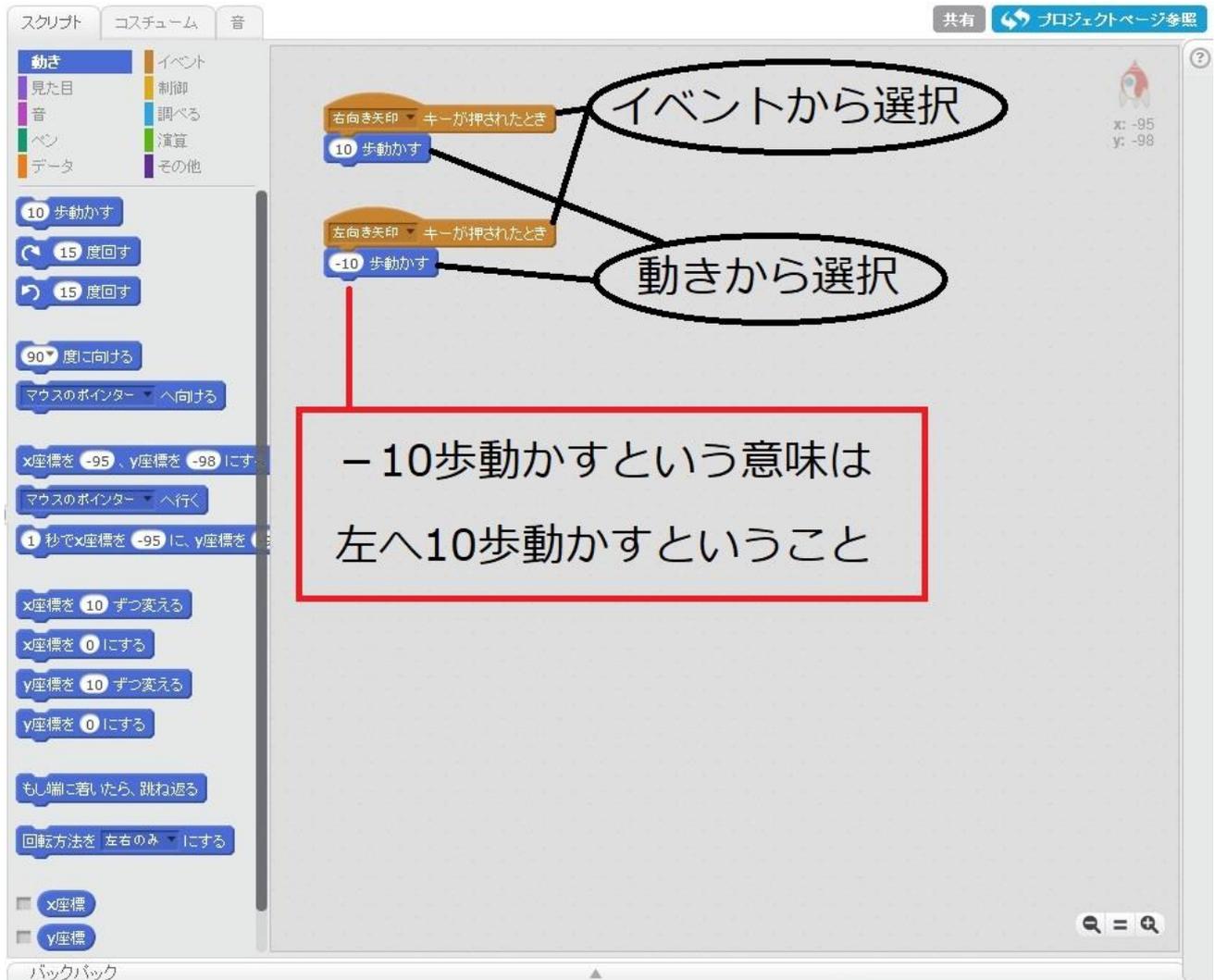
出来ましたか？

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。

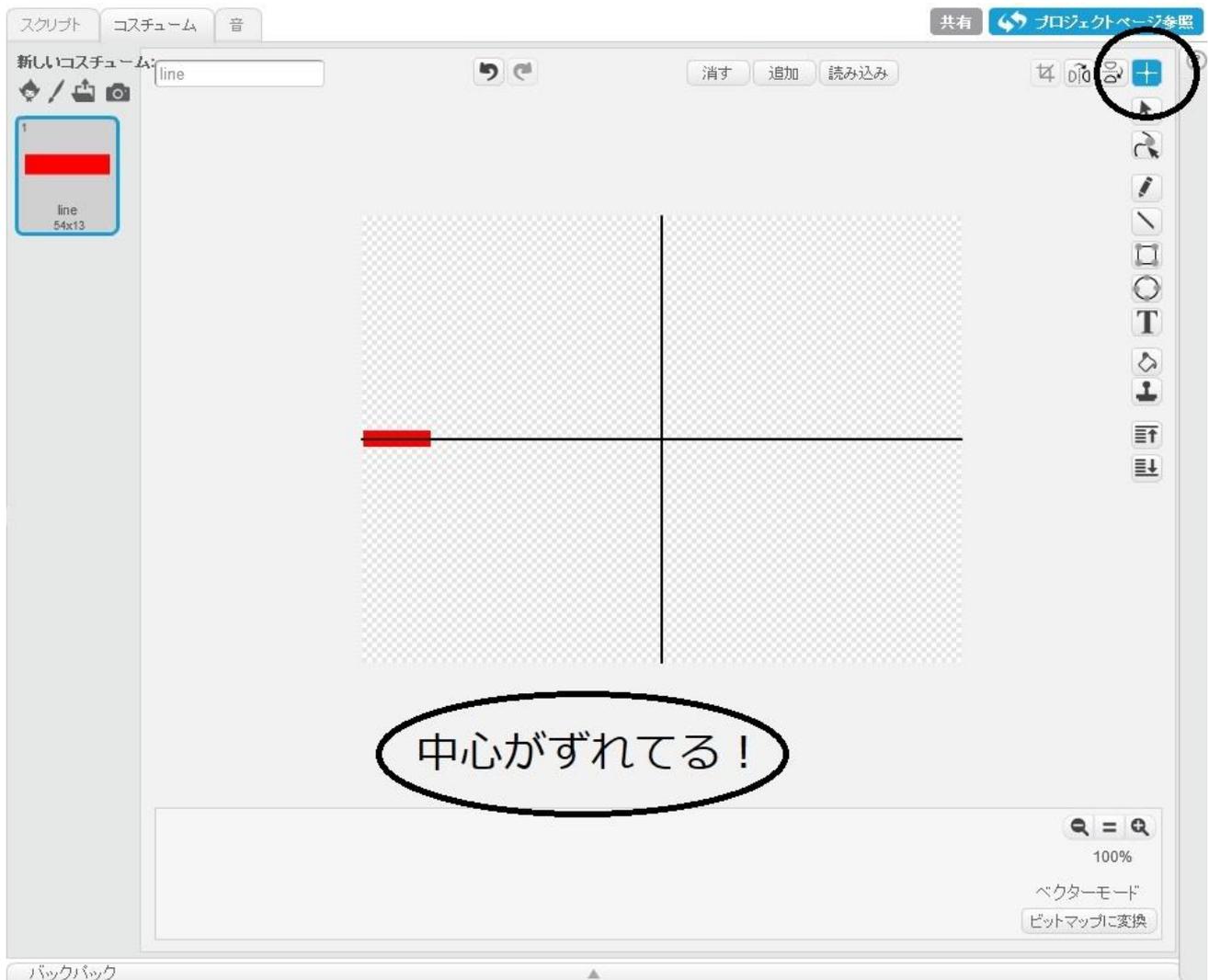


自機を高速で動かしたい場合は、10、-10 という数字を 20、-20 とすれば早くなります。もちろん 100、-100 ならもっと速く、反対に 3、-3 なら遅くなります。

