

理数科オリエンテーション STEAM 指導案

1. 単 元 名 Processing 入門～プログラミングで数学アートを作ろう～
2. 日 時 令和8年4月24日（金）第5～7限
3. 対 象 1年3組40名（理数科1年次生徒）
4. 場 所 川口市立高等学校 小ホール
5. 本時の目標
 - (1) 簡単なプログラミングから原理を理解し、アルゴリズム的な構造に親しむ
 - (2) 数式やルールから、アートの構造を作ることができることを理解し、理数的な考え方（規則性・変数・関数・パラメータ調整）が、表現活動にも生きることを体験する
 - (3) グループ制作を通して、共同して研究する力を育成すると共に制作内容について初日に実施した科学プレゼンテーション講座の内容を踏まえプレゼンテーションを行う

3. 展開

| 過程 | 学習活動 | 指導上の留意点 |
|--------------|--|--|
| 導入 (5分) | Processing とは何か ・2001年にMITのK. リースとB. フライにより開発されたアートから生まれたプログラミング環境で「数学やデータの視覚化」に特化されておりアーティストや研究者が使う →アーティストやデザイナーが躊躇なく着想を形にできるプログラミング環境 コードによる表現を誰でも取り入れられるように設計されている ・数字やルールでアートを作る道具で数学・美術・情報につながる STEAM ツール https://processing.org/ から各自ダウンロード | |
| 展開1 (10分) | Hello world-円を描こう ellipse 関数で円を描いてみる パラメータを変えて、どのように変化するか実験させる (5分) (x, y, 幅, 高さ) 左二つは円の位置 (左上の座標)、右二つは円の大きさに関わる 座標ルールの説明: 左上が原点、下がプラス・・・最初の二つは左上からのピクセル値 画面の大きさを決められるコマンドの紹介 | ellipse(25, 25, 50, 50); 行の最後に必ずセミコロン; をつける |
| 展開2 (10分) | 画面の仕組み-基本コマンドでさらにアレンジ ① size(width, height); : 画面の大きさを決める=画面サイズ ② background(R, G, B(color)); : 背景を塗る=背景色 ③ stroke(R, G, B(color)); : 線の色 ④ strokeWeight(pixel); : 線の太さ ⑤ fill(R, G, B); : 色を塗る これらを使ってコードをアレンジし、どのようになるか実験させる (5分) 複数の円を描くためにはどうすれば良いか? | size(300, 300); fill(0, 0, 255); ellipse(200, 200, 100, 100); ellipseをrect, triangleに 変えてみる |
| 展開3 (15分) | 一歩進んだ応用コード - 繰り返しの for 文・動きの入れ方=setup 関数と draw 関数 ・10個の円を描くために10個の文は必要ない →繰り返しはコンピュータの得意技: アルゴリズムの強み ・もし動かしたいなら、セット (setup / draw) で使うのがルール | Size(500, 500); for(int i=0; i<10; i++){ ellipse(i*50, 300, 40, 40); ※int = 整数 ●毎フレーム動く (draw) |

