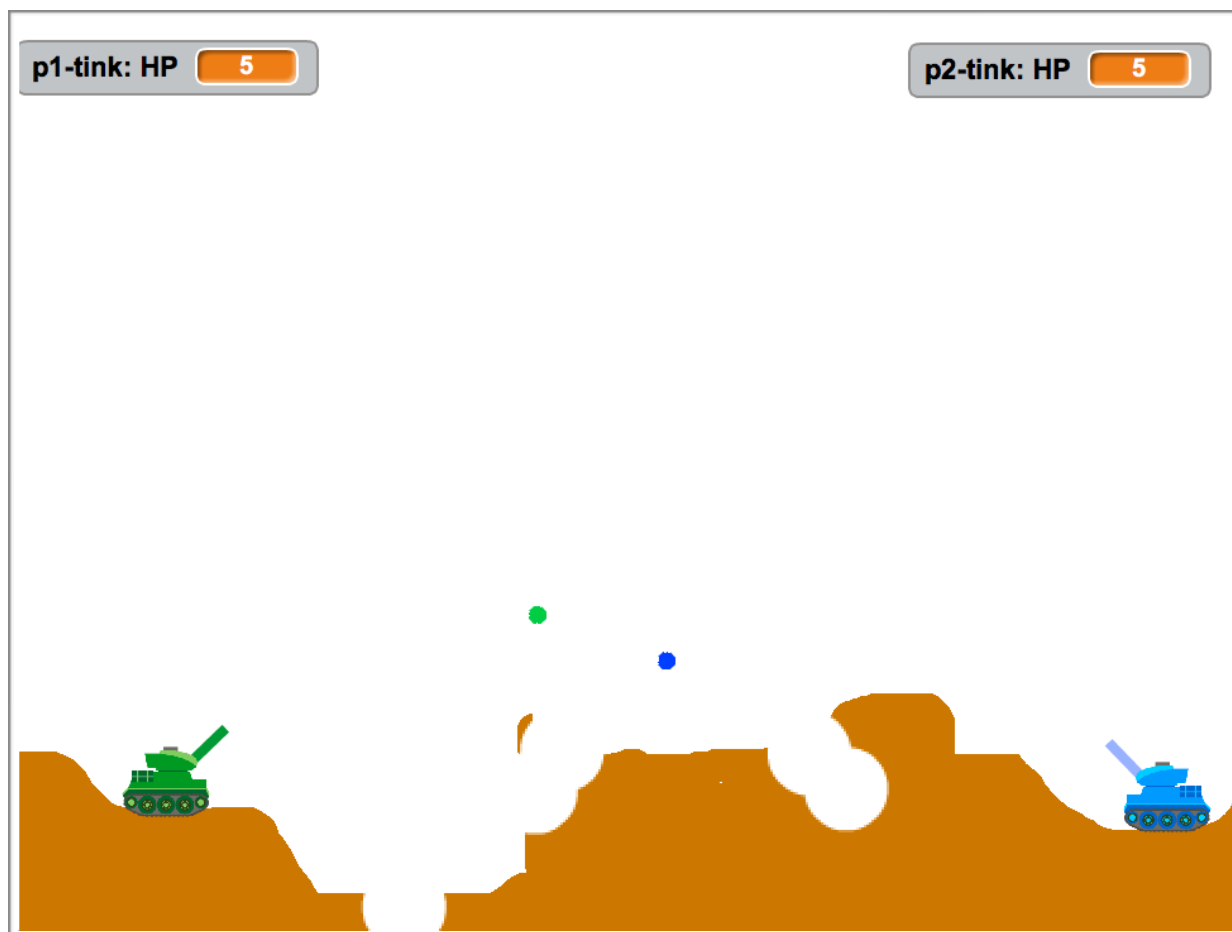


# 対戦 戦車ゲーム



## 対戦戦車ゲームを作ろう

今回作るゲームは、2両の戦車が打ち合って戦う、対戦ゲームを作ります。

ゲームを作る前に仕様を説明します。

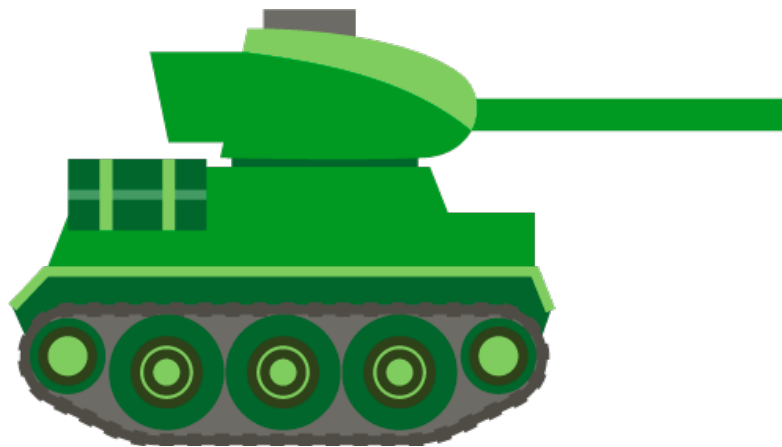
戦車の体力は、5です。敵の弾に当たるとと1減り、0になると負けとします。また、ステージの下に落下しても負けとします。

戦車は、左右移動、大砲の角度変更、射撃、ジャンプが出来るもとし、操作方法は、以下のようになります。

一人目のプレイヤー (1P)	二人目のプレイヤー (2P)
左右移動 : A・D	左右移動 : J・L
大砲の角度 : W・S	大砲の角度 : I・K
射撃 : E	射撃 : O
ジャンプ : Q	ジャンプ : U

撃った弾は、地面と敵戦車にぶつかった時、爆発して地面を破壊します。

それでは、作っていきましょう。



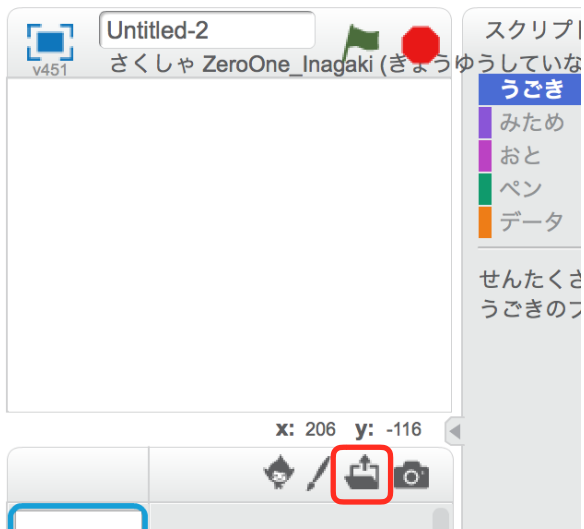
## スプライト（絵）を用意する

まずは、ネコを消して、戦車のスプライトを用意します。  
左下のネコを右クリックして、さくじょをクリックします。



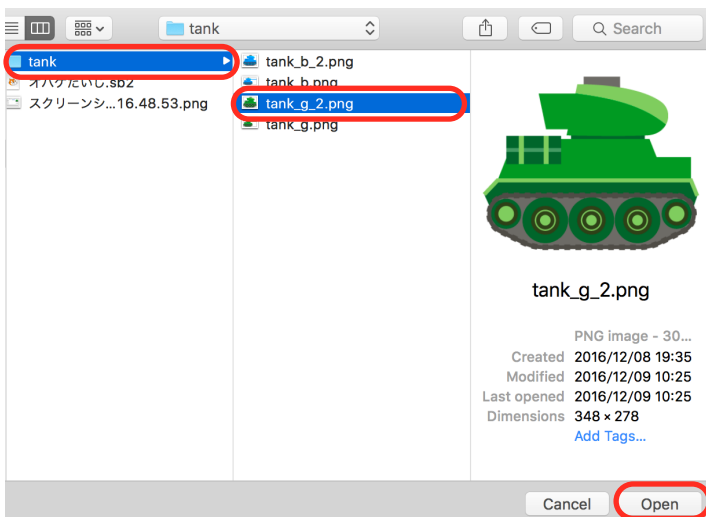
スプライトを追加するのですが、今回は、Scratchで用意されているものではなく、PCから読み込んで使います。

フォルダのアイコンをクリックします。



事前に配布した tank フォルダの中にある tank-g-2 を選択して Open（オープン：開く）をクリックします。また、tank-g-2.png と表示されている場合もあります。

tankフォルダをもらっていない子は、先生にもらっていないよと教えて下さい。



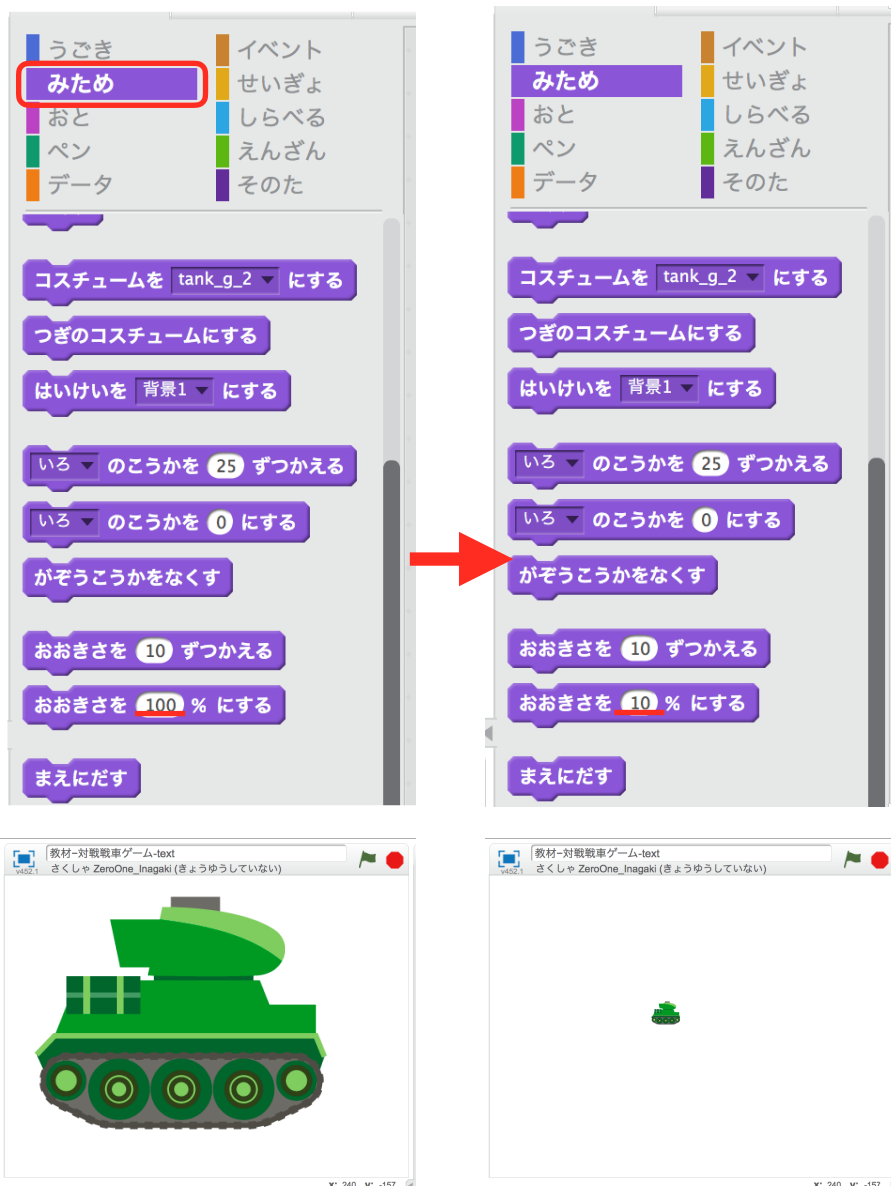
## 大きさを変える

先程、追加した戦車のスクリプトは、大きいです。これでは、戦車が二台収まらないので、大きさを変えます。

みためからおおきを100%にするを10%にします。

100の部分をクリックして、10を入力してください。

その後、ブロックをクリックすると戦車が10%の大きさになり、さっきより小さくなれば成功です。



## 左右移動

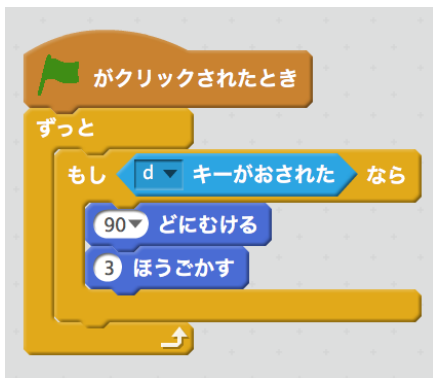
戦車を左右に移動させます。

まずは、右へ進むスクリプトを作ります。下の図のようにブロックを出します。  
(スクリプトとは、ブロックを組み合わせたもの)



そして、下の図のように組み立てます。

スペースキーがおされた を dキーがおされた に変えます。この戦車は、1P用の戦車なのでA・Dキーで移動するためです。



緑の旗を押して動かしてみましょう。Dキーを押したら右へ進めは成功です。

### 解説 スクリプトの解説

D (d : デイ) キーをおした時に、右へ進むスクリプトを作りました。  
緑の旗がおされたら ブロック ずっと により永遠に繰り返されます。繰り返される内容は、もしdキーがおされた時に90度(右)にむけて、3歩動くものです。

### 解説 ○歩動かすと向き

ブロック ○歩動かす は、スクリプトが向いている方向に進むブロックです。  
向きは、○度向ける や ○度回す などを変えることができます。Scratchでは、0度を上として、時計回りに加算されていきます。  
つまり、上は0度、右は90度、下は180度、左は-90(270)度となっています。今回の右に進むスクリプトは、言い換えると90度の方向に進むスクリプトともいえます。