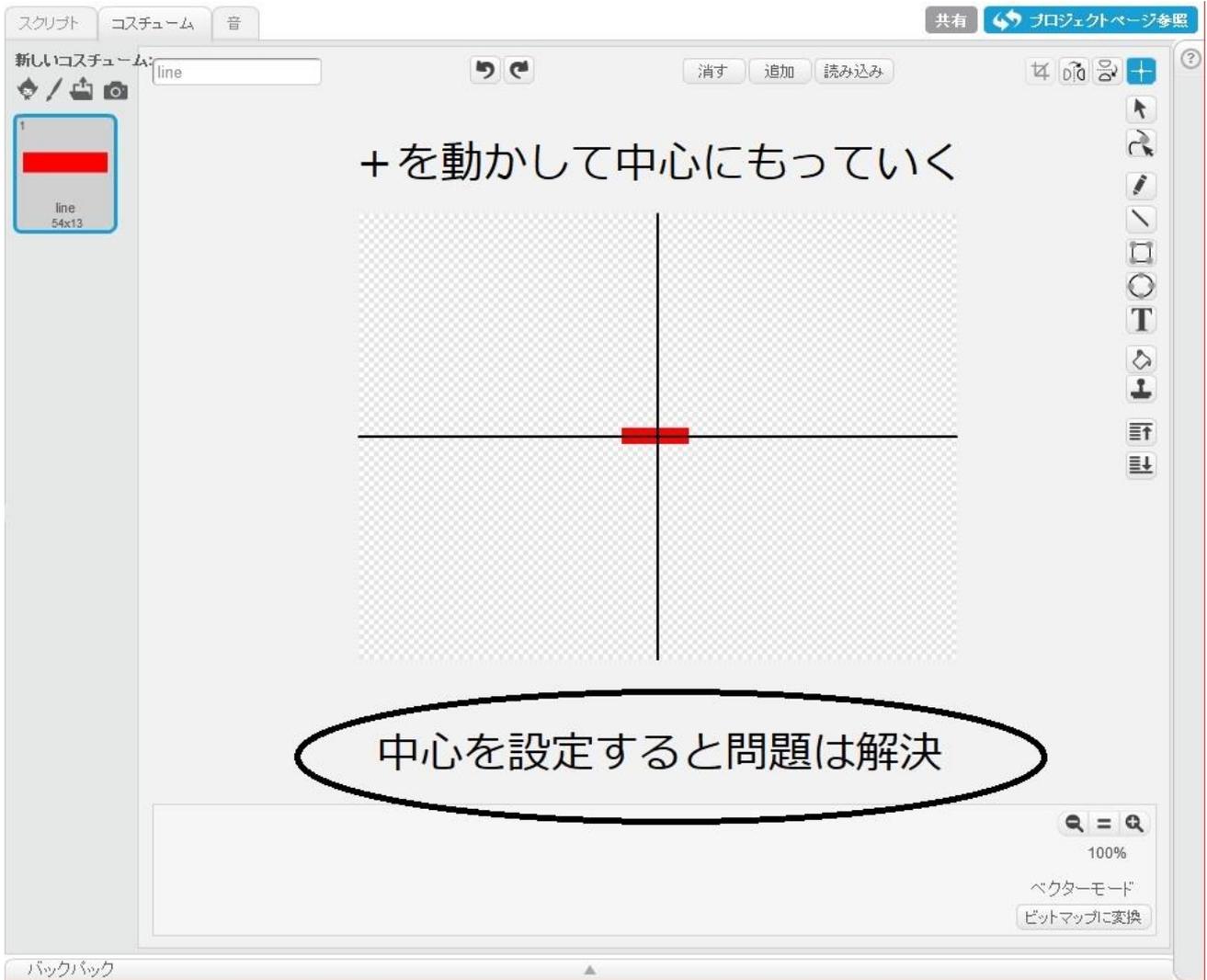
次にミサイルですが、ミサイルの発射ボタンを決めましょう。
今回はミサイルの発射ボタンはスペースキーということにします。
このキーの設定は後でも変更可能です。しかし、次のようにプログラムを組んで、スペースキーを押すと、ミサイルが自機の下にくるという問題が発覚しました。