| | | |
|-----------------------|--|--|
| | <p>→状態 : setup でキャンバスを用意 (最初に1回だけ実行) し、draw が 1秒間に 60 回ループし続ける=Processing におけるプログラムの基本構造</p> <p>コードは主に、setup()関数と draw()関数の2つを記述するところから始まる</p> <p>初期値の固定で setup (初期条件) と draw (時間の経過) に分ける</p> <p>※void=命令を実行するだけ:結果は返さない、一連の動作のパッケージ</p> <p>・軌跡アート</p> <p>三角関数の導入で、より複雑な動きを作ることができる</p> <p>float = 実数 (浮動小数点)、なめらかな動きにつなげられる</p> <p>これまでの int 型との違い</p> | <pre>int x=0; void setup() { size(600, 400); } void draw() { background(255); ellipse(x, 200, 50, 50); x=x+2;} float t = 0; void setup() { size(400, 400); background(255);} void draw() { float x = 200 + 150 * cos(3 * t); float y = 200 + 150 * sin(5 * t); ellipse(x, y, 2, 2); t += 0.01;}</pre> |
| <p>展開 4 (45分)</p> | <p>グループで協働して Processing アート作成</p> <p>random()関数 : 指定した範囲内でランダムな数を生成し、それを使って表現する →0 から指定した数の間でランダムな数を返す</p> <p>日本語メモ入りスターターコード : これを各チームでパラメータを変えていく</p> <pre>void setup() { size(800, 600); // 画面の大きさを決める (横 800, 縦 600) background(0); // 最初に背景を黒でぬる noStroke(); // 図形のふち線をなくす} void draw() { fill(0, 20); // 少し透明な黒を指定する rect(0, 0, width, height); // 画面全体にうすい黒い四角を重ねる // → 前に描いた図形が少しずつ消えて、残像っぽく見える float x = width/2 + sin(frameCount * 0.05) * 200; // x座標を決める // width/2 は画面の真ん中、 sin(frameCount * 0.05) で左右にゆれる動きを作る // *200 で動く幅を大きくしている float y = height/2 + cos(frameCount * 0.05) * 200; // y座標を決める、height/2 は画面の真ん中 //cos(frameCount * 0.05) で上下にゆれる動きを作る //x と組み合わせると円をえがくように動く float s = random(10, 30); // 円の大きさをランダムに決める、10~30 の間で毎回少しずつ変わる fill(random(100, 255), 150, 255, 150); // 円の色を決める、1つ目の値 (赤成分) だけをランダムにしている</pre> | <pre>random(50);0から50未満の 数値の中でランダムな表現 ができる random(30, 70);30 から 70 未満の数値の中でランダム な表現ができる</pre> |

| | | |
|------------------------|---|---|
| | <p>// 最後の 150 は透明度で、少しすけるようにしている</p> <pre>ellipse(x, y, s, s);</pre> <p>// (x, y) の位置に円を描く、横の大きさも縦の大きさも s なので正円になる</p> <p>4人1組でチームを編成</p> <p>以下の条件を示し、創作を開始</p> <p>課題：チームで1つの「動くアート作品」を作成せよ。</p> <p>条件</p> <ol style="list-style-type: none"> 1. 数式要素を1つ以上入れる（例：sin, cos, 比例, 回転, 規則的増減） 2. ランダム要素を1つ以上入れる 3. チームの作品テーマを決める 4. 作品タイトルをつける <p>以上の条件をもとに、作品プレゼンができるように協働して制作を行う</p> | <p>・チームで相談するときの 問い</p> <p>どの数字を変えると、どんな 見た目になる？</p> <p>ゆっくり動くほうが合う？ はやいほうが合う？</p> <p>明るい色と暗い色、どちらが テーマに合う？</p> <p>軌跡を残したい？ すぐ消し たい？</p> <p>丸がいい？ 四角がいい？</p> |
| <p>展開 5 (20 分)</p> | <p>作成した作品のプレゼンテーション</p> <p>発表内容</p> <ol style="list-style-type: none"> i) 作品タイトル ii) テーマ iii) 数学・規則の工夫 iv) 一番苦勞した点 v) 見てほしいポイント | <p>Art にただ一つの正解はな いことの確認</p> |
| <p>まとめ</p> | <p>数学(M)は、科学(S)や技術(T)、工学(E)の基礎であるだけでなく、美しい表現(A:芸術)を生み出すための「言語」であることを体感できたか</p> | |



川口市立高校STEAM講座

Processing入門

～プログラミングで数学アートを作ろう～

この時間にやること

- Processingかんたん入門（40分）
- チームでプログラミングアート制作
（テンプレコードを改造：45分）
- チームで作品をプレゼン（20分）

Processingとは何か

- 2001年にMITのK.リースとB.フライにより開発された、アートから生まれたプログラミング環境
- 「数学やデータの視覚化」に特化されており、アーティストや研究者が使用する
- コードによる表現を誰でも取り入れられるように設計されている
- 数学・美術・情報がつながるSTEAMツール

Processingとは何か

- <https://processing.org/>
から各自ダウンロード
- ClassroomからDL

Hello world - を描こう

- `ellipse(25,25,50,50);`

Hello world - 円を描こう

- `ellipse(25,25,50,50);`

この4つの数字を色々変えて、
どうなるか実験してみよう！

Hello world - 円を描こう

基本的な関数の使い方と、座標の概念を理解しよう

- コマンド : `ellipse(x, y, 幅, 高さ);`
- 座標ルール : 左上が原点(0,0)
右方向がプラス、 下方向がプラス
- 注意点 : 行の最後に必ずセミコロン「;」をつける

