



Girls Who Code At Home

流星キャッチャーゲーム:パート 3

流星を落下させよう

学習概要

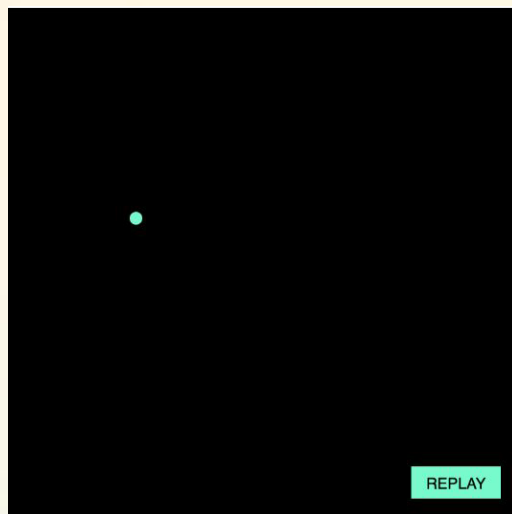
パート2の最後に、座標系を使ってゲームの最初の構成要素である流星を描きましたね！そして、流星の色とスケッチの背景を設定しました。このパートでは、p5.js で変数を作成して使用し、指定された速度で画面上の流星を移動させる方法を学びます。あなたは、変数と $+$ や $=$ のような算術演算子を組み合わせ、動きをシミュレートします。まるで魔法のようですが、実は簡単な数学です。[ここ](#)をクリックすると、このアクティビティが終了するまでに学習する内容をプレビューできます。

なお、このアクティビティに取り掛かる前に、「流星キャッチャーゲームシリーズ」の[パート1](#)と[パート2](#)を終わらせてください。

学習目標

このアクティビティが終わるころには、次のことができるようになります。

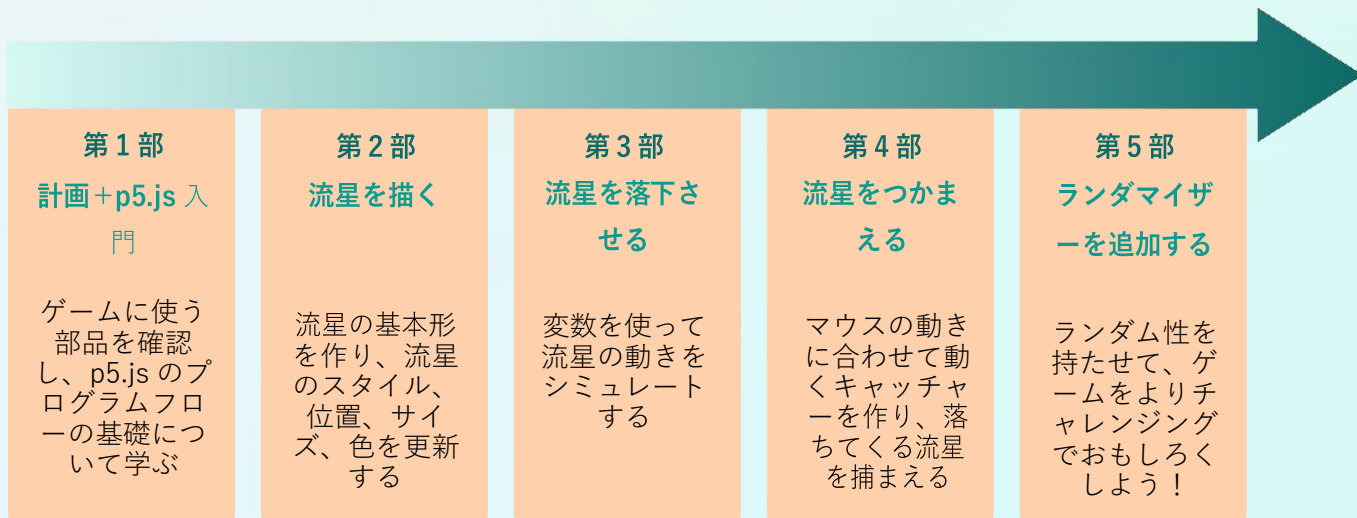
- 基本的な動きをプログラムでシミュレートする方法を説明する。
- 変数と算術演算子を使って、さまざまな動作を要素にプログラムする。