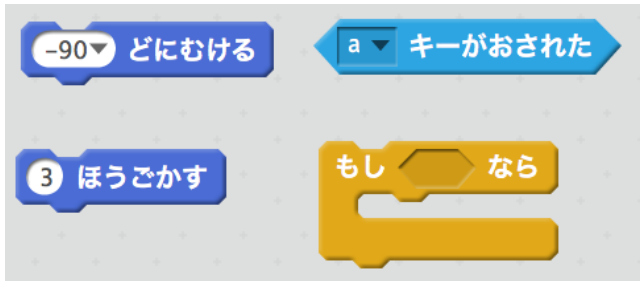
つづいて、左にも進むようにします。

ここで問題です。

今のスクリプトにブロックを追加して、左に進むようにするにはどうしたらよいでしょうか。

この時、戦車が逆さまになってもよいことにします。

ヒントは、下の図のブロックを使います。



ちなみに、プログラムの世界では、答えは一つではありません。

今回の左に動く処理を作る方法は、たくさん存在しています。処理した結果が左に動けばそれが答えの一つです。

ここで出すヒントと答えは、一つの例です。あなたが考えたものと違っていても、左に動いていれば正解です。

答え



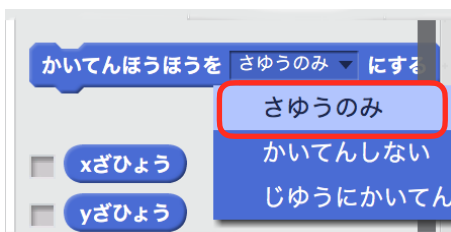
## 逆さまにならないようにする

左に動くと戦車が逆さまになってしまいます。これをならないようにします。

うごき から かいてんほうこうを さゆうのみ にする をクリックします。



さゆうのみ ではない場合は、下向き三角をクリックし、さゆうのみ にします。その後、ブロックをクリックします。



動かしてみましよう。左向きの時、逆さまにならないければ成功です。

### 解説 ブロック 回転方向を~~にする

ブロック 回転方向を~~にする は、○歩動かすで動く方向、正面方向に対して絵をどのように回転させるかを指定します。

自由回転 : 正面方向に向く

左右のみ : 向きが 0~180度の時、右向き

-1 ~ -179度 又は、181~359度の時、左向き

回転しない : 向きに関係なく回転しない

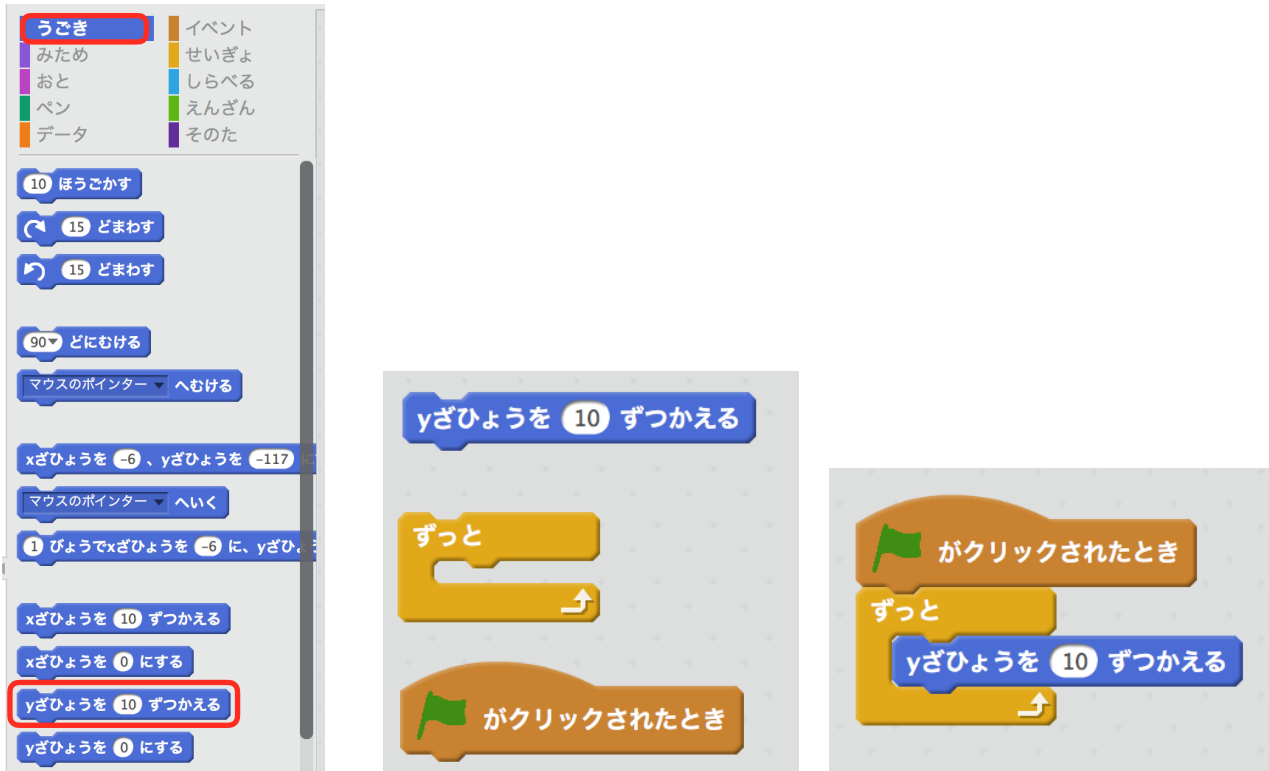
## 重力により落下する

重力で下に落ちるようにします。

180度にむける を行って 1ほうごかす を繰り返しても良いですが、今回は、別の方法で動かします。yざひょうを○ずつかえる を使う方法です。

うごき から yざひょうを○ずつかえる を出します。

それを ずっと で囲み、 緑の旗がクリックされたら も付けます。

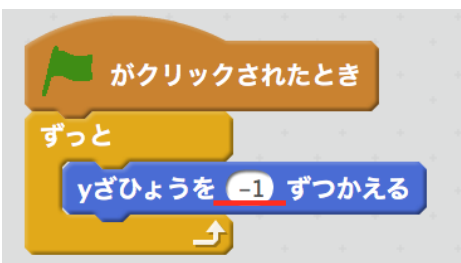


移動処理と区別が付きやすくするため別のスクリプトで作りました。

ブロック 旗がクリックされたとき が複数あっても問題ありません。緑の旗をクリックした時に接続してある全てのスクリプトが動きます。

動かしてみましよう。上にいってしまいます。

yざひょうを10ずつかえる の10を-1に書き変えます。



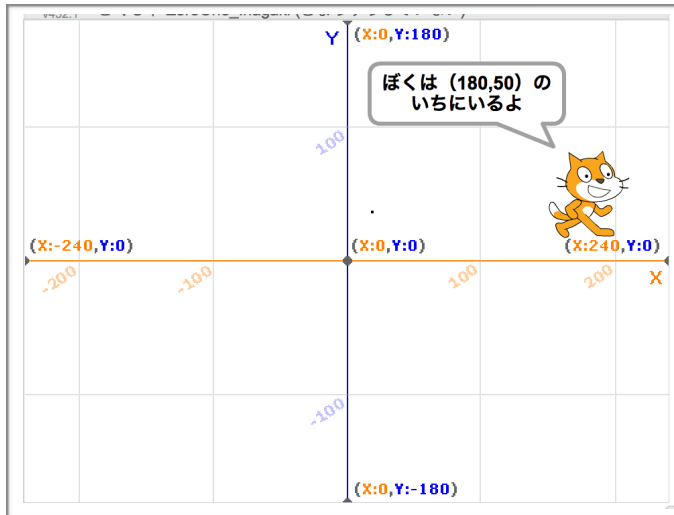
もう一度動かしてみましよう。下に落ちたら成功です。



## 解説 座標とは

ゲームにおける座標とは、モノの位置を数字で示したものです。二次元空間では、(X, Y)のように2つの数字で表現し、三次元空間では、(X, Y, Z)のように3つの数字で表現します。

Scratchは、ステージの中心を原点(0, 0)とした、二次元空間です。  
ちなみに全てのゲームは、座標でモノの位置を管理しています。

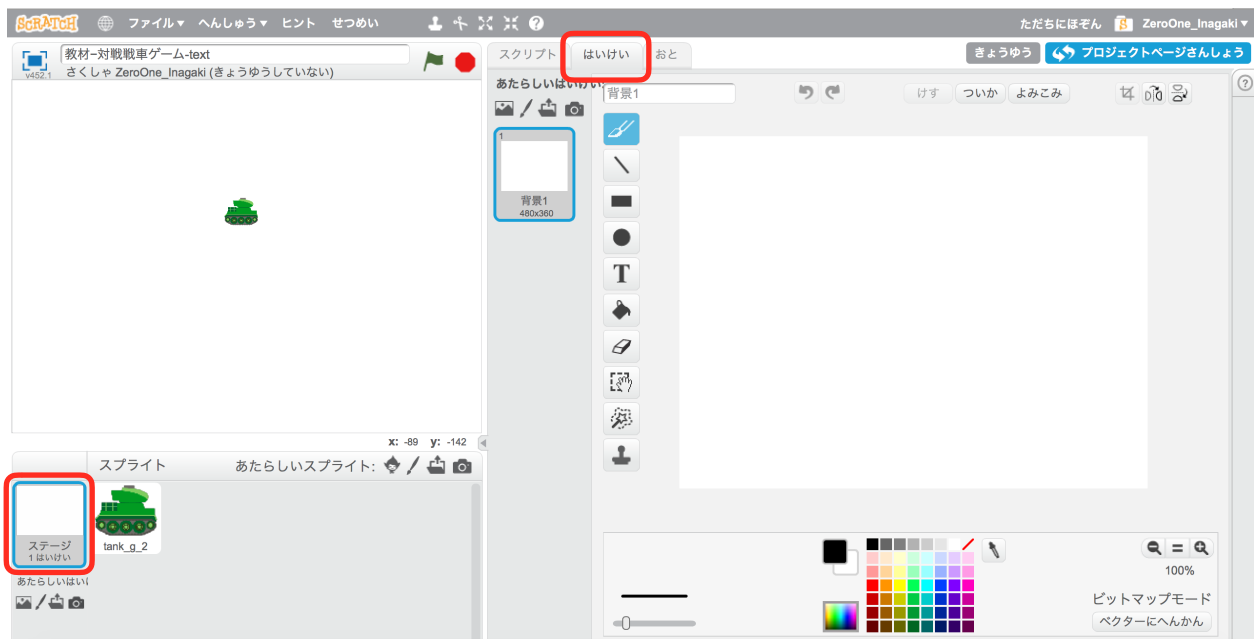


X座標が180、Y座標が50にいる猫の図

## 地面を描く

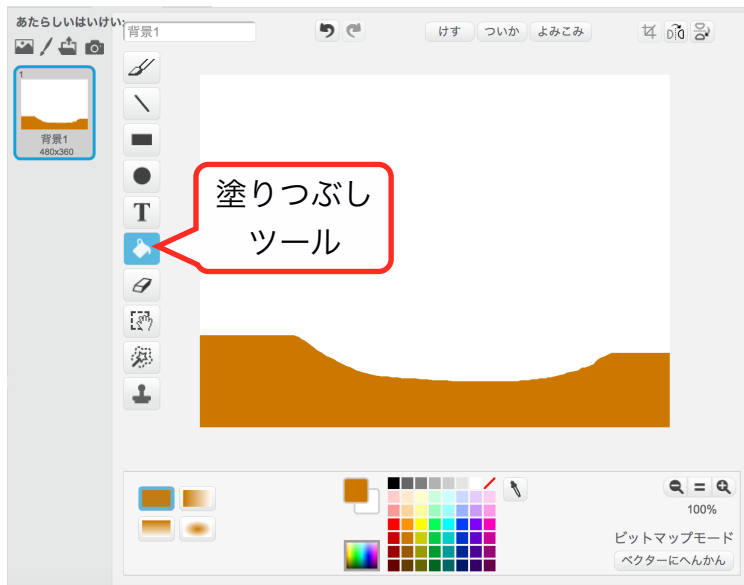
ステージに地面を描きます。

左下から、ステージを選択し、上のタブの はいけい をクリックします。



次に、下のパレットから好きな色を選んで地面を描きます。ただし、使うのは一色だけです。なぜなら、ここで使う色で、地面と戦車の当たり判定を作成するからです。

塗りつぶしたいときは、バケツを使うときれいに塗りつぶせます。



## 地面との当たり判定

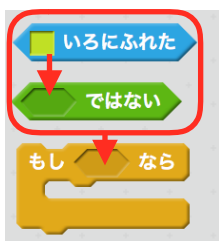
地面についた時、重力によって落下しないようにします。

キーとなるブロックは、しらべる にある □いろにふれた です。

このブロックを もし～なら の六角形に入れて使い、”もし 地面の色に触れたら、重力の処理を行わない”というスクリプトに作り変えます。

左の図のようにブロックを用意します。

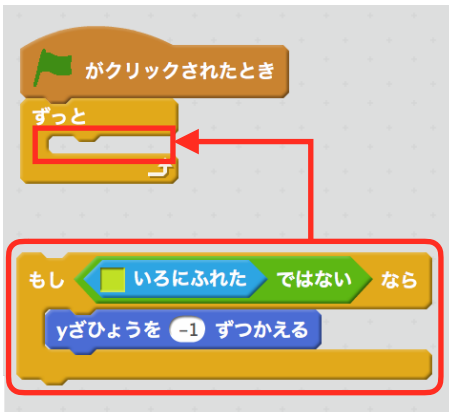
次に重力処理のスクリプトと右の図のように組み合わせます。



y ざひょうを -1 ずつかえる を先程組んだものの中に入れて。



ずっとの中に戻します。



色を変更します。

□いろにふれた の色の部分をクリックします。その後、変えたい色がある場所つまり、地面をクリックします。



動かしてみましよう。戦車が落下して行って地面に触れた時、止まれば成功です。

## よりリアルな重力にする

重力で落下するようになりましたが、速度が一定で不自然です。  
実は実際の重力による落下は、加速していきます。なので、加速するように改良します。

まず、変数を作ります。

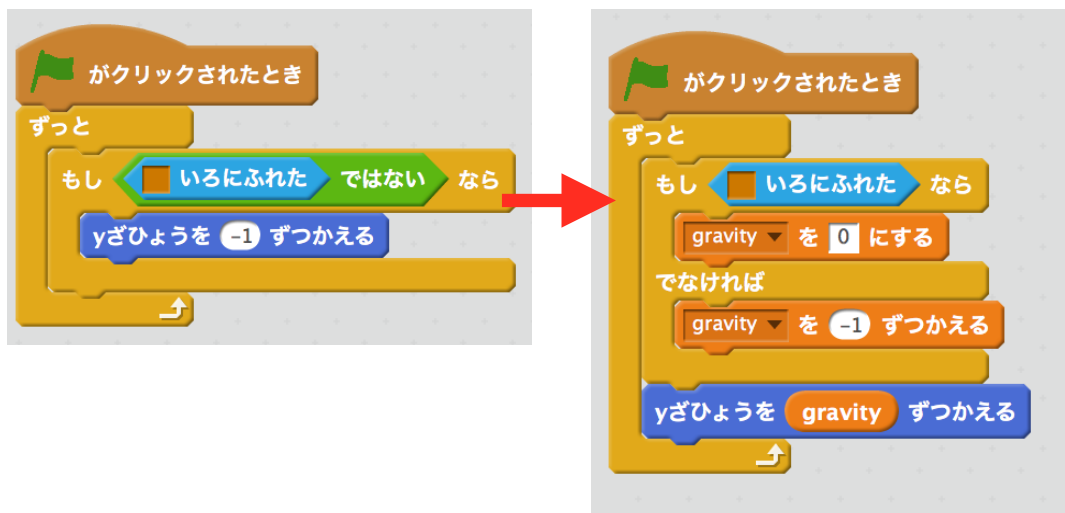
データ から へんすうをつくる をクリックします。

変数名を入力して、このスプライトのみを選択し、OKをクリックします。

変数名は、なんでもいいですが、教材では「gravity (グラビティ：重力)」にしました。



重力処理のスク립トを右の図のように変更します。



動かしてみましよう。落下中に加速すれば成功です。

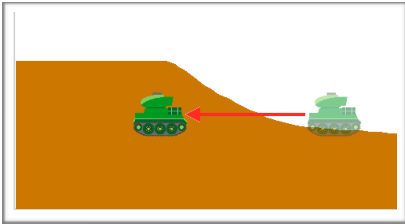
### 解説 スクリプトの解説

変数 gravity には、一回の処理に下に落ちる量が入ります。一回目は-1、二回目は-2、3回目は-3…のように徐々に増えていき、結果として落下速度が加速していくのです。