Hello world - 円を描こう

画面の大きさ変えられないの・・・？

No！変えられる

基本コマンドを使おう

- `size(width, height);` 画面の大きさを決める
- `background(R,G,B(color));` 背景を塗る
- `stroke(R,G,B(color));` 線の色
- `strokeWeight(pixel);` 線の太さ
- `fill(R ,G, B);` 色を塗る

基本コマンドを使おう

R,G,B=

レッド、グリーン、ブルー

この数値の混ぜ具合で色

基本コマンドを使おう

```
size(300,300);
```

```
fill(0,0,255);
```

```
ellipse(200,200,100,100);
```

基本コマンドdeアレンジ

```
size(300,300);
```

```
fill(0,0,255);
```

```
ellipse(200,200,100,100);
```

このコードを色々変えてみよう

基本コマンドdeアレンジ

```
size(300,300);
```

```
fill(0,0,255);
```

```
ellipse(200,200,100,100);
```

ellipseをrectに変えてみたら？

基本コマンドdeアレンジ

```
size(300,300);
```

```
fill(0,0,255);
```

```
ellipse(200,200,100,100);
```

```
triangleに変えてみたら？
```

応用コード - 繰り返し返しと動き

ex.円を10個描くには？

繰り返しのfor文

for文：

繰り返しはコンピュータの得意技

アルゴリズムの強みを活かす

繰り返しのfor文

```
size(500,500)  
for(int i=0; i<10; i++){  
  ellipse(i*50,300,40,40);  
}
```

繰り返しのfor文

※ int = 整数

0 ~ 9 までの整数

繰り返しのfor文

3分間で色々

いじってみよう！

もし動かしたいなら

setup / draw

のセットで使う

もし動かしたいなら

【Processingの基本構造】

setup() : 最初に1回だけ実行
(キャンバスの準備)

draw() : 1秒間に60回ループ
(アニメーションの作成)

もし動かしたいなら

コードは主に、**setup()**関数と**draw()**関数の2つを記述する所から始まる

初期値の固定で**setup**（初期条件）

draw（時間の経過）に分ける

```
int x=0;  
void setup(){  
  size(600,400);  
}  
void draw(){  
  background(255);  
  ellipse(x,200,50,50);  
  x=x+2;}
```

もし動かしたいなら

※void=命令を実行するだけ

結果は返さない、一連の動作の

パッケージ

色々いじってみよう

数学の活用

三角関数（ \sin , \cos ）を導入することで、より複雑で滑らかな動きが作れる
=軌跡アート

数学の活用

float = 実数(浮動小数点)

**なめらかな動きにつなげ
られる**

```
float t = 0;
void setup() {
  size(400, 400);
  background(255);
}
void draw() {
  float x = 200 + 150 * cos(3 * t);
  float y = 200 + 150 * sin(5 * t);
  ellipse(x, y, 2, 2);
  t += 0.01;}
```

色々いじってみよう！

グループで協働して Processingアート作成

- ・ もう一個だけ関数を
⇒ `random()`関数

グループで協働して Processingアート作成

指定した範囲内でランダムな数を生成し、それを使って表現する

→0から指定した数の間でランダムな数を返す

グループで協働して Processingアート作成

```
random(50);
```

0から50未満の数値の中で
ランダムな表現ができる

グループで協働して Processingアート作成

```
random(30, 70);
```

30から70未満の数値の中
でランダムな表現ができる

課題：

チームで1つの

「動くアート作品」

を作成せよ

1. 数式要素を1つ以上入れる
(sin, cos, 比例, 回転, 規則的増減)
2. ランダム要素を1つ以上入れる
3. チームの作品テーマを決める
4. 作品タイトルをつける

以上の条件をもとに、作品プレゼンができるように協働して制作を行う

作成した作品のプレゼンテーション

発表内容

- i) 作品タイトル
- ii) テーマ
- iii) 数学・規則の工夫
- iv) 一番苦勞した点
- v) 見てほしいポイント