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。





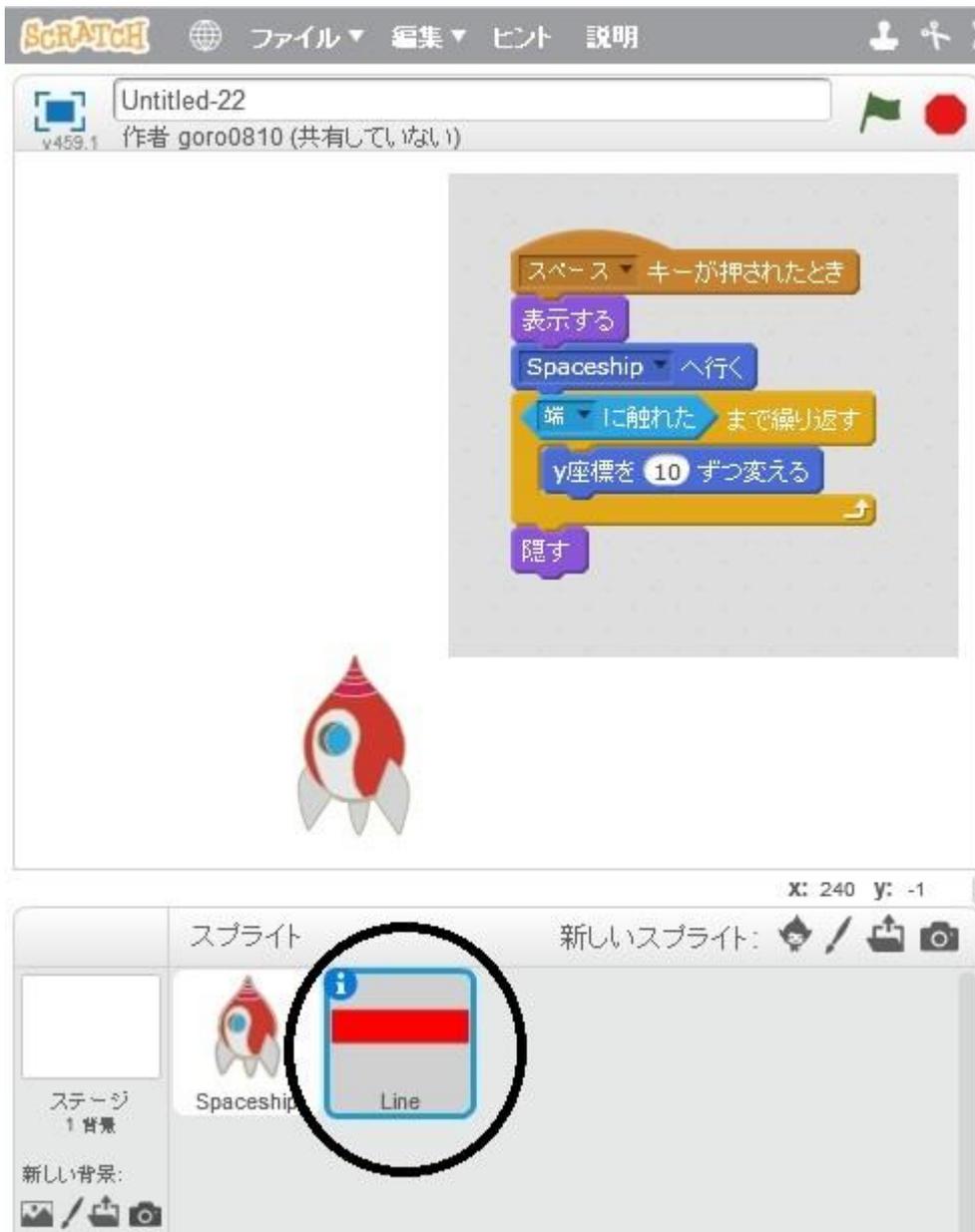
ずれは解消されました。

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。

まずは背景の変更を行います。
新しい背景を追加しましょう。



敵キャラや敵のミサイルを追加しましょう。
新しいスプライトの追加から行います。



敵キャラと敵のミサイルを追加しました。

The screenshot shows the Scratch programming environment. The main stage displays a 3D perspective view of a blue corridor with purple grid lines. In the center, a purple robot with green arms and legs is positioned. To its left, a yellow lightning bolt is visible. In the foreground, a red and white rocket is flying towards the robot. The top of the window shows the Scratch logo and menu options: ファイル (File), 編集 (Edit), ヒント (Help), 説明 (About). The top right corner has icons for saving, undo, redo, and help. The top left corner shows the project title 'シューティングゲーム2' (Shooting Game 2) and the author 'goro0810 (共有していません)' (goro0810 (Not shared)).

The right side of the window is the script editor, showing a list of blocks on the left and a stack of blocks on the right. The block list includes: 動き (Motion), イベント (Events), 見た目 (Looks), 制御 (Control), 音 (Sound), 調べる (Sensing), 演算 (Operators), データ (Data), and その他 (Other). The script editor contains the following blocks:

- 10 歩動かす (Move 10 steps)
- 15 度回す (Turn 15 degrees)
- 15 度回す (Turn 15 degrees)
- 90 度に向ける (Turn 90 degrees)
- マウスのポインター へ向ける (Point towards mouse pointer)
- x座標を 86、y座標を -27 にする (Set x coordinate to 86, y coordinate to -27)
- マウスのポインター へ行く (Go to mouse pointer)
- 1 秒でx座標を 86 に、y座標を -2 にする (Move to x coordinate 86, y coordinate -2 in 1 second)
- x座標を 10 ずつ変える (Change x coordinate by 10)
- x座標を 0 にする (Set x coordinate to 0)
- y座標を 10 ずつ変える (Change y coordinate by 10)
- y座標を 0 にする (Set y coordinate to 0)
- もし端に着いたら、跳ね返る (If touching edge, bounce)
- 回転方法を 左右のみ にする (Set rotation method to only left and right)

The bottom left of the window shows the sprite editor, with a list of sprites: ステージ 2 背景 (Stage 2 background), Spaceship, Lightning, Robot1, and Lightning. The Robot1 sprite is circled in black. Below the sprite editor, the text '敵キャラと敵のミサイルを追加しました。' (Added enemy character and enemy missile.) is displayed.

自機、敵キャラ、敵のミサイルなど大きさを変更しておきましょう。
自機は 60% ぐらいでしょうか。

The screenshot shows the Scratch IDE interface. The main stage displays a blue neon tunnel with a rocket ship (self) and a purple robot (enemy) in the distance. A lightning bolt is also visible. The script area on the right contains several blocks, with the 'Set size to 20%' block circled in black. The sprite area at the bottom shows the 'Lightning' sprite selected.

敵キャラは大きさを50%
ぐらい、ミサイルは20%
ぐらいにしよう。

まず、敵キャラは指定した範囲で、ランダム(適当)に出現させたいので、乱数を使って出現させます。敵キャラに対して、以下のようにプログラムすると、旗のボタンを押すたびに、指定した範囲に適当に現れます。

Scratchのスクリーンショット。タイトル「シューティングゲーム2」のプロジェクトが示されています。ステージには、x座標が-210から210の間、y座標が-30から140の間に敵キャラ（ロボット）が出現するよう設定されています。右側のメニューで「スクリプト」が選択されています。また、下のメニューで「Robot1」が選択されています。

旗をクリックすると指定された範囲に敵が現れます

敵に動きを加えます。

旗のボタンを押すと、指定した範囲に適当に出現させ、そこから端に触れるまで、下方向に1ずつ動かしていきます。その間に自機のみ사일に触れたら隠すようにプログラムします。

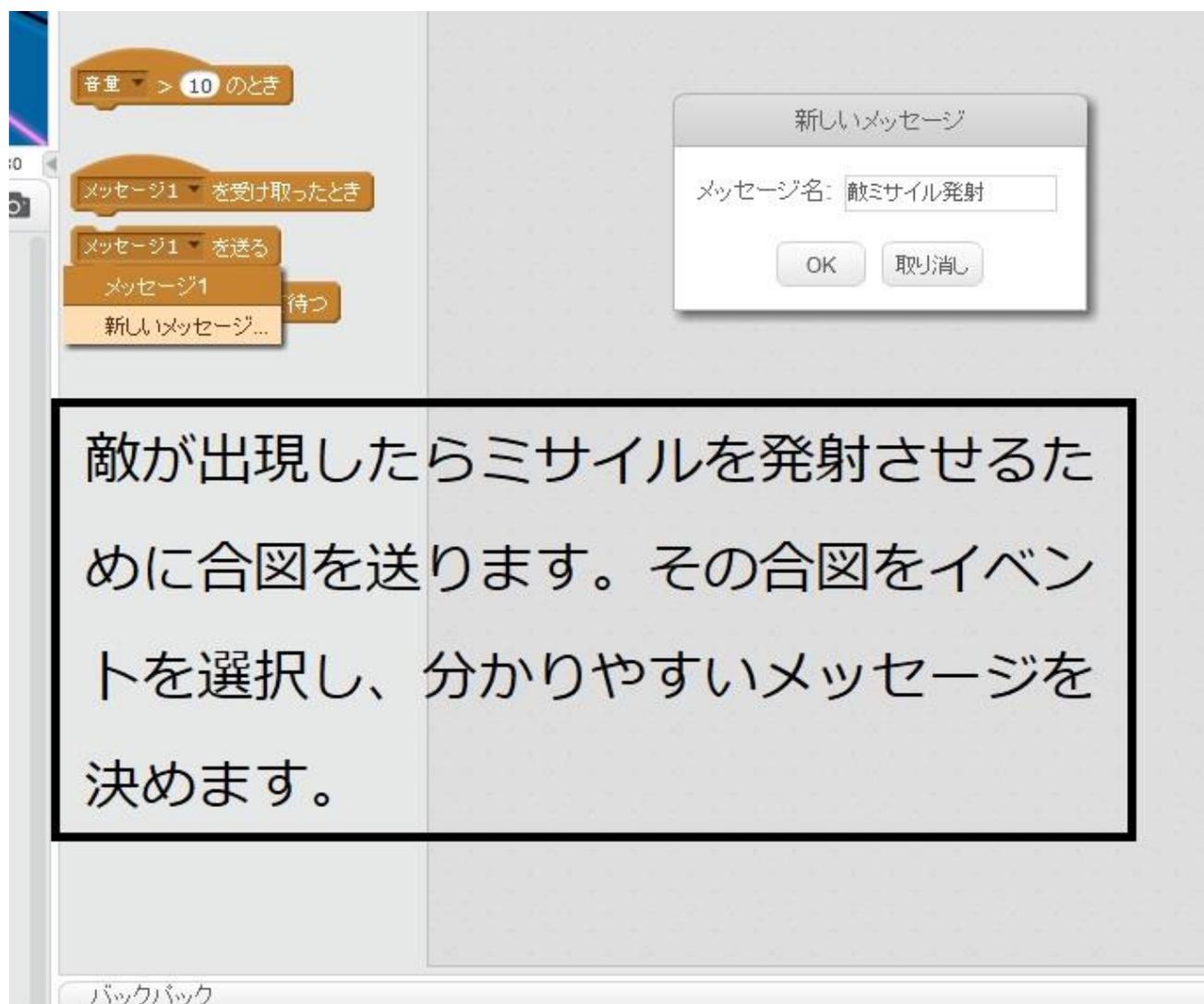


The image shows a Scratch script for a character named 'Robot1'. The script starts with a 'when clicked' event. It then sets the x-coordinate to a random number between -210 and 210, and the y-coordinate to a random number between -30 and 140. The character is then shown. A 'forever' loop follows, where the y-coordinate is decreased by 1. A 'when touching line' condition is used to trigger the 'hide' action.