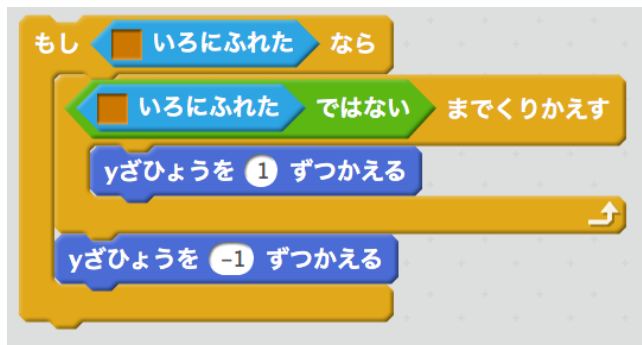
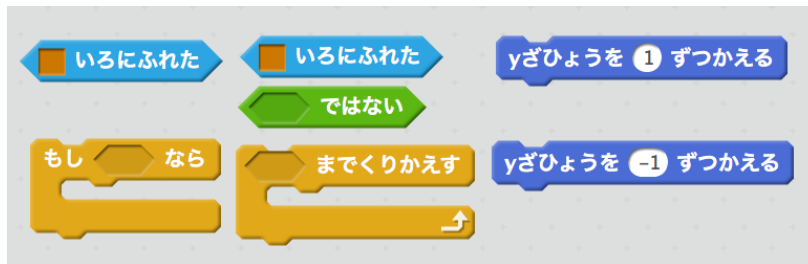
また、茶色に触れたら0にもするようにしているので、茶色に触れているうちは、常に0で地面にめり込むことはありません。

## 坂を登る

重力処理を作ったおかげで坂を下ることが出来ます。しかし、坂に登ろうとすると埋まってしまう。この問題を解決します。



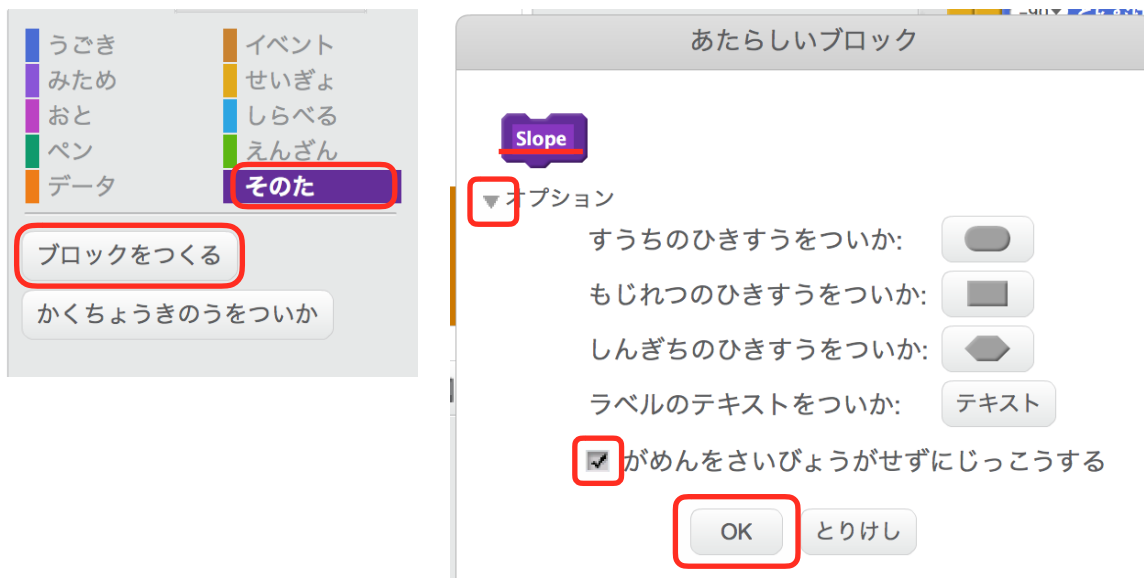
まず、下の図のようにブロックを用意し、組み立てます。



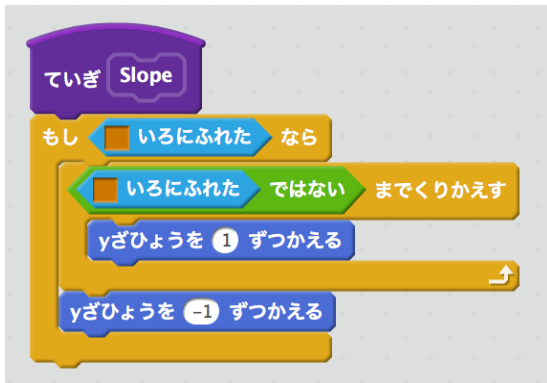
次に、そのた から ブロックをつくるをクリックします。

紫色の中に「Slope」（スロープ：坂）と入力します。ちなみに違っていても動作に支障はありません。

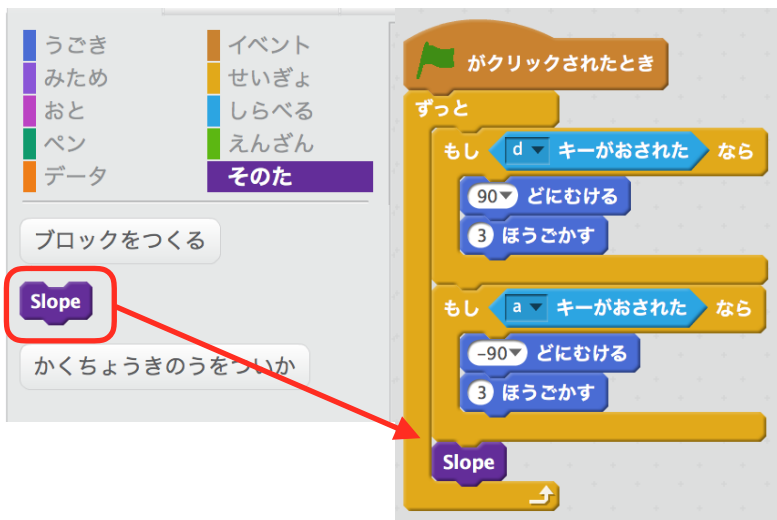
オプションの三角をクリックして画面を再描画せずに実行するにチェックを入れて、OKをクリックします。



すると、Slopeと書かれた紫色のブロックが出てくるので、それを先程作ったスクリプトの上に付けます。



そのた の中に Slopeというブロックが増えています。それを左右移動処理のスクリプトの最後に付けます。



動かしてみましょう。坂を登れば成功です。

## 解説 紫のブロック

紫のブロックは、定義に付けたブロックの代わりとして動作します。

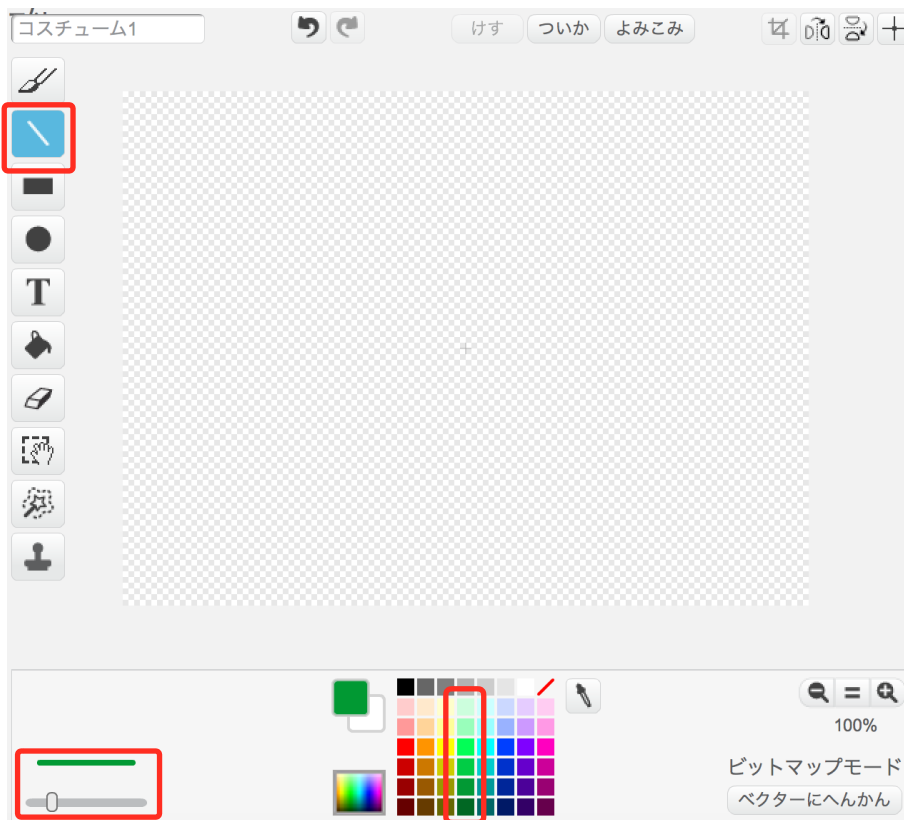
主に同じ処理を複数の場所で使う場合に使用します。

また、画面を再描画せずに実行する にチェックを入れると、定義に付けたブロックの処理が終わるまでステージに変化がなくなります。

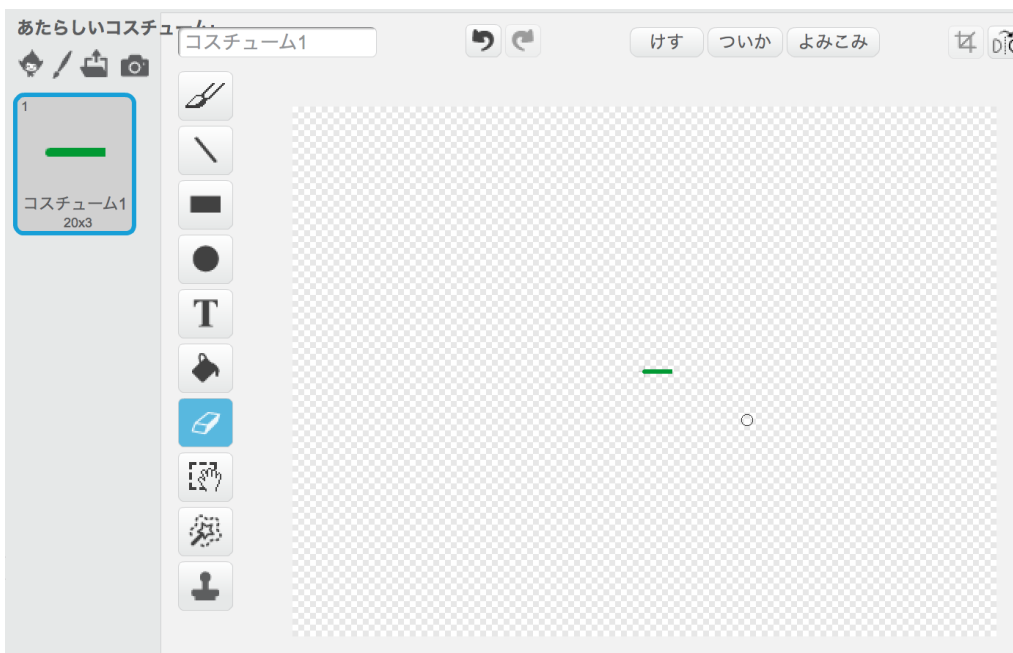
よく分からない人は、一瞬で処理したいときはチェックを入れると覚えましょう。

## 砲身を作る

まず、新しいスプライトを作ります。左下、筆のマークをクリックします。直線ツールを選び、緑色を選択します。左下のバーを左から全体の4分の1くらいにします。これは、ペンの太さです。



