

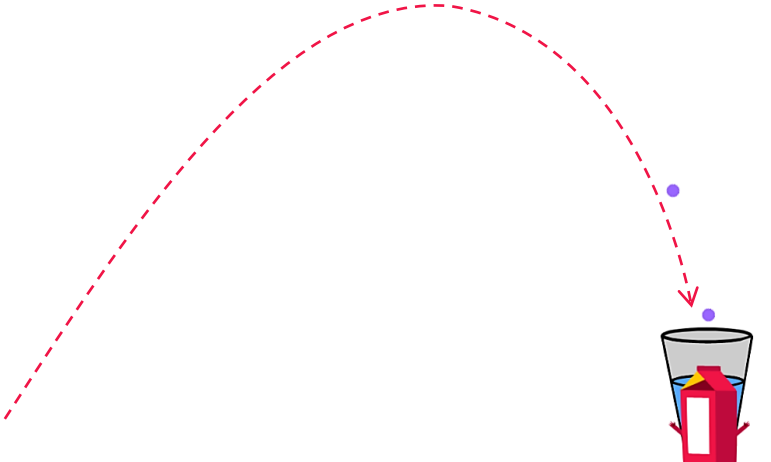
スクラッチ プロジェクト

角度 0

投げる力 0

豆乳を投入！

角度と投げる力を設定して豆乳をクリック



豆乳を投入！



- スプライトやステージの背景を描く
- 「角度」「投げる力」を入力/設定、重力を考えて豆乳の放物線（ほうぶつせん）を描く
- 投入成功か失敗かを判定するアルゴリズムを考える

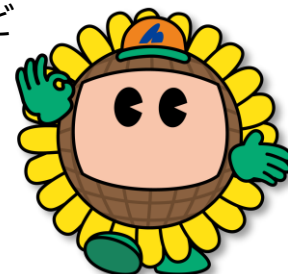
◆ 学習すること ◆

- 変数の表示
- 音を選ぶ、鳴らす
- 論理演算 (and, or)
- ブロック定義
- プログラムを保存する など

※ スクラッチの基礎知識、基本操作はチャレンジ【1】～【5】で確認してね

保護者の方へ

- 課題を解決するための手順を「アルゴリズム」と言います。トライ＆エラーの繰り返しで問題解決能力を育みます。
- スクラッチのアカウントを作成するとインターネット上にプログラムを保存できたり公開できたりします。



「放物線」を描くロジック（仕組み）

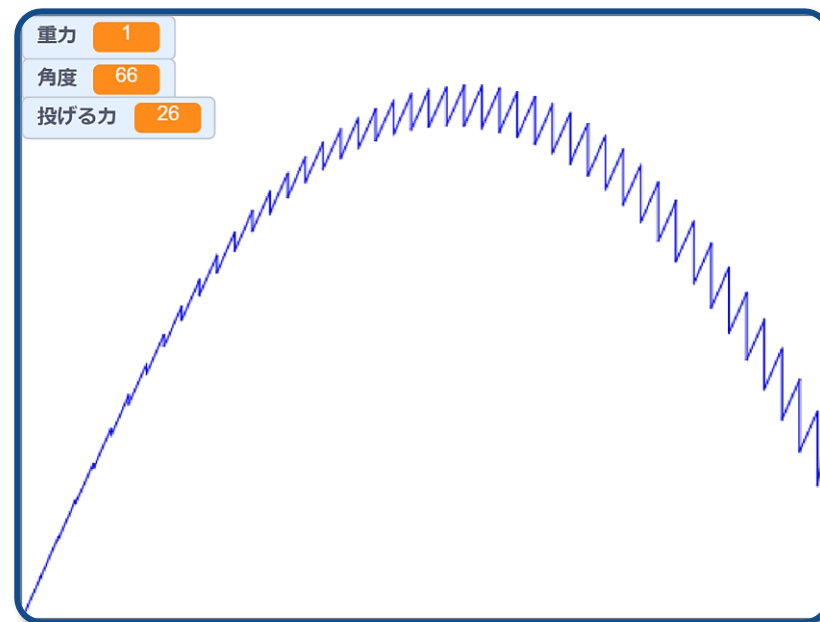
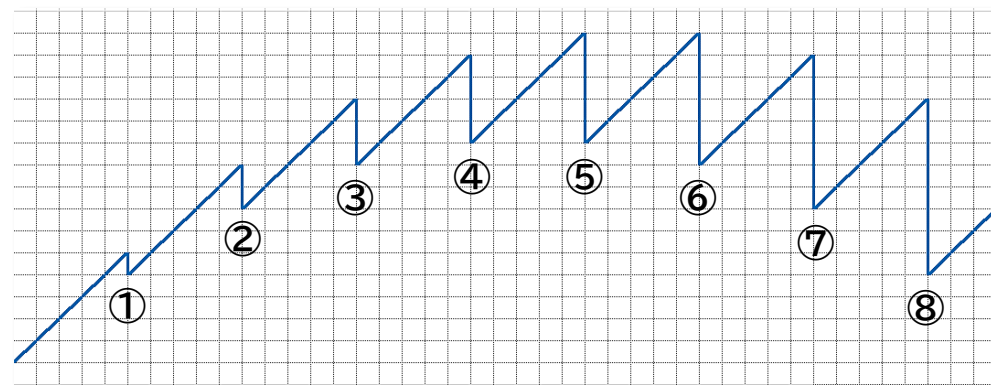
放物線（ほうぶつせん） … 投射した物体が落ちるまでに描く軌跡 投げる力、角度、重力や空気抵抗で高さや距離は変わるけど、すべての放物線は相似。（形が同じ、重なる）