敵を出現させた後は、下に1ずつ動かしていきます。動いている間で自機のみ사일に当たったら隠すようにプログラミングします。

敵も攻撃するようにします。

敵が出現したら、敵にミサイル発射をさせたいので、敵が出現したら、敵ミサイル発射という合図を敵のミサイルに送ります。そのためにイベントからわかりやすいメッセージをつくっておきます。



ここでは、イベントのメッセージを敵ミサイル発射としていますが、分かりやすければ何でも OK です。

つくったイベントを敵キャラのプログラムに組み込みます。



The image shows a Scratch script for a character named "Robot1". The script starts with a "when clicked" event block. This is followed by a "set x coordinate to random number between -210 and 210, and y coordinate to random number between -30 and 140" block. Then, there is a "show" block, followed by a "send enemy missile launch" block. A "forever" loop follows, containing a "decrease y coordinate by 1" block, a "when touches line" block, and a "hide" block. A black line connects the "send enemy missile launch" block to a text box below.

Robot1

がクリックされたとき

x座標を -210 から 210 までの乱数、y座標を -30 から 140 までの乱数 にする

表示する

敵ミサイル発射 を送る

ずっと

y座標を -1 ずつ変える

もし Line に触れた なら

隠す

ミサイル発射のイベントを追加します。

敵のミサイル側では、敵キャラから敵ミサイル発射の合図を受け取ると、敵のミサイルは敵キャラのところにいて表示され、端に触れるまで、下に6ずつ動いていくようにプログラムします。端に触れたら敵のミサイルは隠れます。

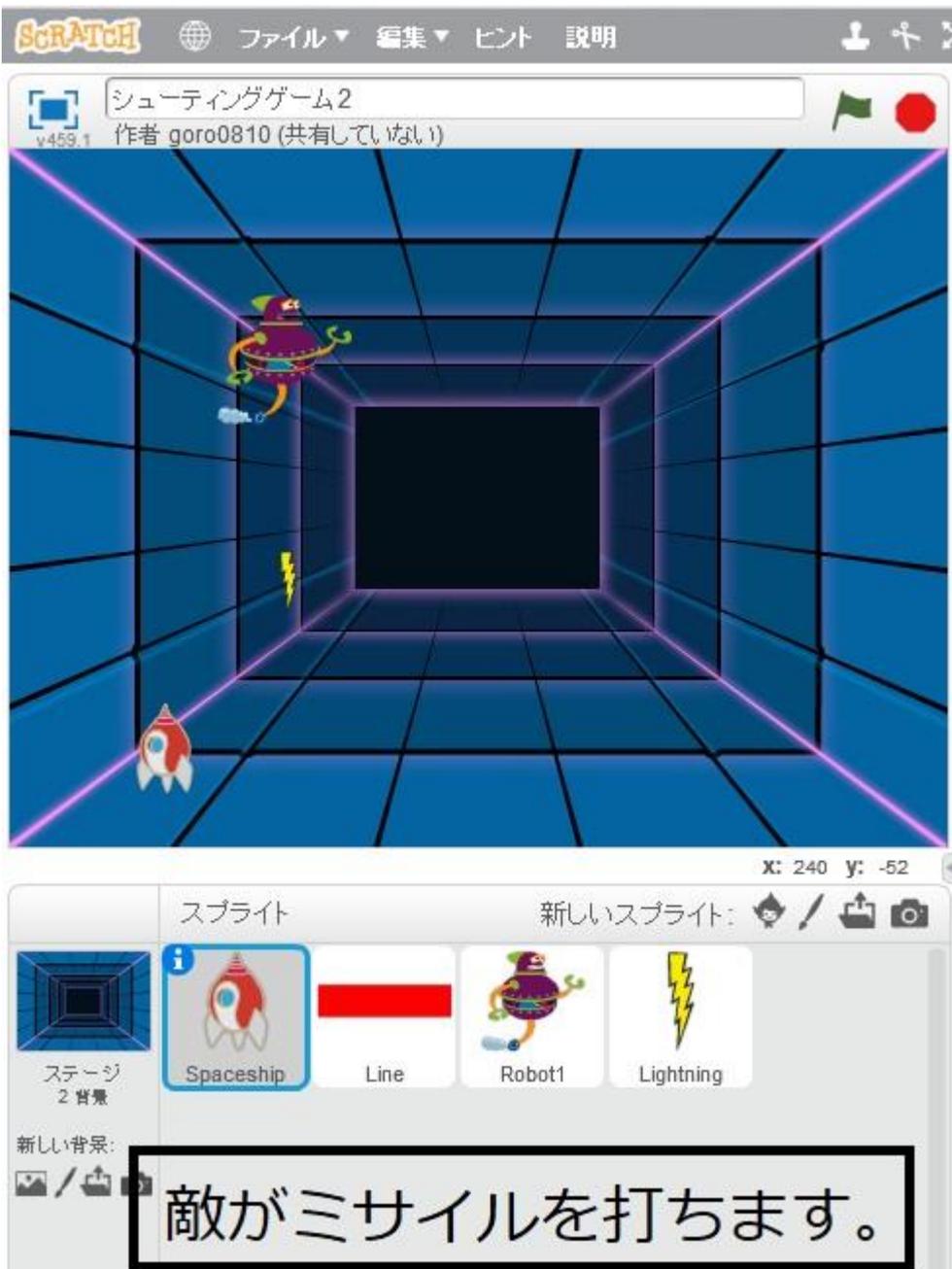


The image shows a Scratch script for an enemy missile. The script starts with an event block '敵ミサイル発射' (Enemy missile launch) triggered 'を受け取ったとき' (when received). The actions are: 'Robot1' (selected) 'へ行く' (go to), '表示する' (show), a loop '端' (edge) 'に触れた' (when touched) 'まで繰り返す' (repeat until), 'y座標を -6 ずつ変える' (change y coordinate by -6), and finally '隠す' (hide). To the right is a 'Lightning' sprite icon.