真ん中に十字の印があります。これは、回転する時に中心にして回転する点です。この点からまっすぐ右に線を引きます。



## 砲身を本体に追従する

砲身を本体にくっついて移動するようにします。

～へいく を ずっと行えば、くっつけて移動させる事ができます。

左の図のブロックを用意し、右の図のように組み立てます。

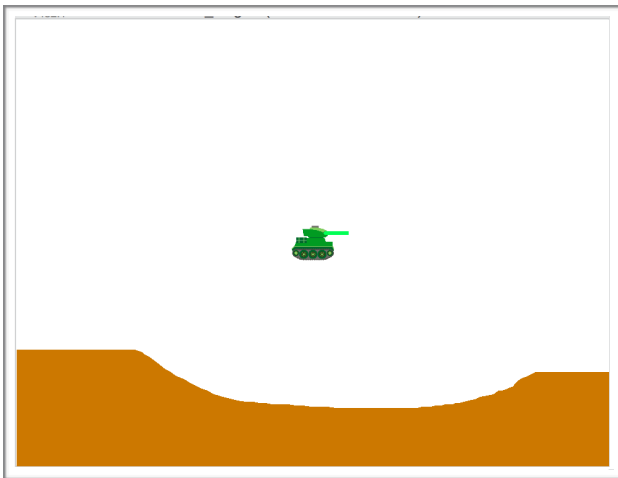


動かしてみましょう。戦車本体と一緒に動けば成功です。しかし、位置がおかしいです、なので修正します。

まずは、どれだけずらすかを測ります。

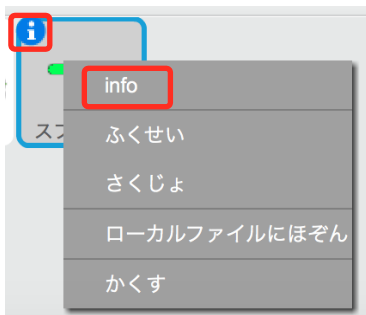
戦車本体を選択し、x 座標を 0 にすると y 座標を 0 にする をそれぞれクリックします。すると、戦車本体がステージ中心、つまり、原点にきます。

この状態で、砲身を本体にくっつけます。



次に、砲身のインフォメーションを開き座標を確認します。

インフォメーションの開き方は、左下の砲身の spritesheet を選択し、青丸の i のマークをクリックするか、右クリックのメニューから Info をクリックします。

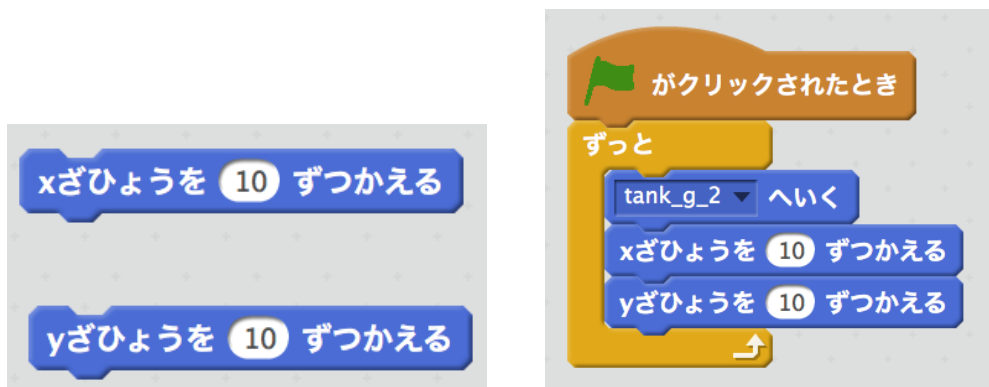




教材での砲身の座標は（10，8）である事がわかります。この数値を覚えておきましょう。あとで使います。



x 座標を○ずつかえる と y 座標を○ずつかえる を用意して、tank\_g\_2へいくの下に付けます。



x 座標を○ずつかえる を調べた x 座標と同じ数値にします。  
y も同じように、y 座標を○ずつかえる を調べた y 座標と同じ数値にします。



動かしてみましょう。砲身が正しい位置で一緒に動けば成功です。  
しかし、左向きの際の位置がおかしいです。次は、そこを修正します。

まず、変数を作ります。

データ から へんすうをつくる をクリックします。

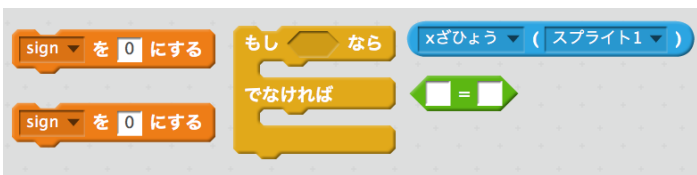
変数名を入力し、このスプライトのみを選択し、OKをクリックします。

変数名は、分かりやすいものならなんでも良いです。教材では、「sign」（サイン：符号）にしました。

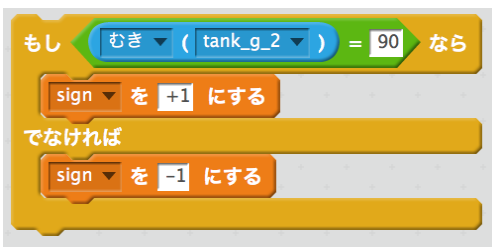
ちなみに、この変数には、+1か-1が入ります。



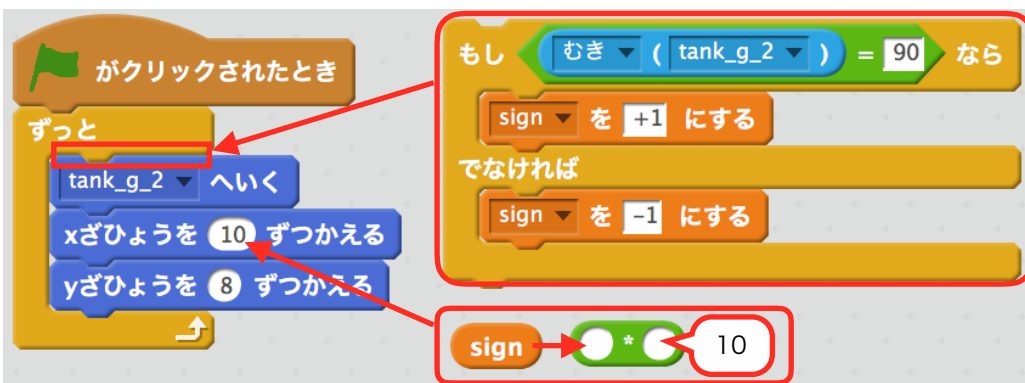
下の図のようにブロックを用意します。  
x ざひょう (スプライト 1) は、しらべる の中にあります。



次に、下の図のように組み立てます。  
ちなみに、x ざひょう (スプライト 1) のカッコ内を tank\_g\_2 に変え、その左側を むき に変えます。sign を 0 にする を sign を +1 にすると sign を -1 にする にします。



組み立てたものを追従処理スクリプトのずっとブロックの中の最初に入れます。  
次に、掛け算のブロックと変数 sign ブロックを新たに用意します。  
掛け算ブロックに sign を入れて、もう一つの白い枠の中には、x ざひょうを〇ずつかえる の〇に書いてある数値を入れます。教材では、10 です。  
sign \* 10 のようになったら、x ざひょうを〇ずつかえる の〇の中に入れます。





これで完成です。

動かして確認してみましょう。左を向いた時、砲身の根本が正しい位置に来れば成功です。今度は、砲身の向きが変わりません。W・Sキーで砲身の角度を変更する処理を作った後に修正します。

## 砲身の角度を変える処理

W・Sキーで砲身の角度を変更する処理を作ります。

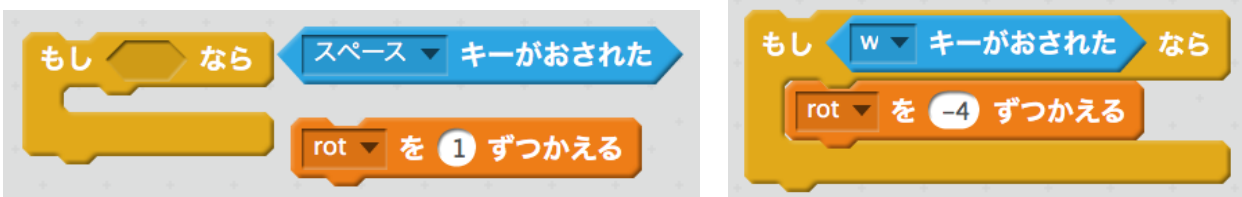
まず変数を作ります。この変数は、向きを管理する変数です。スプライトは、向きを持っていますが、戦車本体の向きと合わせる関係上、必要なので作ります。