注：流星の挙動をリセットできるように、REPLAY ボタンを設けました。REPLAY ボタンが設置されていないと、流星が画面の底まで落下した時に、黒い画面だけの表示になってしまいます。次回のアクティビティ：パート4では条件文を使い、この挙動を修正する予定です。

マテリアル

- ➔ [p5.js オンラインエディタ](#)
- ➔ [流星キャッチャーゲーム サンプルプロジェクト](#)
- ➔ [流星キャッチャーゲーム その3 リファレンスガイド](#)



流星キャッチャーゲームの[動画チュートリアル](#)のパート3を参照しながら、進めることもできます。

Women in Tech スポットライト：ロビン・フニッケ



カリフォルニア大学サンタクルーズ校で教鞭をとるゲームデザイナーで、Funomena の共同設立者でもあります。エレクトロニック・アーツでリードデザイナーとしてキャリアをスタートし、MySims のデザインを担当しました。その後、独立系ゲーム会社の thatgamecompany で、[Journey](#) というゲームの制作チームに参加した2人の女性のうちの1人です。Journey は Game of the Year を複数受賞し、2013 年グラミー賞の Best Score Soundtrack for Visual Media にノミネートされたこともあります。

画像の出典はこちら: [UCSC DAMN](#)

ロビンは [Funomena](#) のチームとともに、バーチャルリアリティゴーグルを含むあらゆるプラットフォームを活用したビデオゲームの制作を開始しました。彼女のチームは『Luna』や『Woorld』などの実験的なゲームを

制作しています。バーチャルリアリティ（VR）ゲームの制作における利幅は小さいものの、ロビンはリスクを負って自分の創造性を押し出すようなゲームを開発することが重要だと考えています。2008 年、ロビンは Gamasutra の「ビデオゲーム業界で働く女性トップ 20」に選ばれ、2009 年にはマイクロソフトの「ゲーミング・アワード・フォー・デザイン」を受賞しています。

ロビンは、ゲーム業界の多様性を支持しています。彼女の活動は主に、Underrepresented group(少数派であったり過小評価されているグループ)の作品や声を増幅させることを中心に展開されています。カリフォルニア大学サンタクルーズ校の教授として、ゲームデザインのためのアートとプログラミングの両コースを組み合わせたプログラムを学生に提供することが、彼女の仕事の大部分です。

この[ビデオ](#)では、ロビンについて、また彼女がゲーム業界で前向きな力を発揮するためにどのように活動しているかをご紹介します。また、[ロビンの略歴](#)や、AR ゲーム「[Woorld](#)」、ゲーム「[Journey](#)」や彼女のその他のプロジェクトを通じて、ロビンをより知ることができます。

考えてみましょう

コンピュータサイエンティストであることは、単にコーディングが得意というだけではありません。ロビンや彼女の仕事が、偉大なコンピュータサイエンティストが身につけようとしている強み、すなわち勇気、回復力、創造性、目的にどのように関係しているのか、時間をかけて考えてみてください。



CREATIVITY

Robin はどのように予想と異なる方法でゲームにアプローチしているのですか？予想外の方法でプロジェクトに取り組むことのメリットは何でしょうか。

あなたの回答を家族や友人と共有しましょう。他の人にもロビンについてもっと読むように勧めて、議論に参加してもらいましょう！

ステップ 1: p5.js で変数を使う（5 分～10 分程度）

JavaScript の変数のおさらい (3-5 分)

本題に入る前に、変数について簡単におさらいしておきましょう。変数とは、コンピュータ・プログラムの中で情報（データ）を格納するための容器です。プログラム中に変数の値を簡単に変更できるため、特に強力です。