敵のミサイル側では敵ミサイル発射というイベントが発生したら、ミサイルを発射させるようにプログラムします

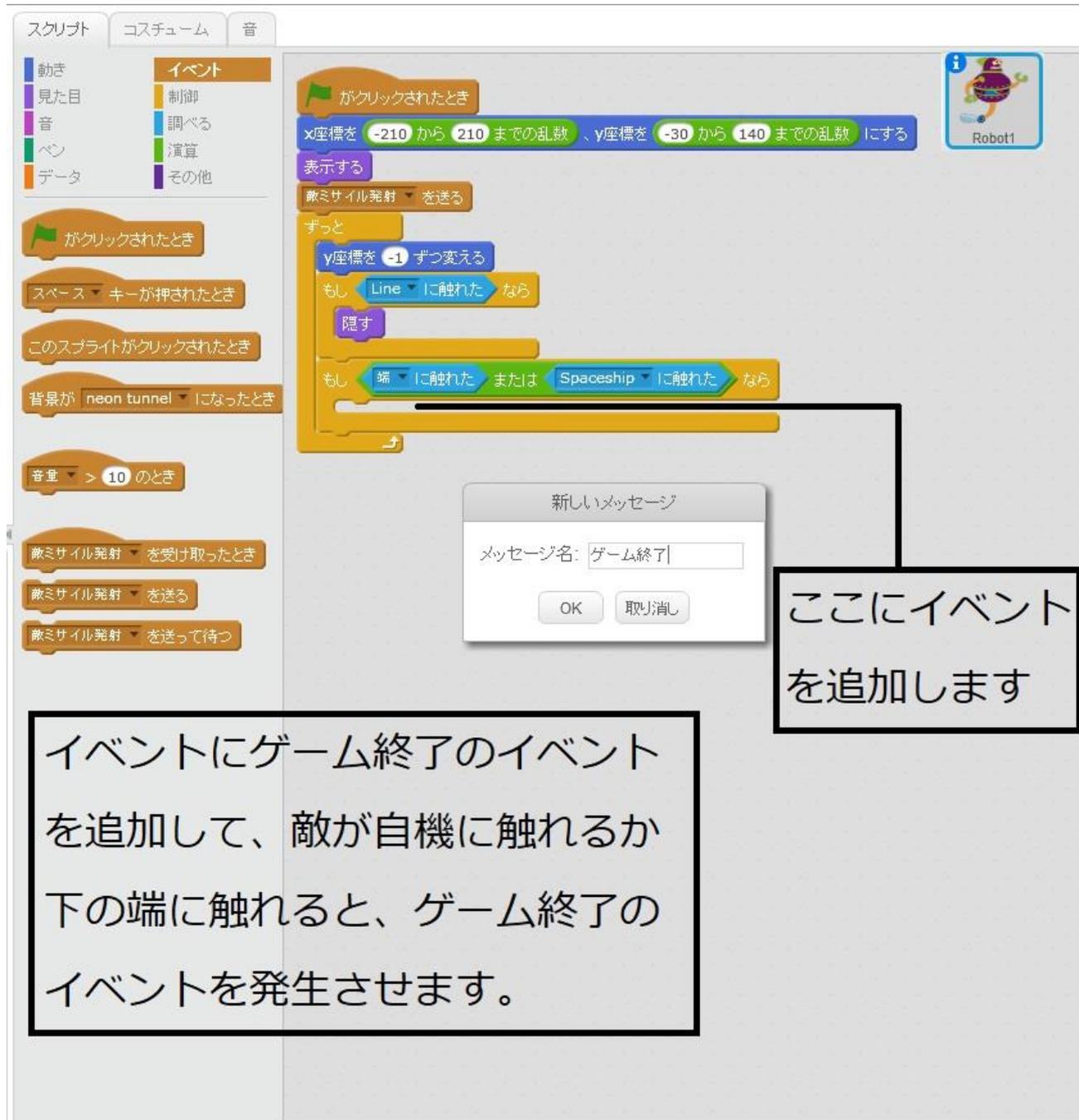
上ではイベント発生後、敵ミサイルを敵のところにもっていき、表示させて端に触れるまで下に6ずつ動かしていき、端に触れたら隠すようにプログラムしています。

これで、敵キャラが現れたら敵からミサイルが発射されます。



次は敵キャラが下に進んでいき、その途中で、自機に触れたり、下の端に触れたりするとゲームオーバーとします。そのため、敵キャラが端に触れるか、自機に触れるかしたら、ゲーム終了の合図を送るようにプログラムします。

制御から、もし○または○に触れたならを選択して、端に触れた、Spaceship にふれたならをはめ込みます。



Scratchのスクリーンショット。左側のパネルには「イベント」カテゴリが選択されており、「がクリックされたとき」のイベントブロックが追加されている。このイベントブロックには、「x座標を -210 から 210 までの乱数、y座標を -30 から 140 までの乱数 にする」ブロック、「表示する」ブロック、「敵ミサイル発射」を送るブロック、「ずっと」ループブロック、「y座標を -1 ずつ変える」ブロック、「もし Line に触れた なら」ブロック、「隠す」ブロック、「もし 端 に触れた または Spaceship に触れた なら」ブロックが追加されている。また、「敵ミサイル発射」を受け取ったとき、「敵ミサイル発射」を送る、「敵ミサイル発射」を送って待つなどのブロックもパネルには表示されている。