変数名を入力し、このスプライトのみを選択します。その後、OKをクリックします。

変数名は、rotation（ローテーション：回転）を略して「rot」としました。



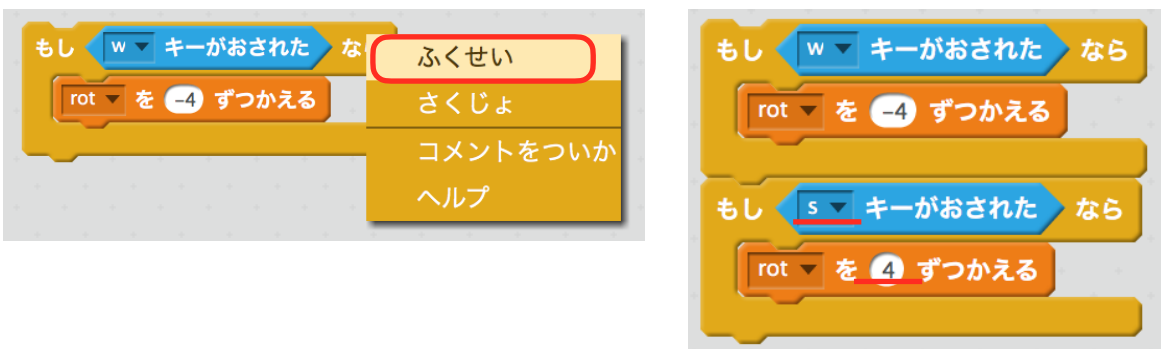
次に下の図のようにブロックを用意し、組み立てます。



複製します。

もしあたりを右クリックし、ふくせいをクリックします。

複製したものを下に付けます。そして、複製したブロックのwをsに、-4を4に変えます。



次に、変数 rot の値をスプライトに反映する処理を作ります。

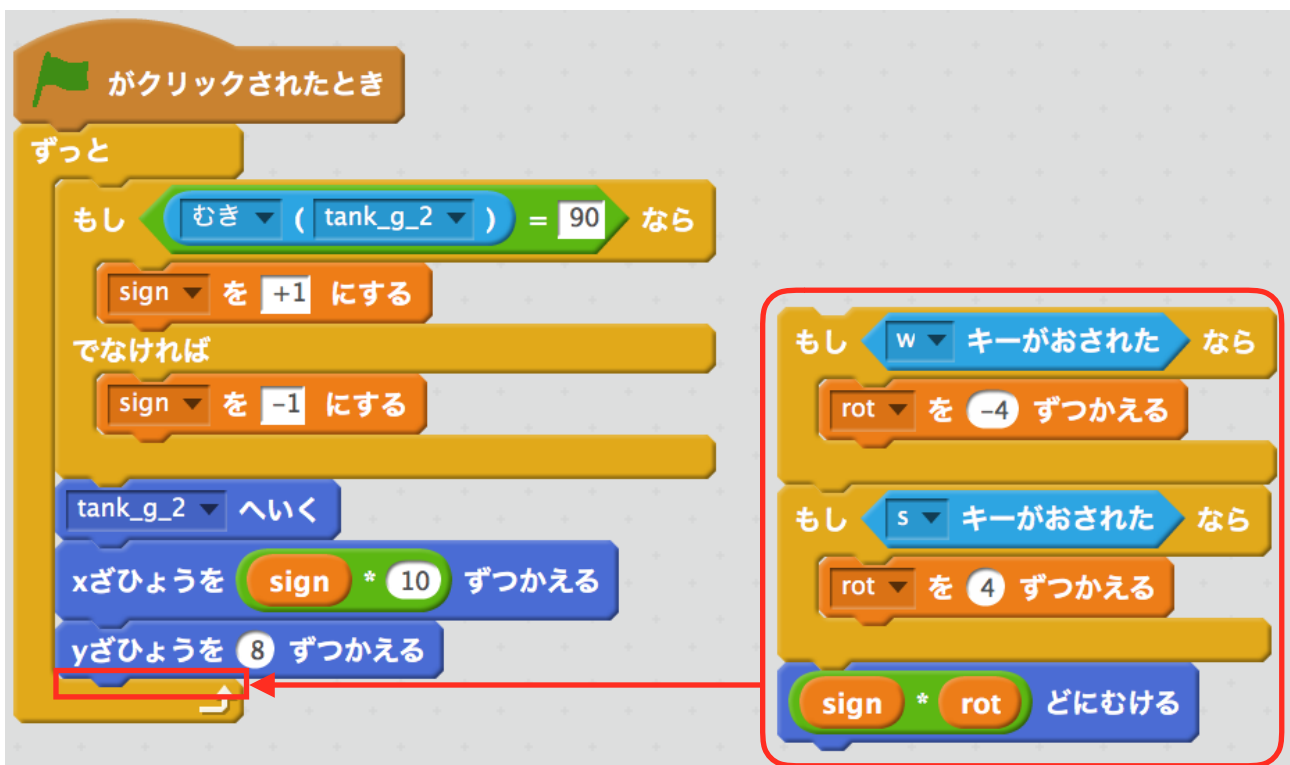
まず、左の図のブロックを用意し、右の図のように組み立てます。



組み立てたものを もし s キーがおされたら の下に付けます。



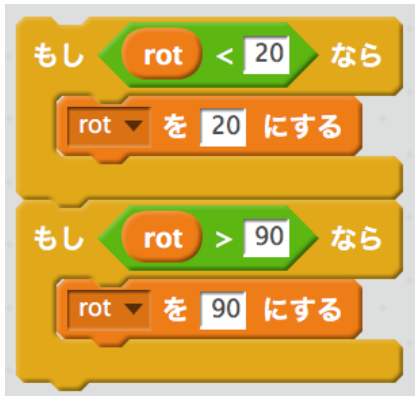
これを、追従処理のずっとブロックの中の一番下に付けます。



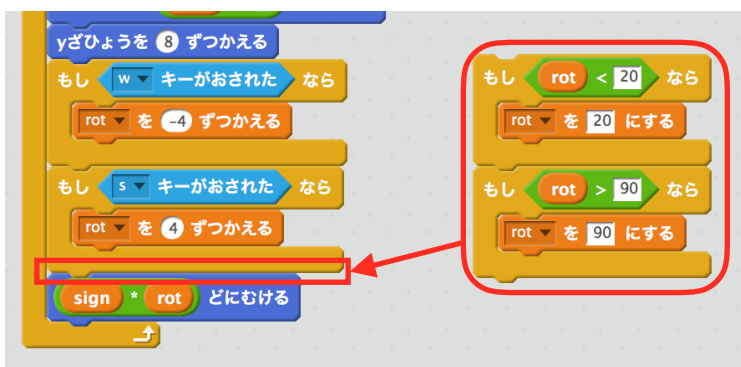
動かしてみましょう。Wキーで上に向きSキーで下に向けば成功です。

## 砲身の角度を制限する

図のようにスクリプトを作ります。



作ったものを○どにむけるの上に付けます。



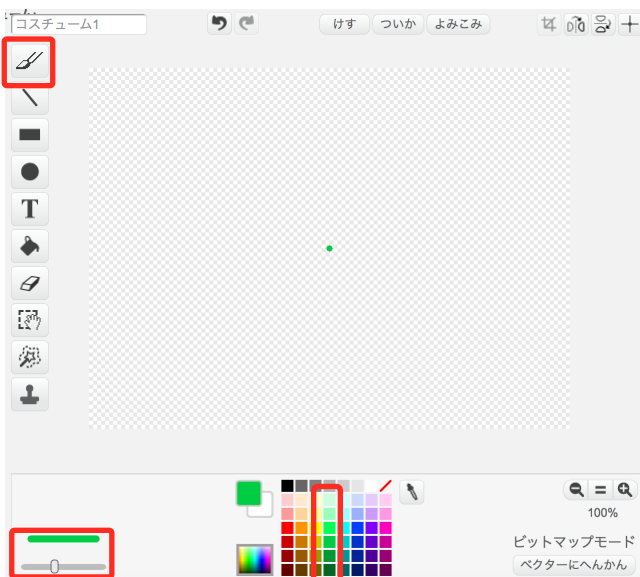
動かしてみましょう。砲身が一定の角度で止まれば成功です。

## 弾を用意する

射撃処理を作って行きます。まずは、弾を用意します。

新しいスプライトを作ります。左下、筆のマークをクリックします。

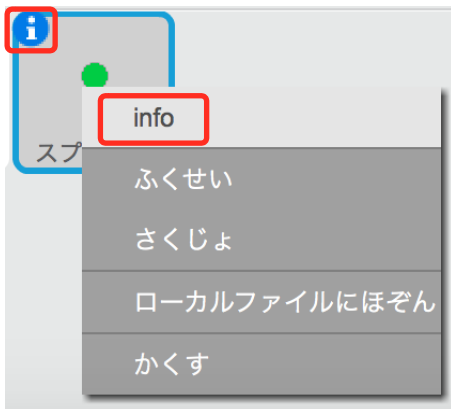
ペンツールを選び、緑色を選択します。さらに、左下のペンの太さを真ん中より少し左の位置にします。キャンバスの中心でクリックし、点を描きます。



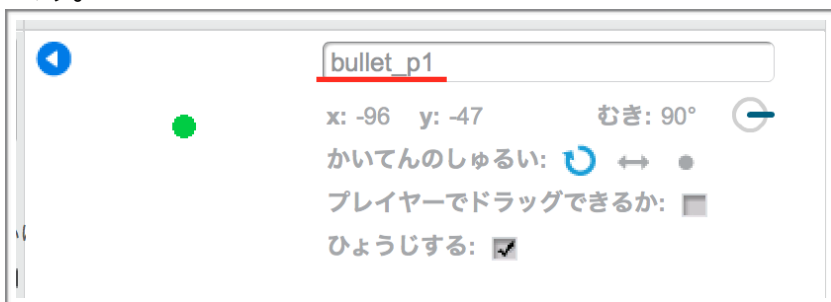
これを弾として使います。

## スプライトの名前を変える

弾のスプライト名が「スプライト2」になっていると思います。わかりにくいので変えます。左下の弾を選択し、青丸のiのマークをクリックするか、右クリックのメニューからInfoをクリックします。

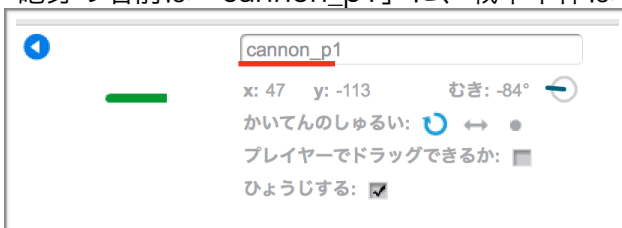


「スプライト2」と書いてある入力欄を書き換えることで名前を変えられます。名前は、プレイヤーの弾であることが分かる名前が好ましいです。教材では「bullet\_p1」にしました。bullet（バレット：弾）とPlayer1（プレイヤー：選手）を略したp1を組み合わせたものです。



砲身と戦車本体も名前を変えましょう。

砲身の名前は「cannon\_p1」に、戦車本体は「tank\_p1」にしました。



## 弾を撃つ

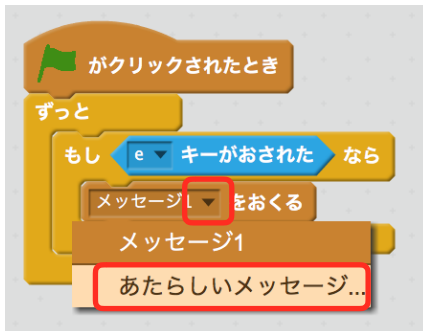
弾を撃てるようにします。

まず、戦車に図のスク립トを追加します。

メッセージ1をおくる は、イベントの中にあります。



メッセージ1をおくる の下向き三角をクリックし、あたらしいメッセージをクリックします。



メッセージ名を入力しOKを押します。

メッセージ名は、プレイヤー1が弾を撃ったというのが分かる名前にしましょう。

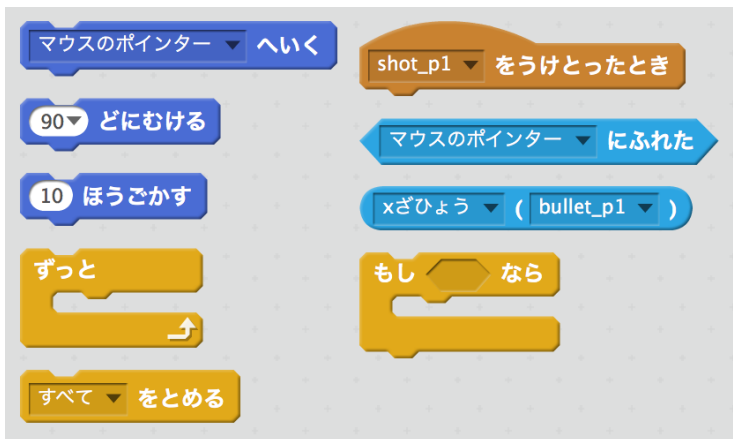
教材では、「shot\_p1」にしました。shot（ショット：発射）と Player1（プレイヤー：選手）を組み合わせたものです。



弾に下の図のようにスク립トを組みます。

shot\_p1 をうけとったときは、イベントの中にあり、このスク립トをとめる は、せいぎょの中にあります。

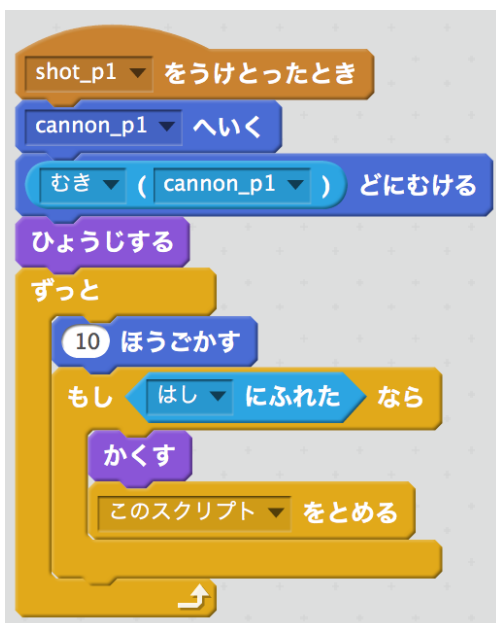




動かしてみましょう。Eキーをおした時に、砲身の向きで弾が飛んだら成功です。

次は、表示と非表示を設定します。

メッセージを受け取り、準備ができたなら表示し、スクリプトを止める前に非表示にします。



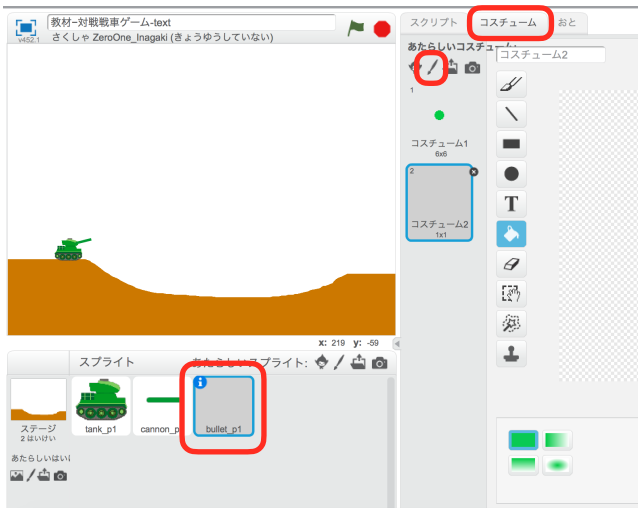
## 弾を爆発させる

地面に触れた弾は、爆発して、地面を破壊するようにします。

まずは、爆風を描きます。

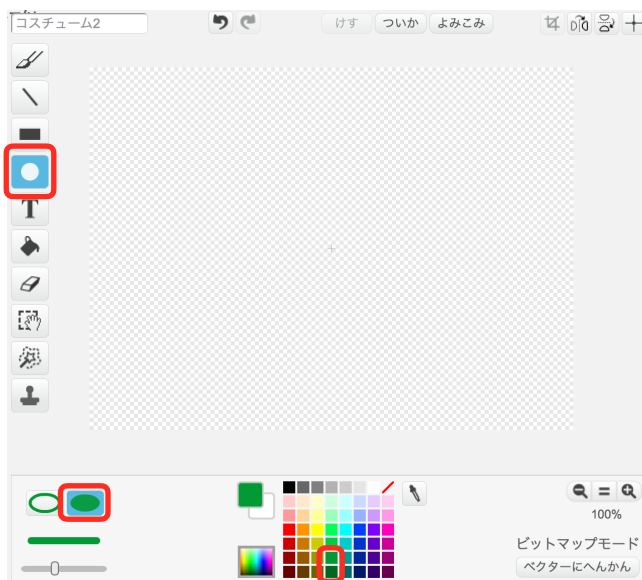
弾を選択して、上のタブのコスチュームをクリックします。

あたらしいコスチュームの下、筆のマークをクリックします。コスチューム2という新しいコスチュームが作られます。



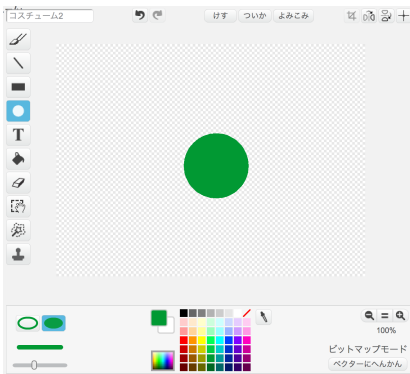
コスチューム2に爆風を描きます。絵に自身がある人は、自由に描いても構いません。うまく描けない人は、教材にそってコミカルな爆風を描きましょう。

まず、円形ツールを選び、濃い緑を選びます。左下の塗りつぶされた丸を選択します。



そして、正円を描きます。

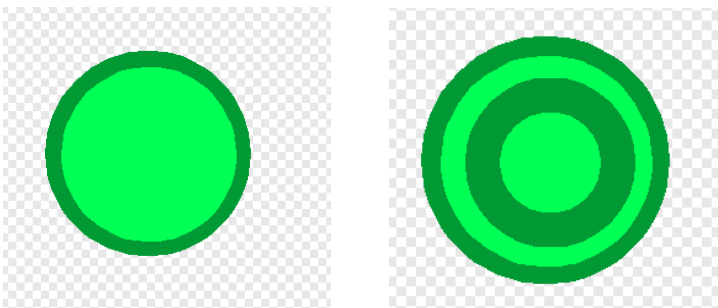
正円の描き方は、shift（シフト）キーを押しながらドラッグアンドドロップと描けます。