◆ 放物線の軌跡を簡単にしたロジック ◆

- ① 5 (投げる力) 飛んで、1 (重力) 落ちる
- ② 5 (投げる力) 飛んで、2 (重力×2) 落ちる
- ③ 5 (投げる力) 飛んで、3 (重力×3) 落ちる
- ④ 5 (投げる力) 飛んで、4 (重力×4) 落ちる

⋮

The image shows a Scratch script for drawing a parabolic trajectory. The script starts with a '全部消す' (Clear) block, followed by several 'を' (set) blocks: '重力' (gravity) to 1, '落下値' (falling value) to 0, '角度' (angle) to 66, and '投げる力' (throwing force) to 26. Then, 'x座標を' (set x coordinate) to -240 and 'y座標を' (set y coordinate) to -180. A '90 - 角度' (90 - angle) block is used to '度に向ける' (turn degrees). The script then enters a loop: 'ペンを下ろす' (pen down), '50 回繰り返す' (repeat 50 times), '投げる力 歩動かす' (step throwing force), 'y座標を 落下値 ずつ変える' (change y coordinate by falling value), '落下値 を 0 - 重力 ずつ変える' (change falling value by 0 - gravity), and finally 'ペンを上げる' (pen up).

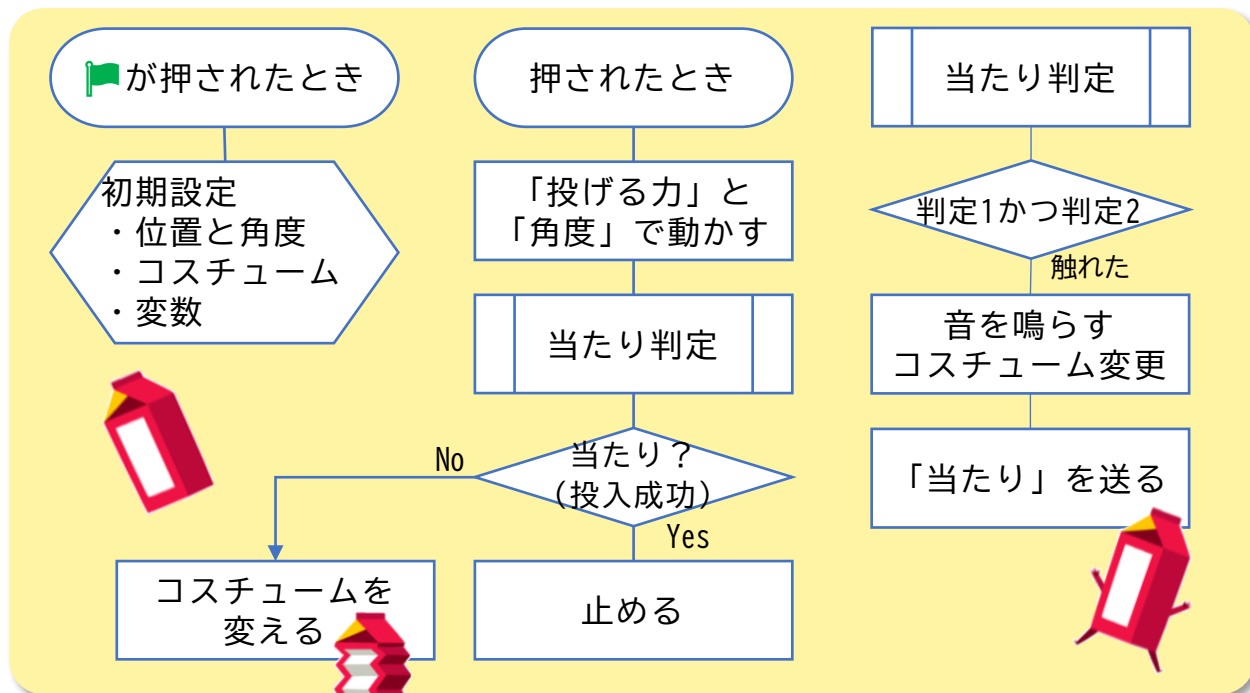


角度 0
投げる力 0
角度と投げる力を設定して豆乳をクリック

豆乳を投入!