中央には「新しいメッセージ」ダイアログボックスが開いており、「メッセージ名: ゲーム終了」と入力されている。OKと取り消しボタンがある。

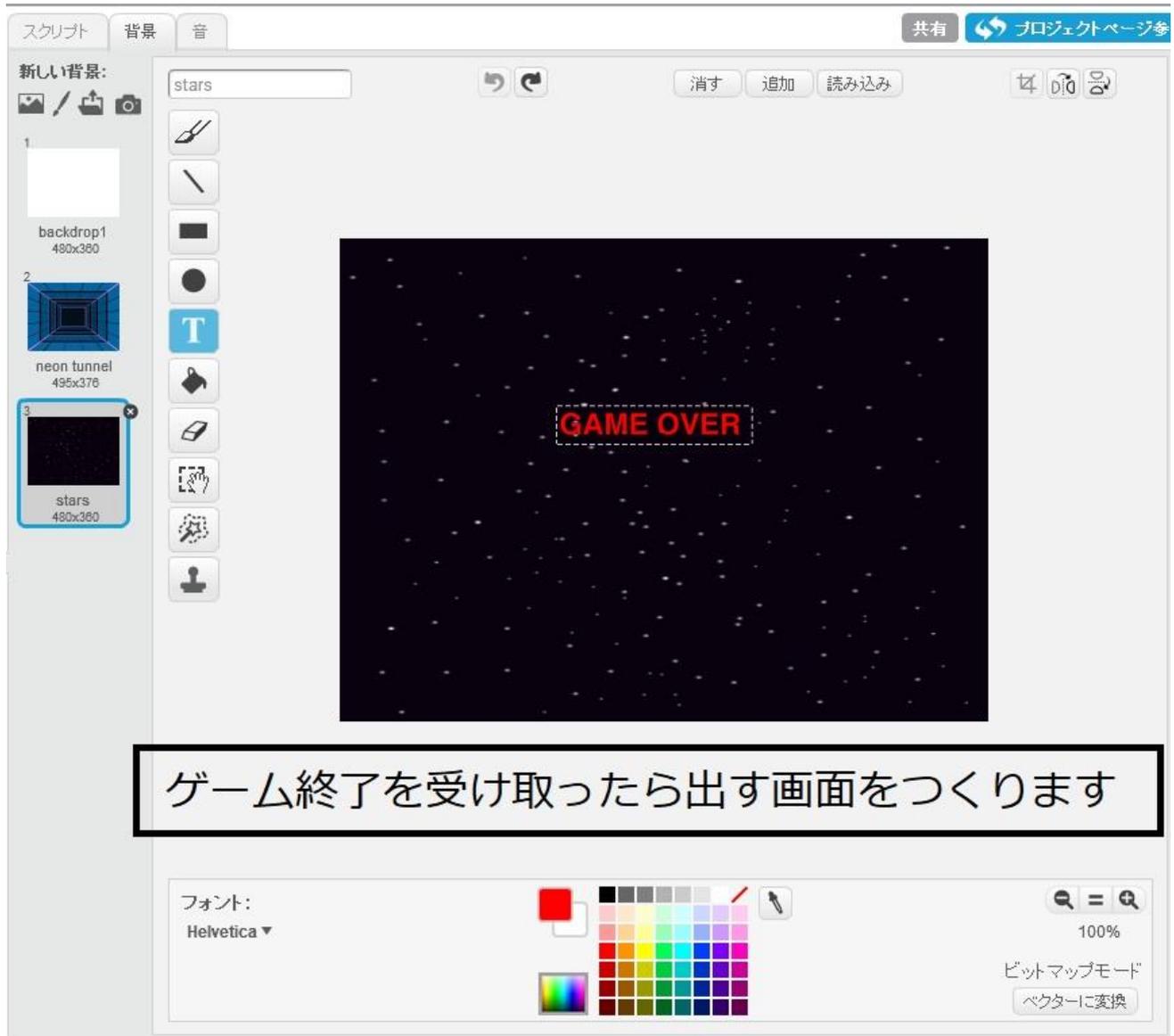
右側の黒い枠には「ここにイベントを追加します」というテキストがある。

左下の黒い枠には「イベントにゲーム終了のイベントを追加して、敵が自機に触れるか下の端に触れると、ゲーム終了のイベントを発生させます。」というテキストがある。

イベントでゲーム終了というイベントをつくっておきましょう。つくり方はP25に載っています。

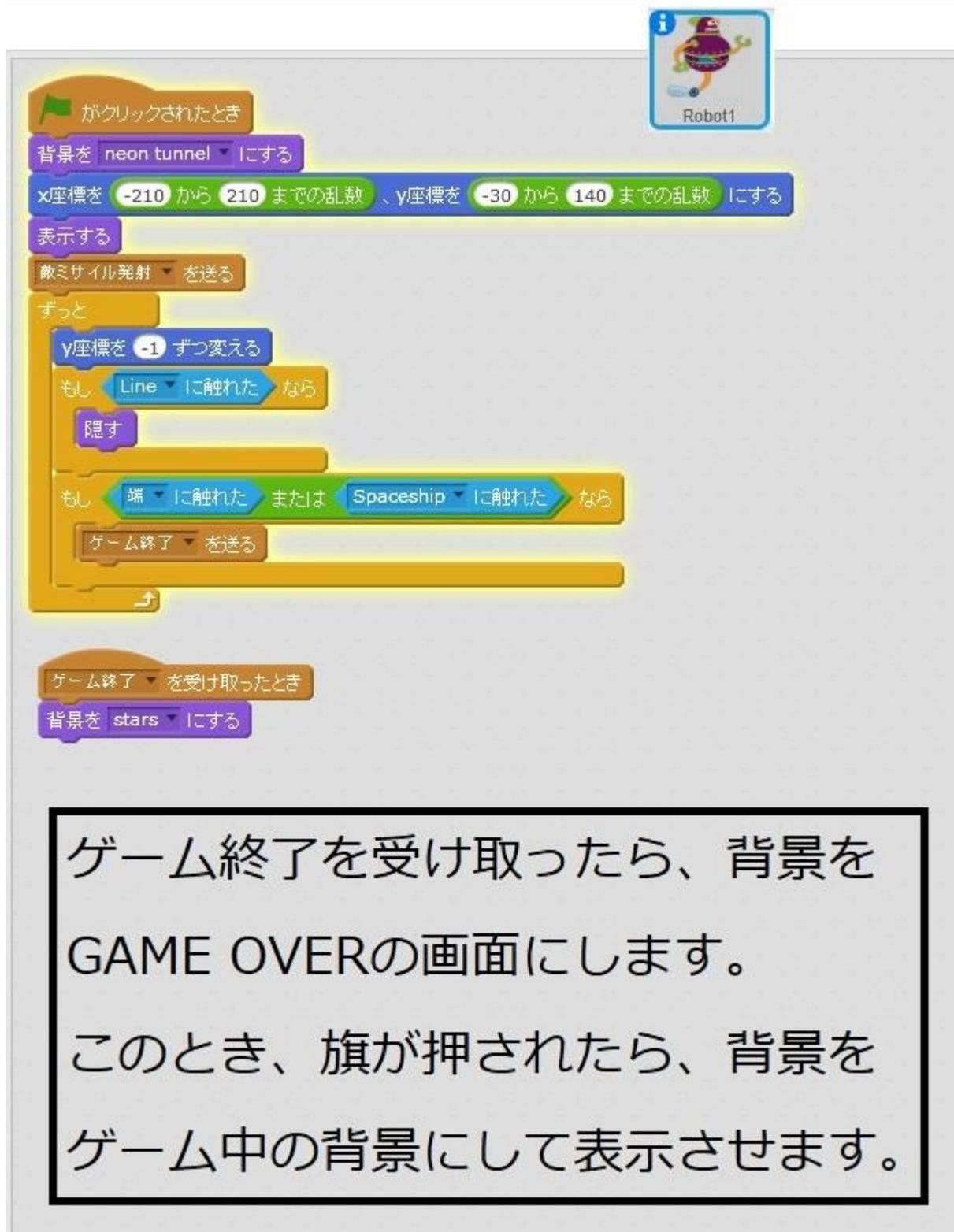
ゲーム終了のイベントが発生したときに出すものをつくります。

今回は新しい背景を追加して、ゲーム終了の画面を出すようにします。その画面をつくっておきます。



stars をゲームオーバーの画面とします。

敵キャラが端に触れるか、自機に触れるかすると、ゲーム終了のイベントが発生するようにプログラムしました。ゲーム終了のイベントが発生するとゲームオーバーの画面に切り替えるようにプログラムしました。



The image shows a Scratch script for a character named 'Robot1'. The script is divided into two main event-driven sections. The first section is triggered by a 'when clicked' event. It sets the background to 'neon tunnel', randomizes the x and y coordinates to specific ranges, displays the character, and sends a 'missile launch' message. A 'forever' loop follows, where the y-coordinate is decremented by 1. If it touches a 'line', the character is hidden. If it touches either the 'edge' or a 'spaceship', a 'game over' message is sent. The second section is triggered by receiving a 'game over' message, which sets the background to 'stars'. A large text box at the bottom of the script area contains Japanese instructions: 'ゲーム終了を受け取ったら、背景を GAME OVERの画面にします。' and 'このとき、旗が押されたら、背景を ゲーム中の背景にして表示させます。'

上のプログラムでは、ゲーム終了を受け取ったなら、背景を stars(ゲームオーバー)にするとしています。

自機のみサイルの当たったら敵キャラの動きのスク립トを停止させます。



The image shows a Scratch script for a character named "Robot1". The script is divided into two main sections. The first section starts with a "when clicked" event, followed by setting the background to "neon tunnel", randomizing the x and y coordinates, and displaying the character. It then sends a "enemy missile launch" message. A "forever" loop follows, where the y-coordinate is decremented by 1. A "when touching line" condition is used to hide the character and stop the script. Another "when touching end or spaceship" condition is used to send a "game over" message. The second section starts with a "when game over message received" event, setting the background to "stars".

敵が自機のみサイルに当たったら隠れますが、隠れた状態でプログラムは動きます。ですから、敵が自機のみサイルに当たったら、敵のプログラム自体を止めてしまいます。

自機のみ사일も敵のみ사일か、敵キャラに触れたなら隠して、ゲーム終了の合図を送るようにプログラムします。



The image shows a Scratch script for a character named 'Spaceship'. The script includes three event triggers: 'when the right arrow key is pressed', 'when the left arrow key is pressed', and 'when the green flag is clicked'. The first two triggers are connected to 'move 20 steps' and 'move -20 steps' blocks, respectively. The third trigger is connected to a 'show' block, followed by a 'forever' loop containing an 'if' condition. The condition checks if the spaceship touches 'Lightning' or 'Robot1'. If true, the script performs 'hide' and 'send game over message' actions.