次に、今の色より明るい緑を選びます。

今描いた丸の中に、一回り小さい正円を描きます。

そして、暗い緑を選び明るい丸の中に小さな円を描きます。これを繰り返してシマシマな丸を描きます。

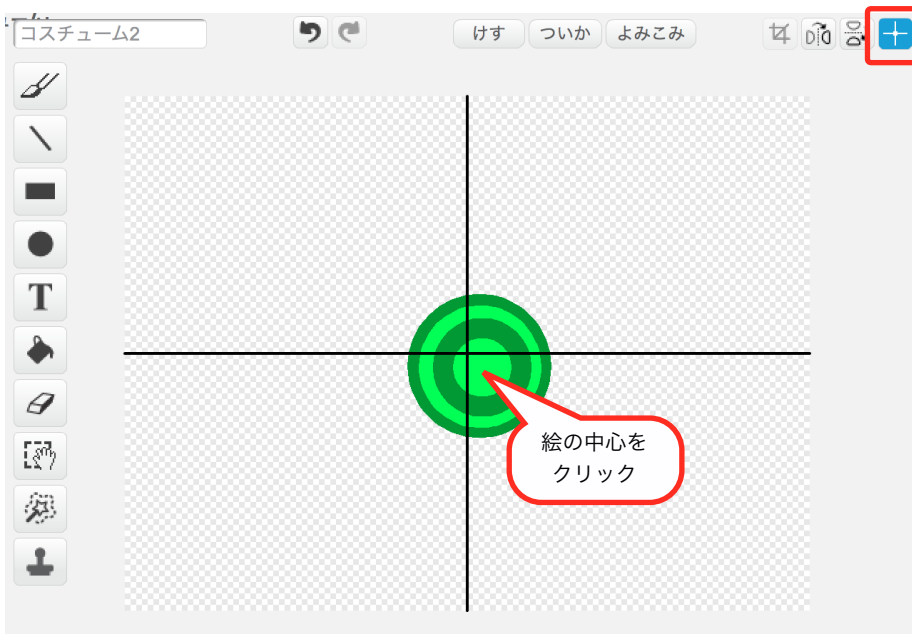


これで爆風の絵は、完成です。

最後に絵を中心に合わせます。

右上の十字のマークをクリックします。すると、十字の線がでてきます。

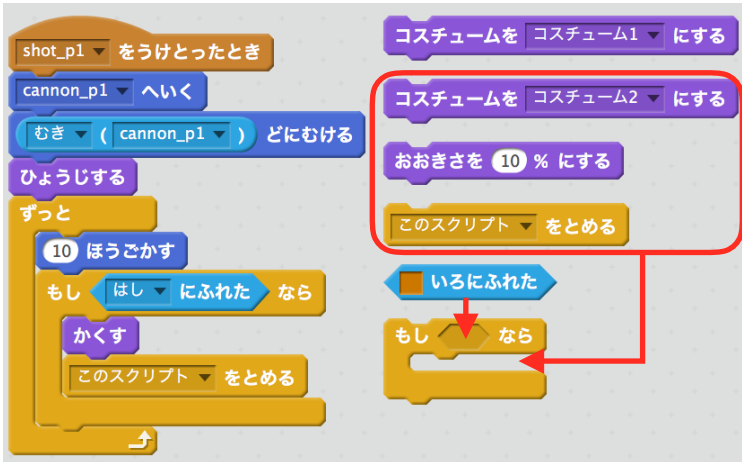
この状態で、絵の中心をクリックします。



これで爆風は、完成です。

## 地面に触れたら爆発するようにする

弾のスク립トを下の図のように変えます。



動かしてみましょう。

地面に当たった時、爆風のコスチュームになれば成功です。

次は、爆風に動きを付けます。

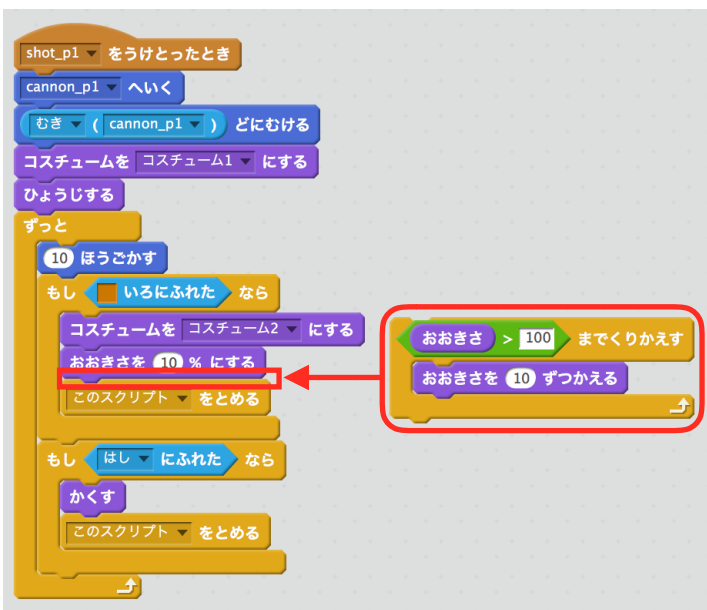
## 爆風に動きをつける

爆風は、広がっていくものです。簡単な動きを付けてみましょう。  
簡単な動きとは、爆風を拡大していくものです。

まず、下の図のようにスクリプトを組みましょう。



大きさが100より大きくなるまで拡大し続けるスクリプトです。これを地面に会った時にコスチュームと大きさを変更した後に入れます。



動かしてみましょう。爆風が大きくなり一定の大きさに止まれば成功です。

最後にスクリプトを止める前に かくす を付けます。  
これにより、爆発を終えたら消えるようになります。



## 爆発で地形を破壊する

爆風の拡大が終わった後の場所に、地面を破壊処理を作ります。

ペン から 左の図のように ペンのふとさを1にすると ペンのいろを□にする、ペンをおろす、ペンあげる を出します。

これらを右の図のように組み立てます。



ペンの色を背景の色にします。

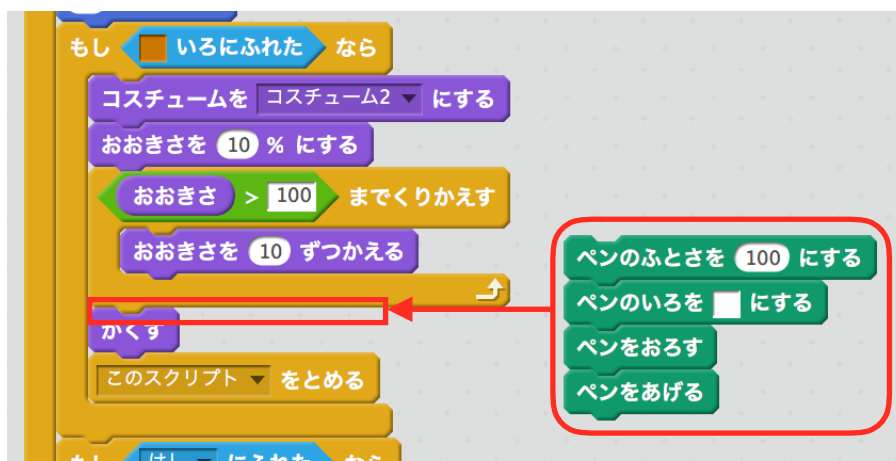
ペンの太さは、爆風のコスチュームの大きさと近い数値にします。

コスチュームの大きさは、コスチュームタブ から コスチューム名の下に表示されています。

教材では、101x100なので、100に設定しました。



この地形破壊の処理を、爆風の拡大が終わった後の場所に付けます。



動かしてみましよう。爆風がなくなった後に地面がなくなれば成功です。

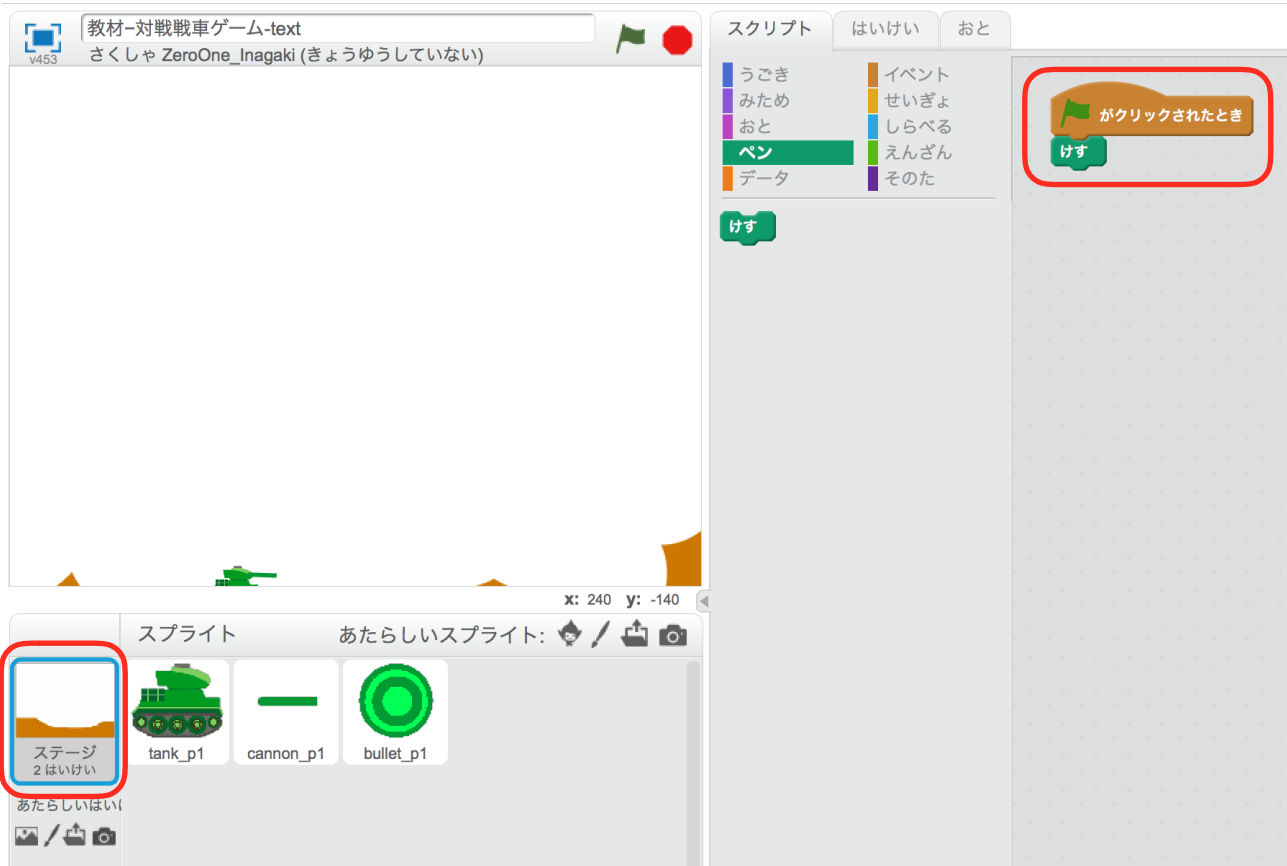
## 地形を元に戻す

破壊したを緑の旗をクリックするたび戻るようにします。

左下のステージを選びます。

イベント から 緑の旗がクリックされたとき と

ペン から けす を出して、組み合わせます。



動かしてみましよう。緑を旗をクリックした時に、地形が戻れば成功です。

## 射撃に制限をつける

射撃に関して、問題があります。それは、弾が飛んでいる時に、射撃するとその弾が消えてしまいます。この問題を解決するために、弾が動作している間は、撃てなくするようにします。

Scratchには、スクリプトが動いているかどうかを判断出来る機能がありません。なので作ります。まず、弾に変数を作ります。この変数は、スクリプトが動作している時、「true (トゥルー：真)」になり、そうでないときは、「false (ファルス：偽)」にするものです。変数名を入力し、このスプライトのみ を選択し、OKをクリックします。変数名は、教材では「Active (アクティブ：機能している)」にしました。



Activeを0にする を4つ出して、1つを Activeをtrueにする にして、3つを Activeをfalseにする にします。



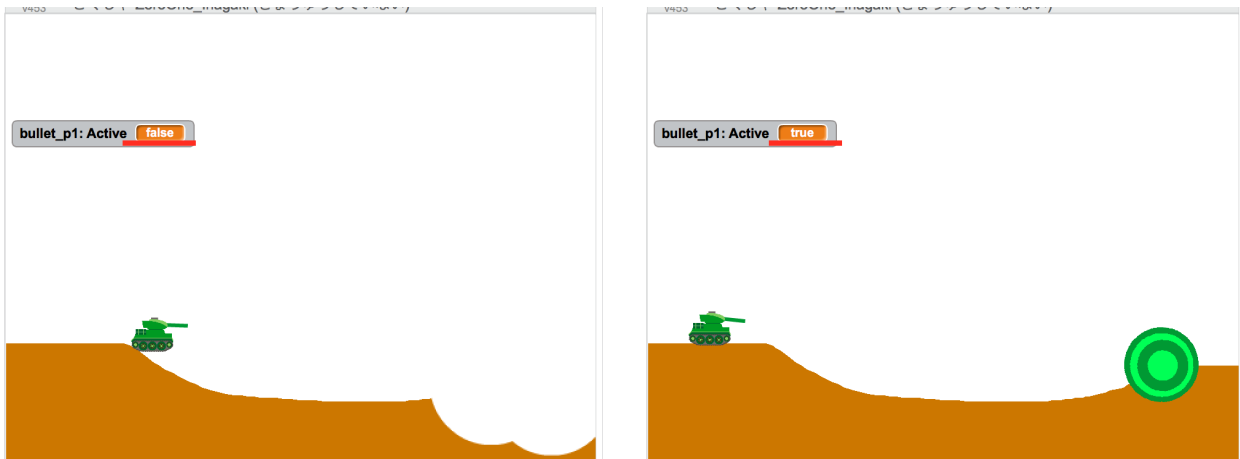
true にする ものは、ひょうじする の下に、falseにする ものの内2つは、それぞれのかくす の下に、残りの1つは緑の旗がクリックされたとき を付けます。







動かして、弾を撃ってみましょう。弾が動いている間 変数 Active の値が true になれば成功です。

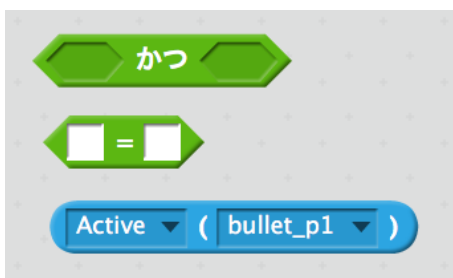


これで、変数 Activeの値を見れば、弾が動作しているかが分かるようになりました。次は、Activeを使って、動作中は、撃てないように制限を付けます。

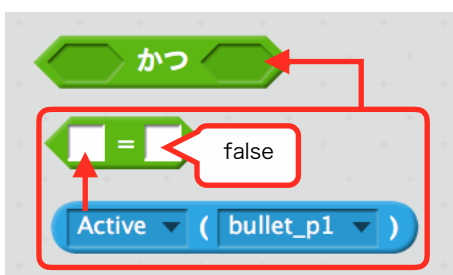
戦車を選択します。

図のようにブロックを用意します。

Active (bullet\_p1) は、しらべる の中の x ぎびょう (tank\_p1) を変えたものです。



= に Active (bullet\_p1) と false を入れ、それを〜かつ〜にいれます。



～かつ～の開いている場所には、射撃処理に使われている eキーがおされた を入れます。



出来たものを もし～なら の六角形に戻して完成です。



動かして、弾を撃ってみましょう。弾が動作している間は、射撃出来なければ成功です。

## 弾に重力にかける

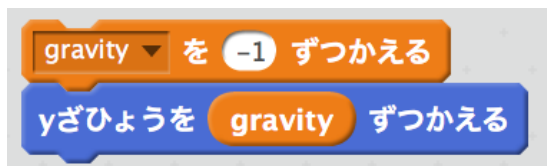
弾に重力をかけます。

原理は、戦車に重力を与えたときと同じです。

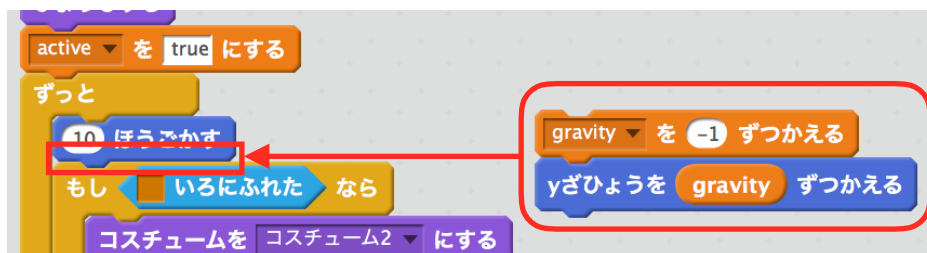
変数を作ります。今回も、このスプライトのみ を選択します。名前は「gravity」にしました。