スプライト
Milk
Glass Water

ステージ
豆乳を投入!
背景 1



戻る スプライトを選ぶ

検索

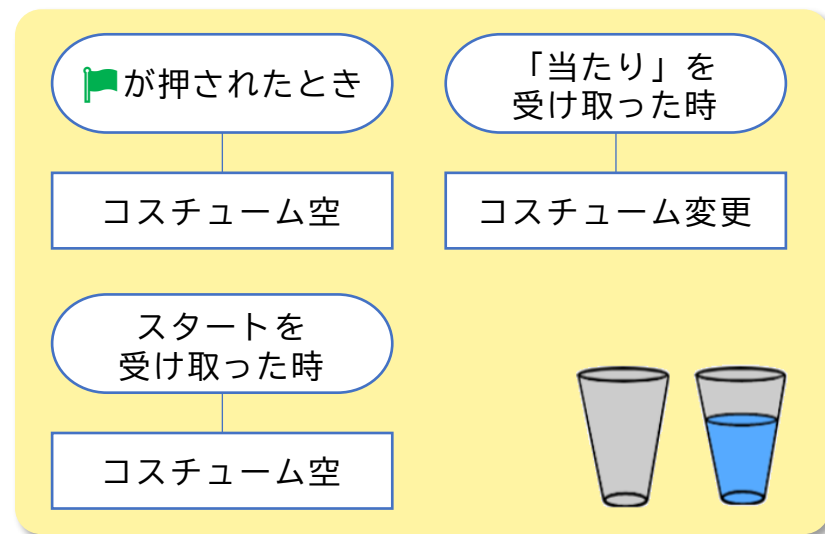
すべて 動物 人 ファンタジー ダンス 音楽 スポーツ 食べ物 ファッション

Glass Water Heart Candy Jar Milk Muffin Orange

変数

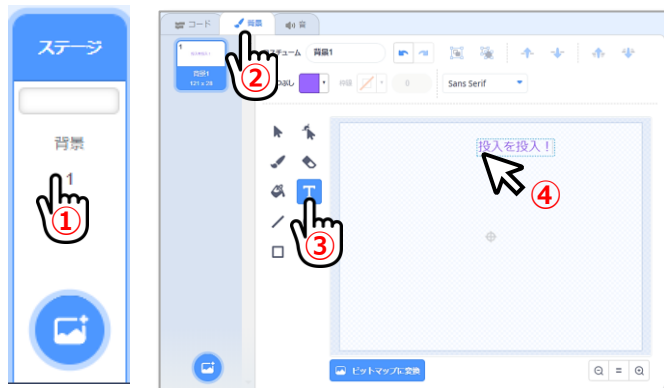
変数を作る

- IN1
- IN2
- 角度
- 重力
- 投げる力
- 当たり
- 落下値

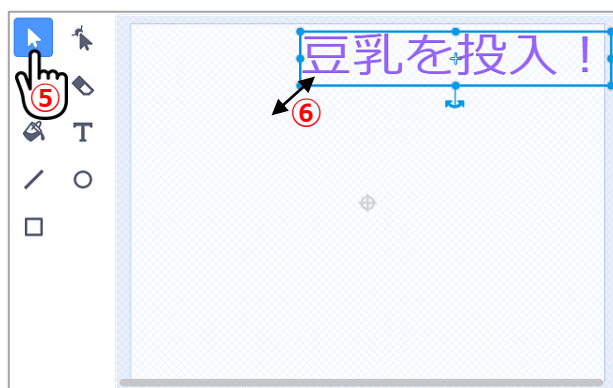


「背景」と「スプライト」を作る

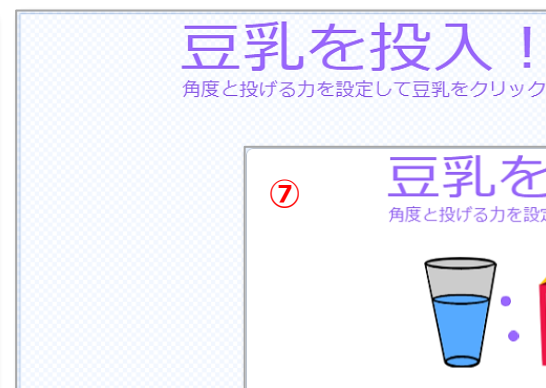
1. ステージの背景に文字を書く 「ゲームタイトル」と「操作の説明」
2. スプライトを作る 当たり判定用のポイントを2つ「判定1」「判定2」