変数を作るには、まず宣言する必要があります。これはプログラムに容器を作ることと、それに名前をつけることを伝えるものです。

JavaScript では次のように変数を宣言します：`let meteorDiameter;`

`let meteorDiameter = 50;` のように宣言すると同時に初期化（初期値を代入）することができます。では、構文を分解してみましょう。

JAVASCRIPT	DESCRIPTION
<code>let meteorDiameter = 50;</code>	<ul style="list-style-type: none">→ <code>let</code>: スケッチに変数を作成するように指示するキーワード→ <code>meteorDiameter</code>: 変数の名前です。この名前は何でもかまいませんが、必ず説明的な名前にしましょう！1 語しか使えないので、<u>キャメルケース</u>を使用します。→ <code>=</code>: 変数に値を代入します。初期化ともいいます。→ <code>50</code>: 現在変数に格納されている値。変数には、数字、文字、文字列など、どのような種類のデータでも格納することができます。→ <code>;</code>: p5.js のコード行はすべてセミコロンで終わらなければなりません。

プログラマーは通常、必要となるすべての変数を先頭で作成し、定義します。これらはグローバル変数と呼ばれます。これは、これらの変数がコードのどこでも使えることを意味します。また、プログラマーにとっても、コードを読む人にとっても読みやすいコードになります。変数についてもっと詳しく知りたい方は、[Coding Train のこのビデオ](#)をご覧ください。

変数の追加 (3-5 分)

現時点で、`ellipse` には変数が含まれていません。もし、動きをシミュレートしたいのであれば、この静的な値を変数に置き換えて、`x` と `y` の位置の値を時間と共に変更できるようにしなければなりません。`ellipse` の各パラメータ (`x`、`y`、幅、高さ) に対して変数を宣言し、初期化する必要があります。

変数に名前をつけましょう。このテキストで使っている変数名を使ってもよいですし、ご自身でオリジナル変数名を使っても構いません。オリジナルの変数名を使用する場合は、後で、その変数名を正しく参照することをわすれないでください。

`setup()` 関数の上に、以下の変数を追加します。

- ☐ `x` の位置を格納する変数を作成し、その値に `200` を代入します。テキストでは、この変数の名前を `meteorX` としますが、独自の変数名を作ることができます。
- ☐ `y` の位置を格納する変数を作成し、値に `0` を代入します。テキストでは、この変数の名前を `meteorY` としますが、独自の変数名を作ることができます。
- ☐ 幅と高さを格納する変数を作成し、値として `20` を代入します。流星は円形なので、これらの値は同じになります。テキストでは、この変数に `meteorDiameter` と名付けましたが、独自の変数名を作っても構いません。

さて、変数ができたので、使ってみましょう。`ellipse()` 関数の中で、次のパラメータについて、数値を先ほど作成した変数に置き換えてください。

- ☐ `x` 位置
- ☐ `y` 位置
- ☐ 幅と高さ



「リファレンスガイド」の p.2 でコードを見直すことを忘れないでください。

ステップ 2：目標の動きについて考えてみましょう（5～10 分）

このパートでは、流星を画面上部から画面下部へ落下させることを目標としています。

しかし、それをどのようにコードに反映させればよいのでしょうか？

それを理解するための例を考えてみましょう。

60 秒から 90 秒かけて、この [スケッチ](#) で、左から右へ動く円の動きを観察してください。

そして、以下の質問について考えてみてください。

- 円はどのような軸で動いているのでしょうか？
- 円の位置はどのように変化しているのでしょうか？ `ellipse()` の中のどの値を変更すれば、このようなことが起こるのでしょうか？
- これを実現するコードは `setup()` または `draw()` の中にあるのでしょうか？



あなたの観察をもとに、ボールをどのように動かすかをプログラムに指示する疑似コードを書いてください。書き終えるまで次のパートに進んではいけません。



「リファレンスガイド」の p.2 で自分のアイデアを見直すことを忘れないでください。

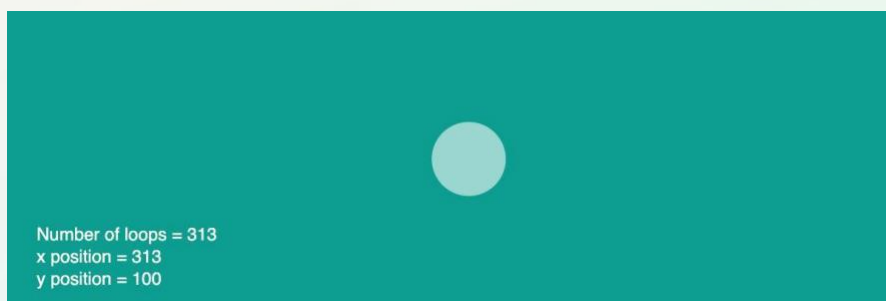
Step 2: Make observations about motion (cont.)