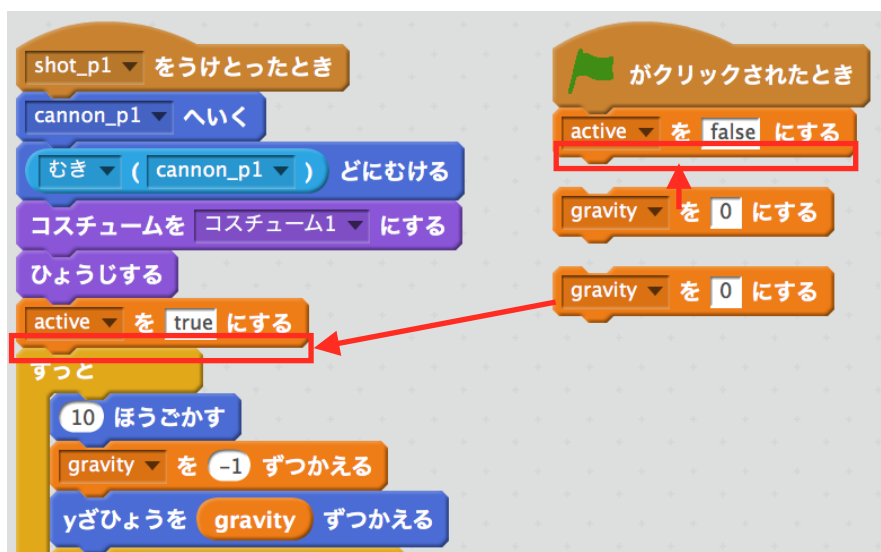
下の図のようにスクリプトを組みます。



作ったものを 10 ほうごかす の下に入れます。



gravity を 0 にする を 緑の旗がクリックされたときの スクリプトと active を true にする の下に付けます。



動かしてみましょう。弾が下に落下していったら成功です。

重力が強くて遠くに飛ばないので修正します。  
gravity を -1 ずつかえる を -0.5 に変えます。



## 瞬間的に壁を登るのを直す

現在 紫のブロックSlopeにより、坂を登っていますが、壁でも同じ動作をします。瞬間的に登るのは、不自然なので修正します。

まず、変数を作ります。

変数名を入力し、このスプライトのみを選択して、OKをクリックします。

変数名は、教材では「slopeY」としました。

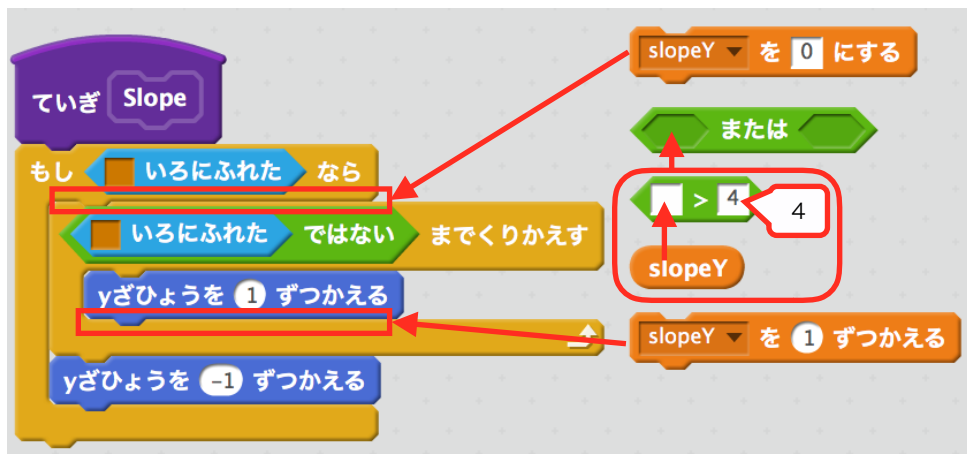
この変数は、Y座標をどれだけずらしたかを数えるものです。



下の図のようにSlopeを変えます。

slopeYを0にする をもし茶色にふれたならの直後に、1ずつ変える をY座標を1ずつかえるの直後におきます。

slopeY は > (だいなり) の左側に入れて、右側には4を入力します。それを または に入れます。



茶色にふれたではない を または の開いている場所に入れます。



出来たものを ~までくりかえす に戻します。



動かしてみましよう。ゆっくりと壁を登れば成功です。

## プレイヤー2の戦車を作ろう

さて、1P用の戦車は、だいたい完成しました。対戦ゲームなので2P用の戦車も用意しましょう。

2P用の戦車は、先程作った1P用とほぼ変わりません。読み返して2P用の戦車を作ってみましょう。この時、2P用にするには、どうしたらよいか考えて、改変しながら作ってください。

答えとして、1Pと違うところのみ説明していきます。

### スプライト（絵）を用意する

2P用の戦車の画像名は「tank\_b\_2(.png)」です。

### 左右移動

左右移動に使うキーは、J（j：ジェイ）、L（l：エル）です。

### 砲身を作る

砲身の色は、青です。

\*間違っても動作に支障はありません。

### 砲身を本体に追従する

ブロック ~へいく は、2P用の戦車です。スクリプト名を変えていなければ tank\_b\_2 になっています。

ブロック むき（~）は、tank\_b\_2 です。

### 砲身の角度を変える処理

砲身の角度を変えるキーは、I（i：アイ）、K（k：ケー）です。

### 弾を用意する

弾の色は、青です。

\*間違っても動作に支障はありません。

### 弾を撃つ

メッセージ名は1P戦車のメッセージ名と同じではないものが正解です。1P戦車のメッセージ名とは、教材で言うところの「shot\_p1」です。

射撃をするキーは、O（o：オー）です。

ブロック ~へいく と むき（~）は、2P用の砲身が正解です。

### 射撃に制限をつける

Active（~）は、2P用の弾が正解です。

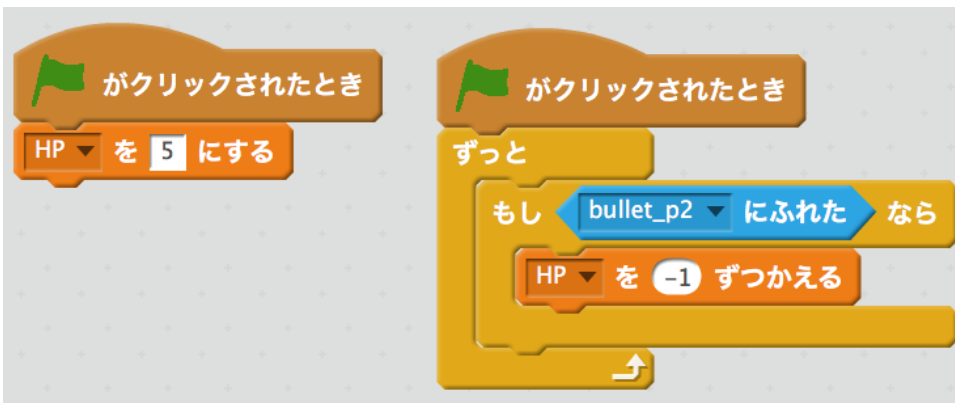
## 体力と当たり判定

戦車に体力と当たり判定を付けます。

まず、戦車に変数を作ります。今回も、このスプライトのみを選択します。変数名は、教材では「HP」としました。



下の図のようにスクリプトを作ります。



動かしてみましよう。HPが5になったのを確認して、2Pの弾にぶつかります。そしたら、一瞬にしてHPが減ったら成功です。

しかし、一発でHPが0以下になっては、勝負にならないので無敵時間を設けます。



## 無敵時間を作る

弾が当たったら数秒間、無敵になる処理を作ります。

まず、変数を作ります。この変数は、無敵が始まった時間を記録するものです。

変数名を入力し、このスプライトのみを選択し、OKをクリックします。

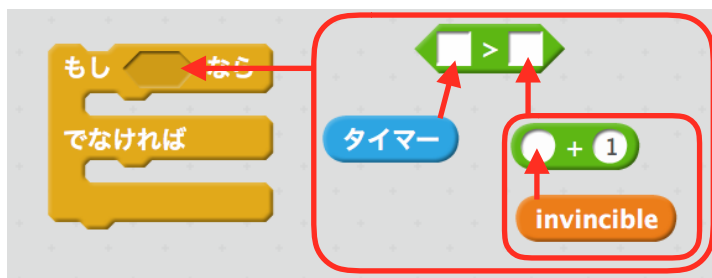
変数名は、教材では、「invincible（インビジブル：無敵）」にしました。



次に先ほど作った2つのスクリプトにブロックを追加します。



新たにブロックを用意して、下の図のように組み立てます。

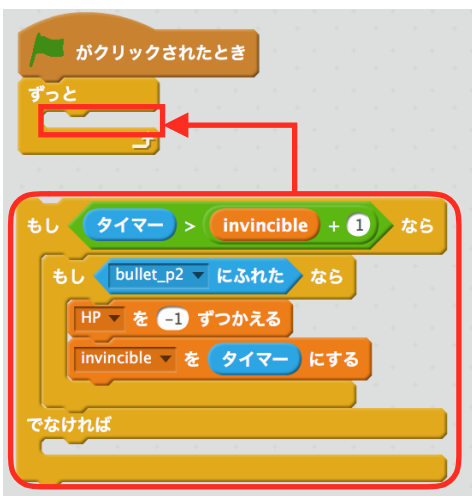


これを当たり判定のスクリプトに組み込みます。

もしbullet\_p2にふれたなら とその中身を全て、もし タイマー>invincible+1 ならに入れます。



入れたら、戻します。



動かしてみましよう。弾に当たったとき、HPが1減れば成功です。ただし、爆風が1秒以上持続するとHPが2以上減る事があります。

また、弾が戦車に当たった時に爆発しません。これは、あとで直します。

## 解説 スクリプトの解説

変数 `invincible` は、弾に当たったときの時間を記録するようにしています。なので `タイマー > invincible + 1` は、以前に弾に当たった時の1秒後に成り立つものになります。1秒以内は、成り立たないので `もしbullet_p2にふれたなら` が処理されません。結果として1秒間は、無敵ということになります。