- ①ステージ ②背景 ③テキストを選ぶ
④文字を入力する



- ⑤選択 ⑥文字の大きさを調整
③～⑥を繰り返し説明文も入力



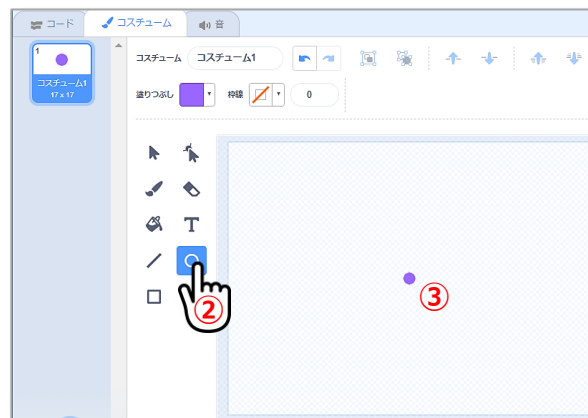
- ⑦完成したらステージで確認

ステージに写真を使ったり、スプライトにオリジナルのイラストをアップロードしたりできるよ。



- ①～④を2回繰り返し「判定1」と「判定2」を描く

- ②円 ③中心にドラッグで丸を描く



- ④スプライトの名前を変える「判定1」

- ①スプライトを追加、描く

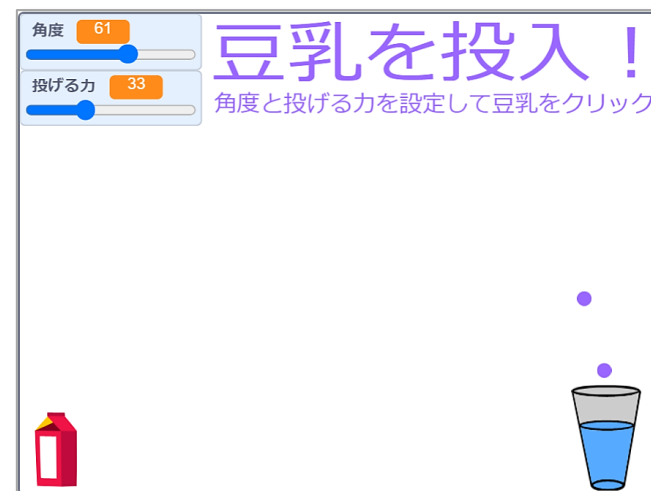


ステージのレイアウトや
必要なデータを考える

1. スプライトの大きさと位置を決める
2. 必要な変数を作る、入力方法を考える

～ユーザーインターフェース～
使う人の使いやすさを考える。
「回す」「押す」「喋る」
「タッチ」などがあるね

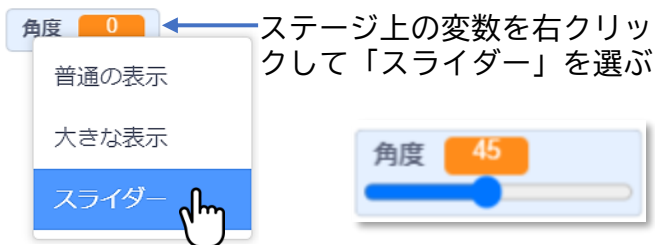
◆ 豆乳、コップをゲームに適した大きさにする スプライトをドラッグして図のように配置



◆ プレイヤーが設定するデータ、プログラムの中で必要なデータの変数を作る

- IN1 ... 当たり判定1用
- IN2 ... 当たり判定2用
- 角度 ... プレイヤー設定
- 重力 ... 難易度を変える
- 投げる力 ... プレイヤー設定
- 落下値 ... 落ちるY座標の内部計算用

- ① 変数を作る (ステージ上に表示される)
- ② プレイヤーが設定する角度と投げる力以外はステージから消す
- ③ 表示をスライダーにする
- ④ 変数の場所をドラッグで移動する