動きをコードで記述する

水平方向の動きをシミュレートするためには、プログラムがループするたびに x の値を変化させたいですね。このプログラムは、ループで動作することを忘れないでください。プログラムは、`draw()` 内のすべてのコードを順番に実行します。プログラムの最後に到達すると、先頭に戻り、また最初からやり直します。これは、あなたが停止を指示するまで永遠に続きます。 x の値を自分自身に数字を足したものに設定することで、動きを一行のコードとして書くことができます。

```
ellipse(xPosition, yPosition, 50, 50);  
xPosition = xPosition + 1; // xPosition++ と書くこともできます。
```

これは、プログラムがループするたびに x の値がその数（上の例の場合は 1）だけ増加することを意味します。この数値は、流星が画面上をどれだけゆっくり、あるいは速く移動するかを決定します。言い換えれば、速度を設定することになります。この [スケッチ例](#) では、スピードを 1 に設定し、ループするたびに x の位置が 1 ずつ増加するようにしています。

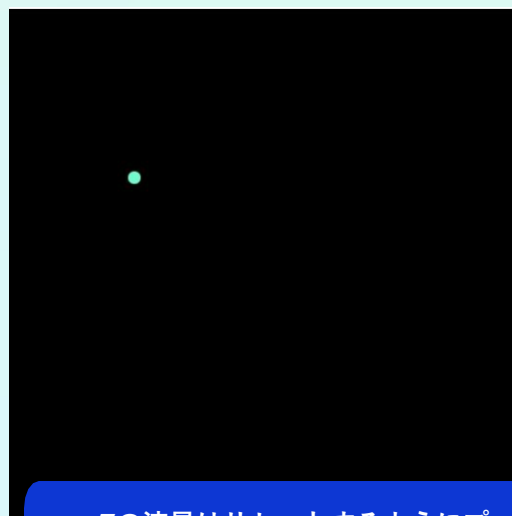


ステップ 3：流星に動きをつける（5 分～10 分）

これまでの例では、円は水平方向に動きましたが、下のスケッチに示すように、流星を y 軸方向に垂直に移動させたいと思います。これは、 x 位置の代わりに y 位置を増加させる必要があることを意味します。[このリンク](#) をクリックすると、 y 軸で動く動作をプレビューできます。

以下の手順で流星を落下させることができます。

- ❑ `setup()` 上に新しい変数を作り、速度を格納します。テキストでは、変数に `speed` という名前をつけましたが、好きな名前をつけてください。ただ、後ほどコードの中で正しく参照できるようにしてください。
- ❑ 流星がゆっくりと落下するような値を割り当てる
ヒント: 小数も使えますよ！



この流星はリセットするようにプログラムされていますが、パート 5 まで終了すると、あなたのスケッチもリセットできるようになります。

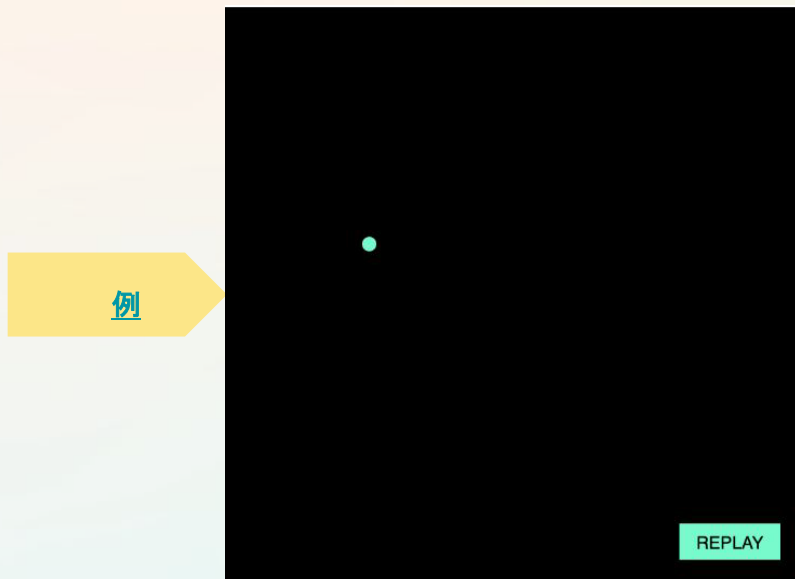
- `draw()` 関数の中に、流星の y 位置を変えて、画面の上から下に落ちるようにするコードを追加してください。