自機のプログラムも少し修正
敵のみ사일または敵に当たったら隠して、ゲーム終了のイベントを発生させます。

敵のミサイルに一工夫。今までなら、敵キャラからミサイルは初めの一発だけでしたが、端に触れた後、ミサイル発射の合図を送ることで、何度もミサイルが発射されるように変更します。



The image shows a Scratch script for a lightning bolt object. The script starts with an event block '敵ミサイル発射' (Enemy Missile Launch) triggered 'を受け取ったとき' (when received). The actions are: 'Robot1' (Robot1) 'へ行く' (go to), '表示する' (show), a loop '端' (End) 'に触れた' (when touched) 'まで繰り返す' (repeat until), 'y座標を -6' (y-coordinate by -6) 'ずつ変える' (change), '隠す' (hide), and finally '敵ミサイル発射' (Enemy Missile Launch) 'を送る' (send).

敵のミサイルが壁に触れたら再度ミサイルを発射させるために、ミサイル発射のイベントを発生させます。これで、ミサイルが何度も発射されます。

敵キャラが自機のみ사일に当たったときのイベントをつくります。

The screenshot shows the Scratch programming environment. On the left is a stage with a blue grid background and a rocket ship. On the right is the script editor with tabs for 'Script', 'Costume', and 'Sound'. The 'Script' tab is active, showing a list of event categories: '動き' (Motion), '見た目' (Appearance), '音' (Sound), 'ペン' (Pen), 'データ' (Data), 'イベント' (Event), '制御' (Control), '調べる' (Cycling), '演算' (Math), and 'その他' (Other). Under the 'Event' category, several event blocks are listed: 'がクリックされたとき', 'スペース キーが押されたとき', 'このスプライトがクリックされたとき', '背景が stars になったとき', '音量 ... になったとき', 'やられた を受け取ったとき', 'やられた を送る', and 'やられた を送って待つ'. The 'やられた を受け取ったとき' block is circled in black. Below the stage, there are two sprite icons: 'Robot1' and 'Lightning'. At the bottom, a text box contains the following text:

敵が自機のみ사일に当たってからの敵のみ사일を制御したいので、イベントでやられたというイベントをつくる。

やられたというイベントを自機に触れたときのところに組み込みます。

Scratch script for a game character named Robot1. The script starts with a 'when clicked' event, sets the background to 'neon tunnel', and initializes random coordinates for x and y. It then shows the character and sends a 'missile launch' message. A 'forever' loop follows, where the y-coordinate decreases by 1. If it touches a 'Line', it hides the character, sends a 'hit' message, and stops the script. If it touches the 'end' or 'Spaceship', it sends a 'game over' message. A separate script block shows that when 'game over' is received, the background changes to 'stars'. A callout box highlights the 'hit' message event in the loop.

やられたというイベントを自機の
ミサイルに触れたときの部分には
め込みます。

自機のみ사일に当たったときの敵のみ사일を制御します。



The image shows a Scratch script for an enemy missile. At the top right, there is a 'Lightning' icon with an information symbol. The script consists of several blocks: '敵み사일發射' (Enemy missile launch) when '受け取ったとき' (when clicked), followed by 'Robot1' (Robot1) 'へ行く' (go to), '表示する' (show), a '端' (edge) 'に触れた' (when reached) 'まで繰り返す' (repeat) loop containing 'y座標を -6 ずつ変える' (change y coordinate by -6), '隠す' (hide), and '敵み사일發射' (Enemy missile launch) 'を送る' (send). Below this, a section is circled in black, containing 'やられた' (hit) 'を受け取ったとき' (when clicked), '隠す' (hide), and 'スプライトの他のスクリプト' (other scripts of this sprite) 'を止める' (stop).

敵のみ사일側では、やられたを受け取ったら、敵のみさいを隠し、敵のみさいの發射部分のスクリプトを止めます。これで敵のみさいの動きが止まりました。

ここままで、ひとまず完成です。

次に敵キャラを2機にしましょう。敵キャラと敵のミサイルの複製をつくりま



x: 240 y: -180



複製をつくって難易度をあげま
しょう。

複製をつくるときの注意点

The screenshot shows the Scratch interface. On the left is a stage with a blue grid background, a purple robot, and a red rocket. On the right is the script editor. The 'Script' tab is selected, showing a list of event categories: 動き (Motion), 見た目 (Looks), 音 (Sound), ペン (Pen), データ (Data), イベント (Events), 制御 (Control), 調べる (Operators), 演算 (Math), and その他 (Others). The 'Events' category is expanded, showing several event blocks. A black circle highlights the 'やられた' (When hit) block. Below the stage is a '新しいスプライト' (New Sprites) panel with 'Robot1', 'Lightning', and 'Robot2' options. A black box at the bottom contains the following text:

複製をつくるときは複製分の
やられたと敵ミサイル発射が
必要になります。

難易度変更 例 1



Scratch script for a game character named Robot2. The script starts with a "when clicked" event, sets the background to "neon tunnel", and initializes x and y coordinates with random numbers. It then enters a loop where the character moves down by 1 unit. If it touches a "Line", it hides, sends a "hit" message, and stops the script. If it touches the "end" or "Spaceship", it sends a "game over" message. A separate script for the "game over" message sets the background to "stars".

難易度変更①

難易度を変えるときは敵キャラの下への動きを速くしたり遅くしたりして調整します。



The image shows a Scratch script for a lightning bolt enemy. The script is as follows:

- 敵ミサイル発射2 を受け取ったとき
- Robot2 へ行く
- 表示する
- 端 に触れた まで繰り返す
- y座標を -6 ずつ変える (circled in red)
- 隠す
- 敵ミサイル発射2 を送る

Below this is another script:

- やられた2 を受け取ったとき
- 隠す
- スプライトの他のスクリプト を止める

Lightning2

難易度変更②

敵のミサイルの発射速度を速くしたり遅くしたりして調整します。

難易度変更 例 3



The image shows a Scratch script for a game titled "Spaceship". The script is designed to adjust the player's speed based on key presses. It features three main event-driven blocks:

- Right Arrow Key Press:** When the right arrow key is pressed, the player's speed is increased by 20 steps. The number "20" is circled in black.
- Left Arrow Key Press:** When the left arrow key is pressed, the player's speed is decreased by 20 steps. The number "-20" is circled in black.
- Click Event:** When the green flag is clicked, the player's speed is displayed on the stage. A loop block follows, which checks if the player has touched "Lightning" or "Robot1". If either is true, the player's speed is hidden, and a "Game Over" message is sent.

The "Spaceship" icon is visible in the top right corner of the workspace.