変数はいつでも追加したり削除したり、名前を変えたりもできるよ



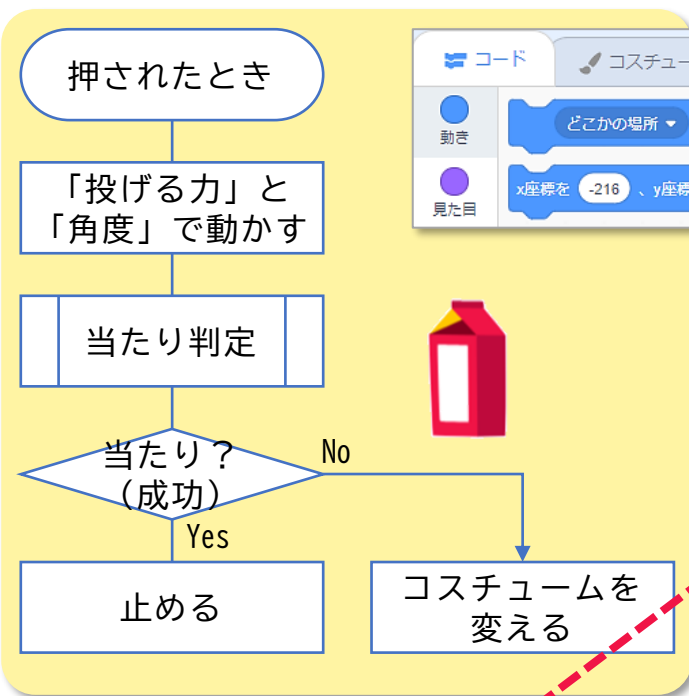
豆乳を投げる① 繰り返しと角度

プログラムを
テストする

1. 豆乳がクリックされたときに「角度」と「投げる力」で動かす

💡 テストを繰り返しながら徐々に仕上げていこう

◆ 最初に、クリックされたら動くプログラムを作る



①

クリックして
試してみよう

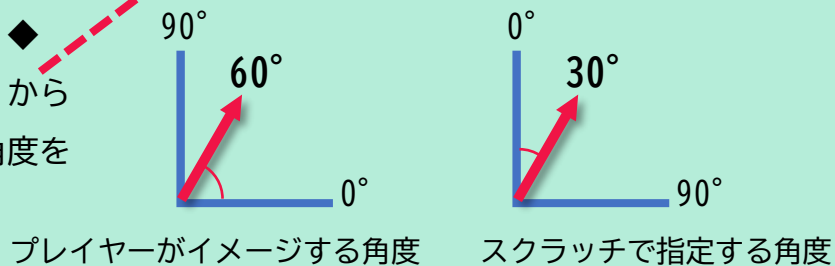
- ・放物線にならない
- ・角度が違う

②

- ・XY座標を変える
- ・「90-角度」にする
- ・端に触れるまで繰り返す
- ↓
- ・放物線にならない

③

◆ 「??° に向ける」 ◆
プログラムでは、90° から
プレイヤーが設定した角度を
引く必要があるね



・落下値を入れる
(P2放物線ロジック参照)

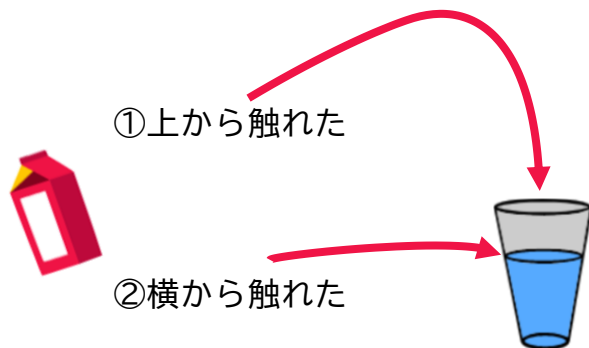
豆乳を投げる② 当たり判定

アルゴリズム
を考える

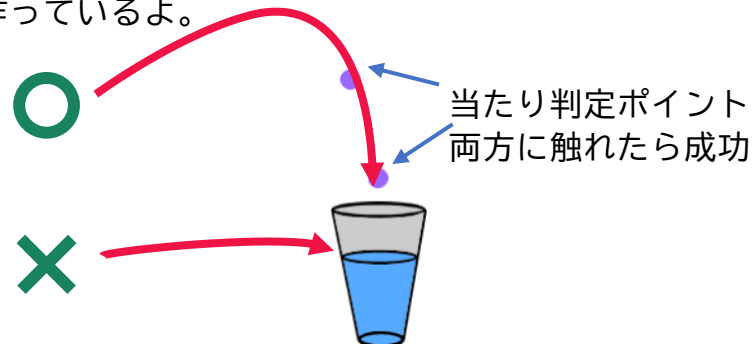
1. 投入成功か失敗かを判定する（豆乳がコップにぶつかったのか上から入ったか）
2. 両方に当てはまる「aで、さらにb」 どちらか片方でも当てはまる「aか、またはb」