「リファレンスガイド」の p. 3 でコードを確認することを忘れないでください

ステップ 4: コードのテスト (5 分)

ここまでに書いたものをテストして、プログラムが思い通りに動くことを確認しましょう。REPLAY ボタンをクリックして、スケッチを実行してください。現時点で、あなたの環境は以下のようになっているはずです。

- 流星が画面上部から画面下部へゆっくりとした速度で落下する。
- 星は一番に到達後、下に消えてしまう。
- REPLAY ボタンは存在していません。



注：私たちの例では、流星の挙動をリセットできるように、REPLAY ボタンを設けました。REPLAY ボタンを入れない場合、流星が画面の下端を超えた後は、黒い画面だけが表示されたままになります。

次回のアクティビティ：パート 4 では条件文を使い、この挙動を修正する予定です。

思い通りに動かない？デバッグのコツをご紹介します。

- あなたのコードは正しい波括弧の中にありますか？
- コードの各行の末尾にセミコロンをつけていますか？
- 変数名や関数名のスペルは正しいですか？
- 関数は正しい位置と順序で配置されていますか？
プログラムの流れは順序が重要であることを忘れないでください。
- 算術演算子は正しい位置にありますか？
- 速度変数の値が速すぎたり（高い値）、遅すぎたり（低い値）していませんか？
- 流星は上から下へ落ちていますか？流星の y 位置変数を更新しましたか？

デバッグのベストプラクティスについて再確認したい場合は、
p5.js コミュニティの [素晴らしい記事](#) をご覧ください。

ステップ 5: 理解度の確認

流星を画面の下から上に移動させるには、速度式をどのように変更すればよいのでしょうか？



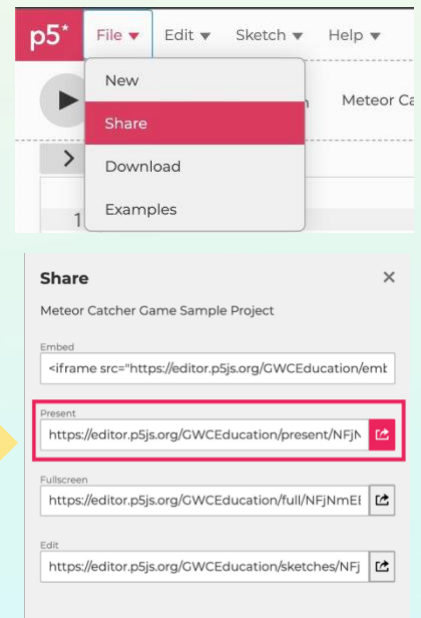
リファレンスガイド」の P.3 で自分のアイデアをレビューすることも忘れないでください。

ステップ 6 : Girls Who Code at Home プロジェクトを共有しよう (5 分)

私たちはあなたの作品を見るのが大好きですし、他の人もそうであることを知っています。あなたのゲームを私たちとシェアしてください。また、[@girlswhocode](#) [#codefromhome](#) タグを忘れないでください。私たちのアカウントで紹介するかもしれません。

以下の手順でプロジェクトを共有することができます。

- 最初にプロジェクトを保存してください。
- **File** メニューのドロップダウンメニューから、「**Share**」を選択します。
- ドロップダウンメニューから「**Link**」を選択します。
- **Present** リンクをコピーして、共有したい場所に貼り付けてください。



Project Link

今後の Girls Who Code at Home のプロジェクトにご期待ください！