## 無敵のときは、半透明にする

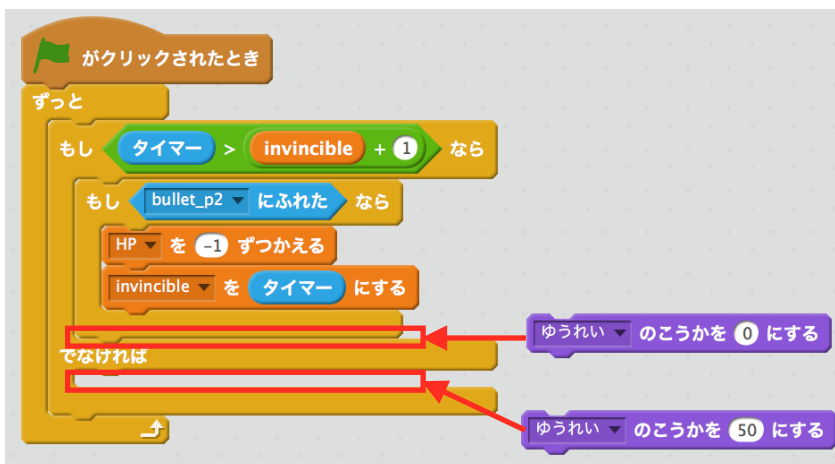
無敵かどうか分かりにくいので無敵時間は、半透明にします。

半透明にするには、**みため** から **いろのこうか**を25ずつかえる か **いろのこうか**を0にする を使います。

まず、**いろのこうか**を0にする を出します。下三角をクリックし、ゆうれいに変更します。それを2つにして、片方を50にします。



0のものを もしbullet\_p2にふれたなら の終わりに付けます。  
50のものを でなければ の中に入れます。



動かしてみましょう。弾に当たって1秒間半透明になったら成功です。

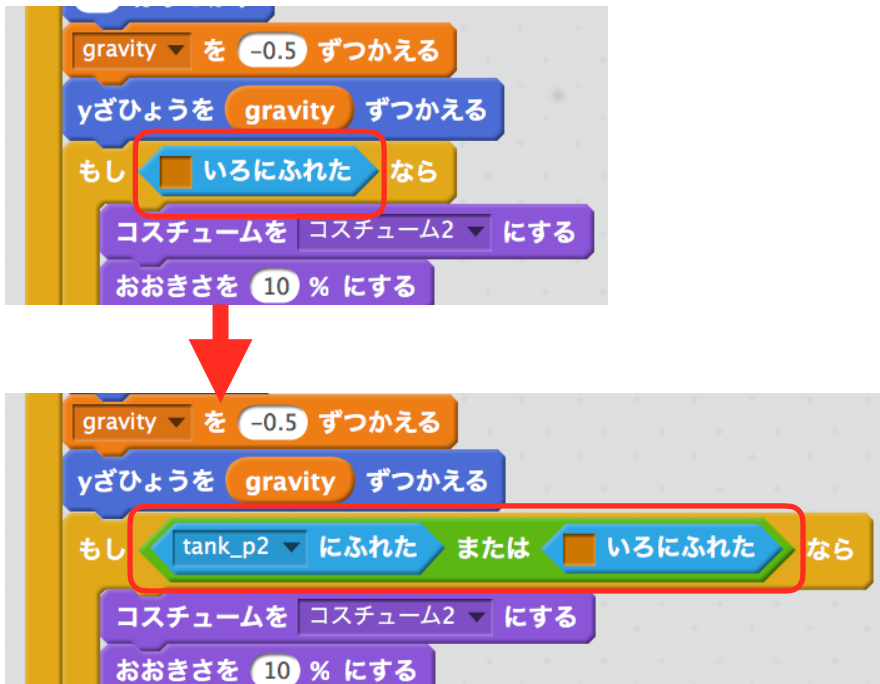
ここまでのスクリプトを2P戦車にも作りましょう。

2P戦車の場合、bullet\_p1になります。

## 弾が戦車に当たったときも爆発する

弾が戦車に当たったときも爆発するようにします。

もし 茶色にふれたら なら を もし tank\_p2 にれてた または 茶色にふれた なら に変えます。



2Pの弾にも同じことをします。ただし、2Pの場合、tank\_p1になります。

動かしてみましょう。弾が戦車当たった時、爆発すれば成功です。

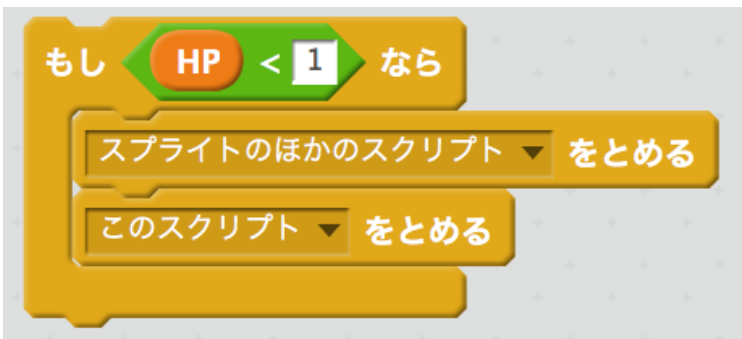
## ゲームオーバー処理を作る

HPが0になったらゲームオーバーとし、戦車をのスク립トを止めます。  
まずは、HPが0以下になった時の判定を取ります。

新たにブロックを用意して、以下の図のようなスク립トを作ります。



この中に スプライトのほかのスク립トをとめる と このスク립トをとめるを入れます。  
この2つは、せいぎょ 中の すべてのをとめる を変えたものです。



このスク립トを当たり判定の処理のずっとの中の一番下に入れます。



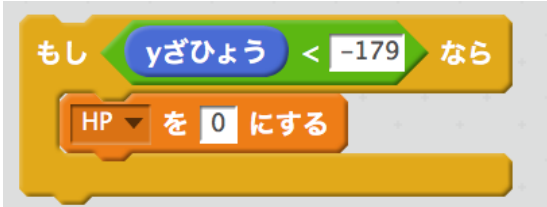
出来たら、動かしてみましょう。HPが0以下になった時、左右に動かなくなったら成功です。  
2P戦車にも同じ処理をつくりましょう。

## 落ちたらゲームオーバーにする

ステージの下に行ったらゲームオーバーつまり、HPを0にします。

戦車に下の図のようにスクリプトを作ります。

yざひょうは、うごきの中にあります。



これを もしHP<1 なら の前に入れます。



2P戦車も同じ物を作ります。

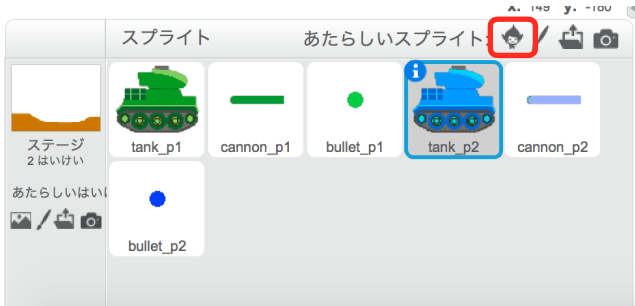
動かしてみましよう。ステージの下に言ったらHPが0になり、操作できなくなったら成功です。

## 戦車が爆発する

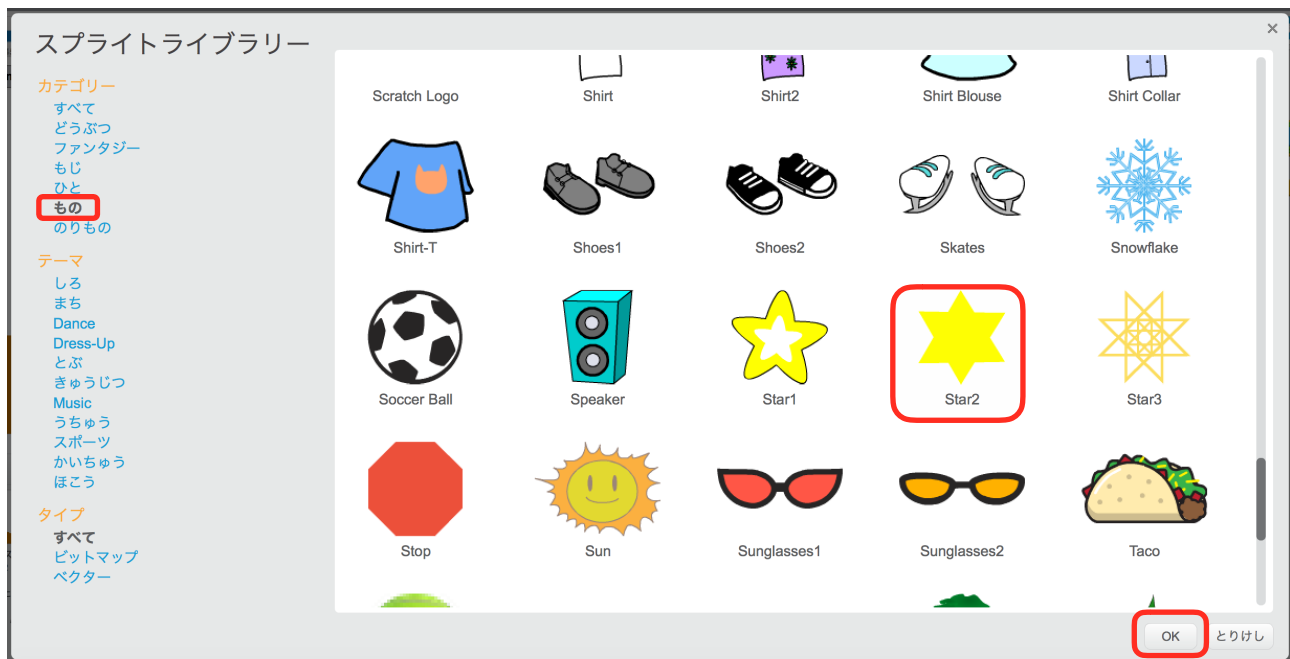
HPが0以下になった時、戦車が爆発するようにします。

まず、爆発の sprites を用意します。

左下の新しい sprites にある顔のようなマークをクリックします。



左側のものをクリックし、右の枠から、Star2を探して、選択し、OKをクリックします。

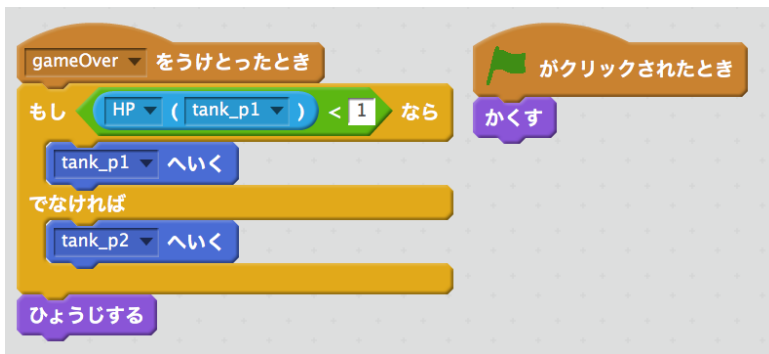


Star2にメッセージをうけとったとき をだして、新しいメッセージを作ります。

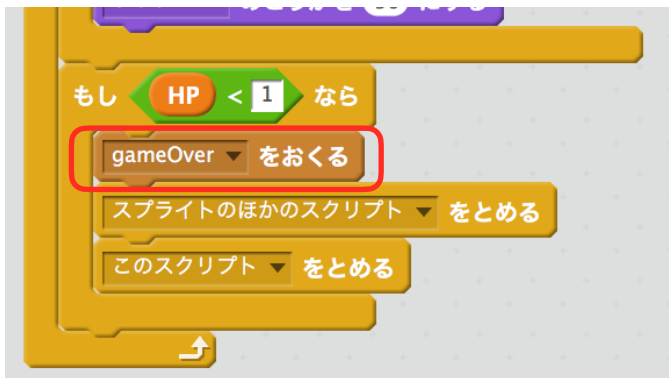
メッセージ名は「gameOver (ゲームオーバー)」にしました。



下の図の スクリプト を作ります。



次に、戦車の当たり判定の無敵処理のスキプトのもしHP<1ならの中の最初にgameOverをおくるを追加します。



2P戦車にも追加します。

動かしてみましよう。HPが0になった時に Star2 が現れたら成功です。



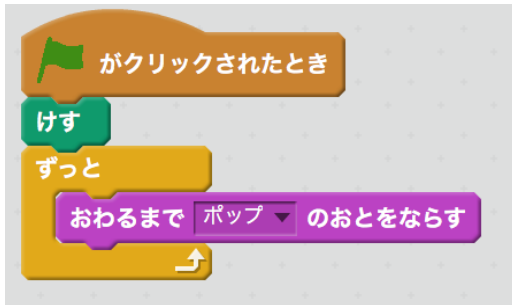
## BGMをつける

音を鳴らすには、おとの中になる～のおとをならす等を使います。

BGMは、ステージに鳴らしてもらいましょう。

ステージを選択し、おとの中からおわるまで～のおとをならすを出します。

そのブロックをずっとで囲み、けすの下に付けます。



動かしてみましょう。何かしらの音がなり続けるとのを確認します。

もしならない場合は、ブロックが間違っているか、PCの音が小さいかのどちらかの可能性が高いです。

次は、音を変えます。

上のおとのタブをクリックします。

あたらしいおとの下、スピーカーのマークをクリックします。



たくさんの音が出てきます。今回は、BGMなのループするものを選びます。

左側のおんがくのループカテゴリーを選び、右側から好きな音楽を選びます。選んだらOKをクリックします。

また、右向き三角（再生ボタン）を押すと音楽を鳴らして確認できます。



あたらしいおとの下に新たに音が追加されたのを確認してします。



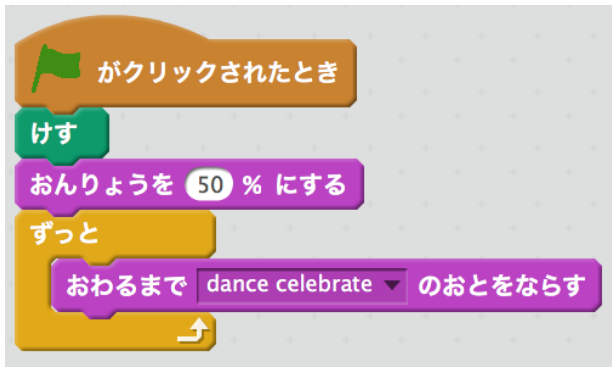
スクリプトタブをクリックします。

おわるまで~のおとをならす の下三角をクリックし、あたりに追加した音を選択します。ここに出てくるものは、先程のおとのタブに表示されている音です。



動かしてみましよう。音が変わっていたら成功です。

これからの作業がやりにくいのでBGMの音の大きさを下げてください。  
おとの中から おんりょうを100%にする をだします。  
これを50%にし、ずっと の前に入れます。



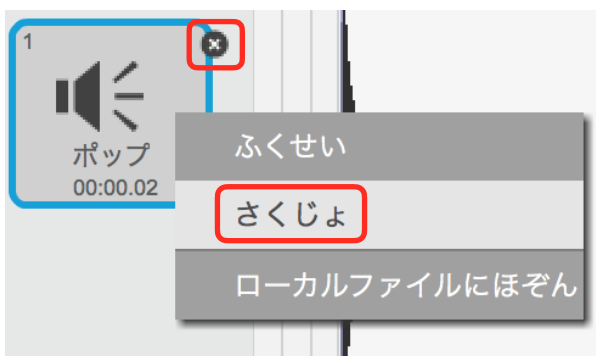
## 射撃SEと爆発SEをつける

射撃時と爆発時の効果音を付けます。

弾を選択し、音楽タブをクリックします。  
新しい音から「drum bass3」と「wolf howl」を追加します。

drum bass3 は、がっきカテゴリーの中にあり、  
wolf howl は、どうぶつカテゴリーの中にあります。

元からある「pop (ポップ)」は、必要なもので消します。選択し、右上のバツをクリックするか、右クリックのメニューからさくじょをクリックします。



スクリプトタブをクリックします。  
drum bass3をならす を ひょうじする の下に入れます。



動かしてみましょう。射撃時に音が鳴れば成功です。

または、BGMの音が大きくて聞き取りづらい時もあります。その時は、ステージを選択し、ブロック おとのおおきさを50%にする の数値を小さくして、BGMの音量を小さくしましょう。

次は、爆発時の効果音を付けます。

Scratchには、爆発の音がなかったので、それっぽい音を作ります。

おとのタブをクリックします。

先程追加した。wolf howl を選択します。



黒い波のようなものが表示されています。これは、音を目に見える状態にしたものです。この音を編集して、それっぽい音を作ります。

全体の右側7割ほど選択します。選択の仕方は、ドラッグアンドドロップ出来ます。



この状態で、へんしゅう から カットをクリックします。  
すると音が短くなります。



次は、右側を少し選択して、こうかを から フェードアウトをクリックします。



なんちゃって爆発音の完成です。

スクリプトタブをクリックします。

wolf howl のおとをならす を用意して、おおきさを 10% する の下に入れましょう。



動かしてみましよう。爆発時におとが鳴れば成功です。

2P戦車に射撃、爆発効果音を付けましよう。

## 戦車が爆発時のSE

戦車のHPが0になったときにも効果音を鳴らすようにします。

Star2を選択して、おとのタブをクリックします。

もとからある音は使わないので消します。

wolf howl を追加します。今回は編集せずにそのまま使います。

スクリプトタブをクリックします。

wolf howlのおとをならす を用意して、ひょうじする の下に付けます。



動かしてみましよう。 戦車のHPが0になった時におとが鳴れば成功です。