◆ 豆乳がコップに触れたら「投入成功！」にしていかな？

```
もし Glass Water に触れた なら
```

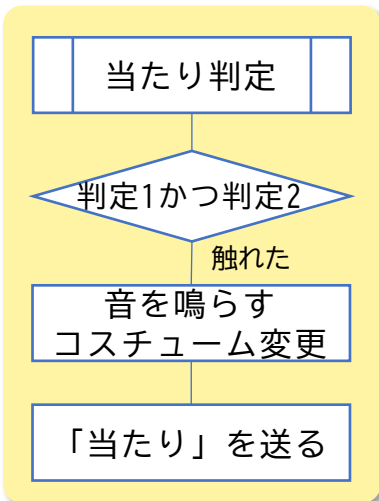


①を成功とするために当たり判定ポイントを使う。
スクラッチでは、たくさんの人がいろいろな「当たり判定」を作っているよ。



```
かつ  
または
```

◆ 当たり判定ポイント1に触れて、さらに2にも触れたら「投入成功！」にする

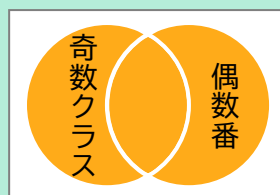
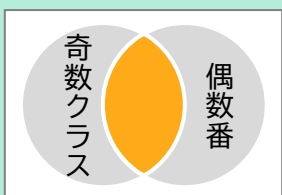


◆ 複数の条件で判断したいとき「論理演算(ろんりえんざん)」◆

- ・「かつ (AND アンド)」→ 両方に当てはまる
- ・「または (OR オア)」→ どちらか片方でも当てはまる

Q. 4年3組9番のひかるさんが当てはまるのは？

- ① 奇数クラス かつ 偶数番
- ② 奇数クラス または 偶数番



判定ポイント両方に触れたら「当たり」にするプログラムを作る

```
もし 判定1 に触れた なら  
  IN1 を 1 にする  
もし 判定2 に触れた なら  
  IN2 を 1 にする  
もし IN1 = 1 かつ IN2 = 1 なら  
  当たり を 1 にする
```

「当たり」変数
を作ろう

当たり

プログラムを機能で分ける

1. オリジナルの命令ブロックを作る

💡 機能ごとにプログラムを分けることでテストもしやすくなるよ

当たり判定 一般的には「関数」や「ファンクション」などという

◆ たくさんの処理があったり、複雑な処理があると、プログラムが長くなる。漏れやミスなどでバグにならないようにプログラムは簡潔にしよう。

◆ 「ブロック定義」の良いところ ◆
・プログラムを分けることができる
・プログラムに名前をつけられる
・プログラムを再利用できる



② 名前をつける
何を
するプログラムか
分かりやすい名前にしよう

プログラムを分ける



←つけた名前
↓ブロック定義の処理



←変数の初期設定をしよう

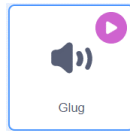
落下値を重力ずつ減らす
1回目の落下値... 0
2回目の落下値... -1
3回目の落下値... -2
4回目の落下値... -3

←ブロック定義を呼んで実行する

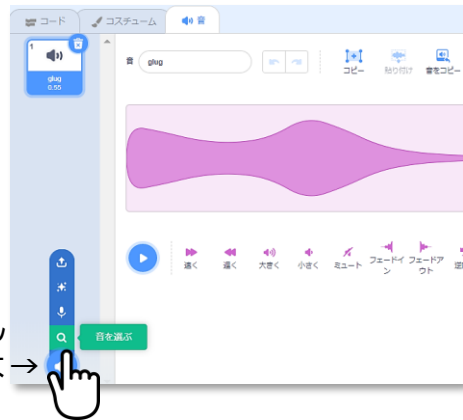
豆乳を投げる③ 成功/失敗を表現する

アルゴリズム
を考える

1. 音を選ぶ 音を鳴らす
2. コスチュームを変える



◆ 成功したら音を鳴らそう



```

このスプライトが押されたとき
  スタート を送る ← コップに知らせる
  コスチュームを milk-a にする ← コスチュームの初期化
  重力 を 1 にする
  ...
  