難易度変更③

自機の動きを速くしたり遅くしたりして調整します。

ゲームづくりはバランスが大切です。やたらと難しくても面白くないですし、やさしすぎても面白くありません。ですから、つくったゲームはたくさんの人にしてもらって難易度のバランスをとってください。難易度の変え方には例の他にも敵を増やすなどありますので挑戦してみてくださいね。

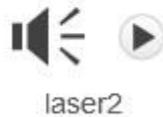
プログラムミスを確認

敵キャラの出現する横方向の(x 軸)範囲を-210~210 とした場合、まれに敵キャラは左右の端に触れて、すぐにゲームオーバーとなります。したがって、敵キャラの横方向(x 軸)の出現範囲を-205~205 程度でプログラミングするといいいでしょう。あと、敵キャラの出現する縦方向(y 軸)も一番下が-10~0 程度がよいかもしれません。その辺はゲームしながら探ってみてください。

最後にゲームをやっている物足りなさを感じませんか？
それはゲームの音がないからです。音を使いしていきましょう。
音を追加するスプライトを選択して、音を追加していきましょう。



自機のミサイルの 発射の音



敵がやられたとき の音

laser1 は自機のミサイルに追加、
laser2 は敵キャラに追加します。敵キャラが複数の場合は全部に音を追加します。
もちろん作成時にはじめから、音を入れてつくってもいいですよ。

最終プログラムは次のページから載せています。
自機最終プログラム



自機ミサイル最終プログラム(音あり)



自機ミサイルの音の追加前のプログラム(音なし)



敵キャラ最終プログラム(音あり)

Robot1

がクリックされたとき

背景を neon tunnel にする

x座標を -205 から 205 までの乱数、y座標を -10 から 140 までの乱数 にする

表示する

敵ミサイル発射 を送る

ずっと

y座標を -1 ずつ変える

もし Line に触れた なら

laser2 の音を鳴らす

隠す

やられた を送る

このスクリプト を止める

もし 端 に触れた または Spaceship に触れた なら

ゲーム終了 を送る

ゲーム終了 を受け取ったとき

隠す

背景を stars にする

すべて を止める

音を追加しておきました。

敵キャラの音の追加前のプログラム(音なし)



The image shows a Scratch script for a character named "Robot1". The script is divided into two main sections: one triggered by a click and another by a "Game Over" message.

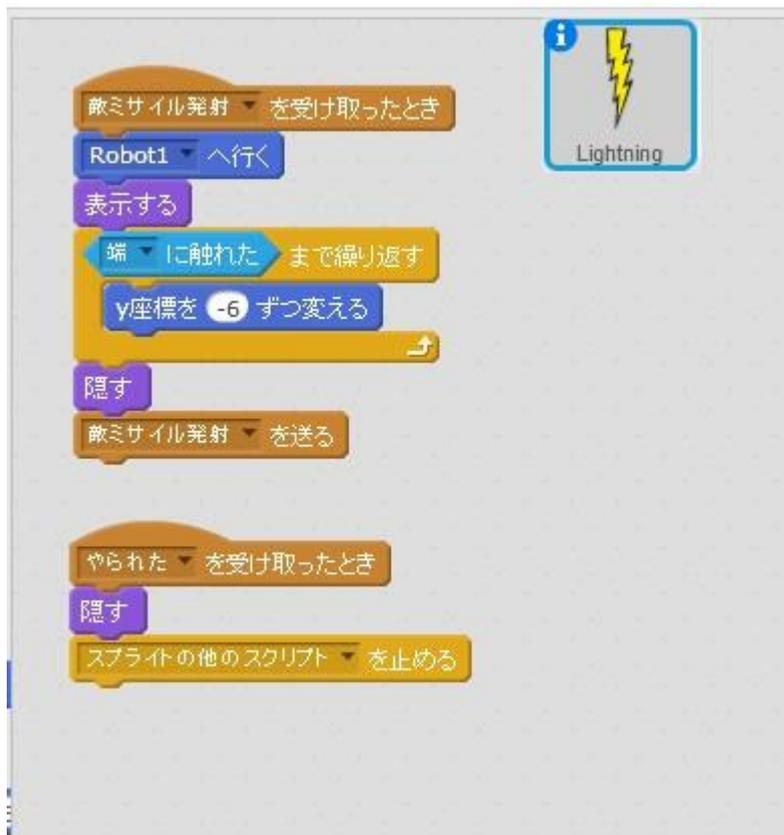
Click Event:

- Background set to "neon tunnel".
- X and Y coordinates are set to random values between -205 and 205, and -10 and 140, respectively.
- The character is shown.
- A "Enemy Missile Launch" message is sent.
- A "Forever" loop begins:
 - The Y coordinate is decreased by 1.
 - If the character touches the "Line" object:
 - The character is hidden.
 - A "Hit" message is sent.
 - The script is stopped.
 - If the character touches the "End" or "Spaceship" objects:
 - A "Game Over" message is sent.

Game Over Event:

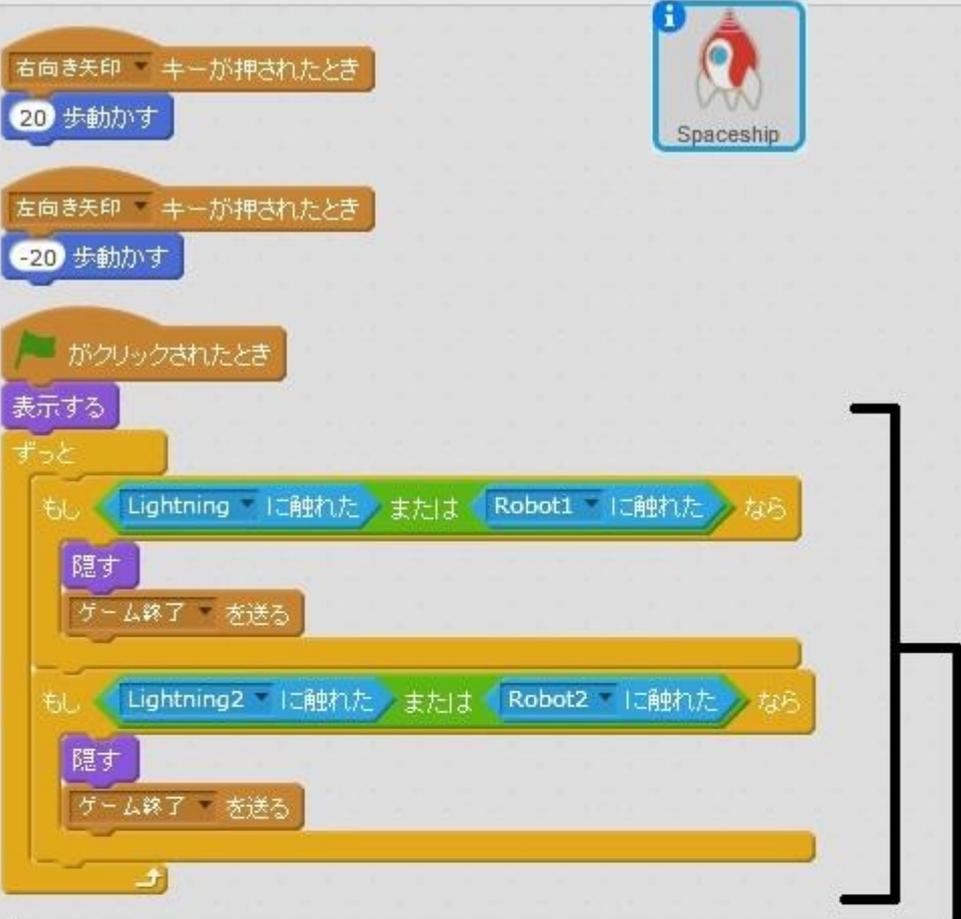
- When the "Game Over" message is received, the background is set to "stars".

敵ミサイル最終プログラム



以上になります。
あとは複製でどうするかはお任せいたします。

敵キャラ 2 機の場合の自機のプログラム



The image shows a Scratch script for a character named "Spaceship". The script is organized into several event-driven blocks:

- Right Arrow Key Pressed:** Move 20 steps.
- Left Arrow Key Pressed:** Move -20 steps.
- Click Flag:** Show the character.
- Forever Loop:**
 - Always:** A loop that runs continuously.
 - Collision Check:** "If Lightning touches Robot1 or Robot2, then...".
 - Actions:** Hide the character and send a "Game Over" message.
- Collision Check:** "If Lightning2 touches Robot2, then...".
- Actions:** Hide the character and send a "Game Over" message.

A black bracket on the right side of the script highlights the collision detection and game over logic. Below the script, a text box explains the purpose of this code.

敵キャラの分の当たり判定をつくります。上は敵キャラ2機の場合です。