```

```

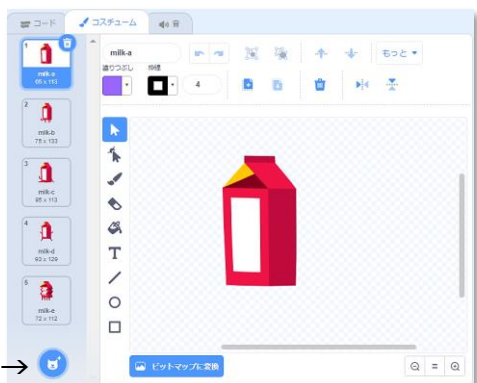
端 に触れた まで繰り返す
  投げる力 歩動かす
  y座標を 落下値 ずつ変える
  落下値 を 0 - 重力 ずつ変える
  
```

```

当たり判定
もし 当たり = 1 なら ← 成功なら 繰り返しを止める
  このスクリプトを止める
  コスチュームを milk-e にする ← 端に触れた時
  
```

用意されている音を自由に選んだり、録音することもできるよ

◆ 成功/失敗でコスチュームを変えよう あらかじめ用意されているコスチュームはスプライトによって違うよ



追加したり、自分で描くこともできるよ

```

定義 当たり判定
もし 判定1 に触れた なら
  ...
もし IN1 = 1 かつ IN2 = 1 なら
  当たり を 1 にする
  glug の音を鳴らす ← 音
  コスチュームを milk-d にする ← コスチューム
  90 度に向ける ← 向きと座標を 整えよう
  x座標を 200、y座標を -140 にする
  当たり を送る
  
```

```

が押されたとき
  コスチュームを glass water-b にする
  スタート を受け取ったとき
  コスチュームを glass water-b にする
  当たり を受け取ったとき
  コスチュームを glass water-a にする
  
```

試して難易度を調整する

1. スプライトを初期状態に戻す
2. 当たり判定を調整する

💡 ゲームが楽しくなる工夫をしてみよう！どんな工夫ができるかな

◆ 入力設定値をクリアし、スプライトをスタート状態に戻す

```
が押されたとき
  角度 を 0 にする
  投げる力 を 0 にする
  コスチュームを milk-a にする
  90 度に向ける
  x座標を -200、y座標を -150 にする
```

角度と投げる力のバランスを考えて、上手に豆乳を投入してね！



ゲームがスタートした時に喋らせるにはどうしたらいいかな？

角度と投げる力のバランスを考えて、上手に豆乳を投入してね！ と言う

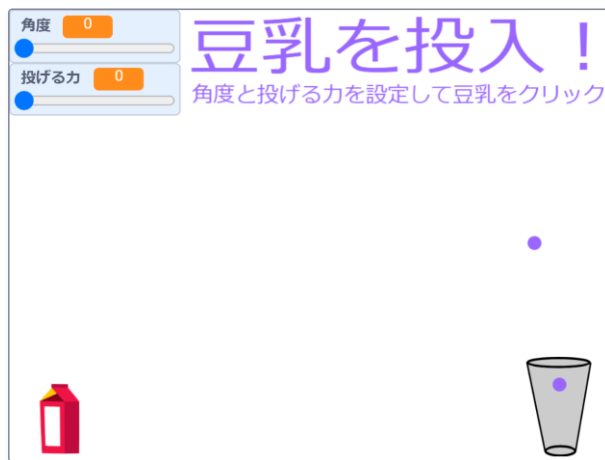
◆ コップにぶつかったプログラムを作ってみよう

```
もし Glass Water に触れた なら
  90 向き 度に向ける
  20 歩動かす
```

このプログラムをどこに入れたらいいかな？



◆ 当たり判定ポイントをドラッグで動かして難易度を調整しよう



プログラムを保存する

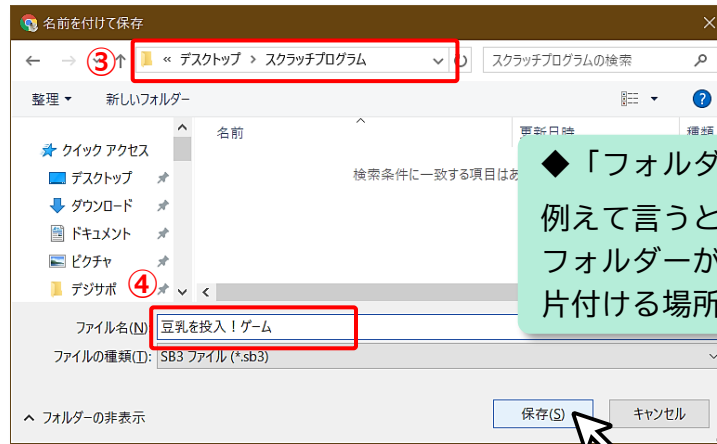
ファイル操作

1. プログラムをパソコンに保存する
2. プログラムをパソコンから読み込む

スクラッチプログラム

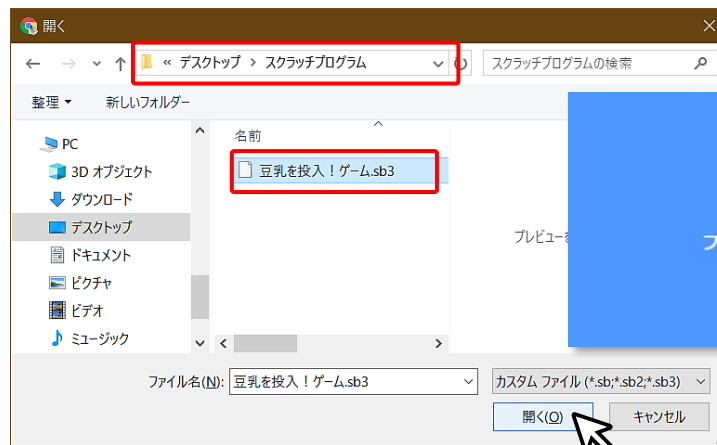
💡 パソコンのデスクトップに「スクラッチプログラム」というフォルダーを作ってもらおう

- ◆ ①メニュー「ファイル」→ ②「コンピューターに保存する」→ ③ 保存する場所を選ぶ → ④ 名前をつける → ⑤ 保存



◆「フォルダー」と「ファイル」◆
例えて言う「本棚」と「本」
フォルダーが本棚で、ファイルが本。
片付ける場所と片付ける物。

- ◆ ①「ファイル」→ ②「コンピューターから読み込む」→ ③ ファイルの場所を選ぶ → ④ ファイルを選ぶ → ⑤ 開く



豆乳を投入！ 全体のプログラム

が押されたとき

- 角度 を 0 にする
- 投げ力 を 0 にする
- コスチュームを milk-a にする
- 90 度に向ける
- x座標を -200、y座標を -150 にする
- 「角度」と「投げ力」のバランスを考えて上手に豆乳を投入してね！ と言う

重力を変えるとどうなるかな？

ブロック定義

変数

ブロックを作る

変数を作る

- IN1
- IN2
- 角度
- 重力
- 投げ力
- 当たり
- 落下値

定義 当たり判定

もし 判定1 に触れた なら

- IN1 を 1 にする

もし 判定2 に触れた なら

- IN2 を 1 にする

もし $IN1 = 1$ かつ $IN2 = 1$ なら

- 当たり を 1 にする
- glug の音を鳴らす
- コスチュームを milk-d にする
- 90 度に向ける
- x座標を 200、y座標を -140 にする
- 当たり を送る

このスプライトが押されたとき

- スタート を送る
- スタート と言う
- コスチュームを milk-a にする
- 重力 を 1 にする
- 落下値 を 0 にする
- 当たり を 0 にする
- IN1 を 0 にする
- IN2 を 0 にする
- x座標を -200、y座標を -150 にする
- 90 - 角度 度に向ける
- 端 に触れた まで繰り返す
- 投げ力 歩動かす
- y座標を 落下値 ずつ変える
- 落下値 を 0 - 重力 ずつ変える
- 当たり判定
- もし 当たり = 1 なら

 - このスクリプトを止める

- もし Glass Water に触れた なら

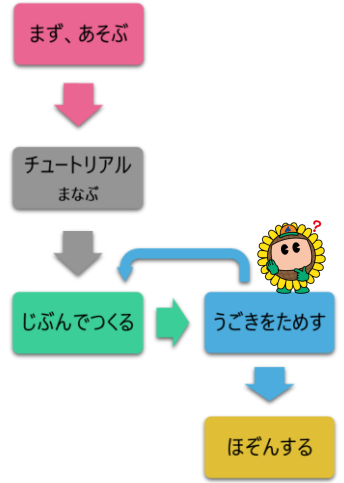
 - 90 - 向き 度に向ける
 - 40 歩動かす
 - コスチュームを milk-e にする



が押されたとき

- コスチュームを glass water-b にする
- スタート を受け取ったとき
- コスチュームを glass water-b にする
- 当たり を受け取ったとき
- コスチュームを glass water-a にする

コップ



これまでの作品【1】～【6】はみんなチャレンジしてみたかな？
 今回はすべてオンライン上でチャレンジしてもらったけれど、今度は教室で一緒に楽しもう！
 チャレンジしてみて、さらに学んでみたい！
 というあなたも
 途中までできたけど・・・、再挑戦したい！
 というあなたも

ひまわりネットワークのプログラミングの先生と一緒にスクラッチをマスターして、オリジナル作品も作れる「プログラミングマスター」になろう！

<保護者の方へ>2020年度から小学校必修化となったプログラミング教育は、知識や技能を習得するだけではなく、思考力・判断力・表現力を育み、生きる力を身に付けることを目的としています。

