氏名　本間大一優

PeriDot/Peree/PeriWorld
詳細説明

目次

1.PeriDot

PeriDotは教育向け、初心者向けにデザインされたビジュアルプログラミングからテキストプログラミングへのステップアップを目的としたシンプルなプログラミング言語です。そのため、教育者や学習者をサポートする機能が随所に埋め込まれています。

シンプル,直感的,覚えやすい

PeriDotの言語設計は「シンプル,直感的,覚えやすい」という３つの要素をベースにして行われています。そのため、言語としての機能はプログラミングにおいて重要となる「逐次,繰り返し,分岐処理」を学ぶために必要な機能に限定しました。結果、構造体やポインタなどの初心者には難しい機能は存在せず、ひたすらに「シンプル,直感的,覚えやすい」を追求した言語設計となっています

氏名　本間大一優

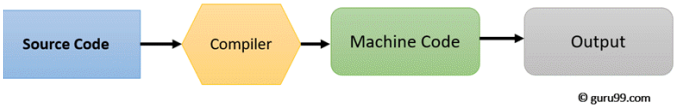
1-1.基本構造

インタプリタ型

プログラミング言語の実行方式は主にインタプリタ型(図1-1)、コンパイル型(図1-2)に分けられます。インタプリタ型ではインタプリタがコードを解析及び評価をし、ステップバイステップでプログラムを実行します。コンパイル型ではコンパイラがプログラムを機械語に翻訳し、その機械語を実行することでプログラムを実行します。つまり、コンパイル型では２つのステップが必要になり複雑です。PeriDotは教育向けの言語であるため、実行方法は簡潔でシンプルであるべきと考え実行方式はインタプリタ型を採択しました。

図1-1

図1-2



手続き型

現在、プログラミングパラダイム(プログラミングにおける考え方)は、主にオブジェクト指向と手続き型とに分けられます。オブジェクト指向は、個々のプログラムを一つ一つのオブジェクトとして捉え、それらが相互作用することで処理が進みます。オブジェクト指向は、オブジェクトの再利用や継承などが可能であるため便利な反面、その分覚えることも多く複雑になり、学習コストが高くなっています。それに対し手続き型は、個々のプログラムを手続き単位で捉え、最初から順番にその手続きを踏むことで処理が進みます。そのため、PeriDotではプログラミングパラダイムに基本シンプルな記法で、処理過程の読み取りも容易な手続き型を採択しました。

動的型付け

プログラミングの型システムには2種類あります。１つ目はコードを書くときにプログラマが型を決める静的型付け、２つ目はコンパイラやインタプリタが実行時に判断して型を決める動的型付けです。静的型付けには処理が高速になりメモリが最適化されるなどのメリットがありますが、コードを書く際に変数の型について留意する必要があり、型に関する知識も必要となります。PeriDotはプログラミングの根底となる逐次,繰り返し,分岐処理を簡単に学ぶことを目的としているため、メモリ最適化や処理の高速化より機能の簡略化を優先し動的型付けを採択しました。

開発言語

Go言語はGoogleが開発した静的型付け,コンパイル言語で高速な動作が可能です。サーバーサイドやコマンドラインツールなどで主に使われています。言語選定の際、PeriDotをweb上で実行可能にするwebサービス(Peree)についても考慮し、Pereeのサーバーサイド言語とPeriDotの開発言語を統一し開発効率を上げるためにGo言語を採択しました。また、インタプリタ内の字句解析器,構文解析器,意味解析器の基礎構造は「Go言語でつくるインタプリタ(著者:Thorsten Ball , 翻訳:設楽洋爾,出版:オライリー・ジャパン)」を参考に作成しました。

氏名　本間大一優

1-2.文法設計

直感的でシンプルな文法

1.予約語の削減

従来の言語は広く普及することを目的に様々な機能が盛り込まれ、それに伴い予約語の数が多くなっていました。予約語の多さは覚えるべき単語の多さに直結し、入門のハードルをあげてしまいます。そこで前述した機能の削減とともに予約語の数も極力減らしました。ちなみに、教育現場で主流のテキストプログラミング言語であるJavaScriptとPeriDotを比較してみると、予約語の数は約1/3に削減されています。また繰り返し処理を行う予約語はloopに、分岐処理はifに統合するなどしてソースの統一性を実現しています。これにより教育者や初心者によるソースコードリーディングが容易になります。

Java Script
Total:38

if
else
while, for
break
var
false
true
function
return
case await catch class const continue debugger default delete do enum export extends finally import in instanceof new null super switch this throw try typeof void with yield

if
else
loop
stop
make
false
true
func
return
and
or

PeriDot
Total:11

2.不必要な記号の削除

プログラムは日常では使わないような記号 (; : . [(@ など)を多く使って記述されます。これらはソースの可読性を補助しますが同時にタイプミスやエラーを誘発します。そのため字句解析において必要のない記号を削除しました。主に以下の 2 つの記号です。

;(セミコロン)

C/C++などで行の終わりにつけられる記号ですが、行末判断は改行コード(＼n)で代用することができるため不必要と考え削除。

( ) (ifやloopの条件を囲む丸括弧)

ifおよびloopの条件式を囲むカッコ、それらの予約語の次に来るのは必ず式であり明示する必要がなく、可読性に影響はないと考えたため削除。

氏名　本間大一優

1-2.文法設計

3.直感的なループ処理

従来の言語(C/JS)では繰り返し処理を行う際に図1-3のように初期化,条件式,変化式を一行で書くことができました。しかし、この形式では繰り返し処理の流れが不明確となってしまいます。そのため、この省略記法をPeriDotでは採用せず、loop のあとには条件式のみ受け付けるようにしました。これにより初期化処理,変化式は図1-4のように配置する必要がありますが、処理の流れがより明確になりました。また、指定回数の繰り返し処理は図1-5のようなシンプルな記法で行えるようにしました。

Java Script

図1-3

```
for (初期化;条件式;変化式) {
        実行処理;
}
```

PeriDot

図1-4

```
初期化
loop 条件式 {
        変化式
        実行処理
}
```

図1-5

```
loop 回数 {
        実行処理
}
```

4.柔軟な型

PeriDotは動的型付けを採用し、少数と整数の演算、文字列と数字の結合を許容しています。そのためほとんど変数の型を考慮せずにプログラミングが可能であり、基礎的な学習に集中できます。

図1-6

```
make x = 2
make y = 3.5
x + y
```

```
5.5
```

図1-7

```
make apple = 3
“りんご:” + apple + ”個”
```

```
りんご:3個
```

5.従来の言語から逸脱しすぎない文法

PeriDotはあくまでも従来の言語(JavaScript, C, Go)へステップアップすることを目的とした言語であるため、その後の言語習得が違和感なく行えるように従来の基礎的な文法設計を踏襲しています。そのため、変数や関数の定義、分岐の記法はJavaScript ,C ,Go言語ライクとなっています。

6.その他

挿入可能なコメント

PeriDotは*<<コメント>>*という形でコメントを書くことができます。そのため、ソースのどこへでもコメントを挿入することができ変数や処理内容の詳しい説明が可能です。

&&,||の変更

C言語やJavaScriptではAND,OR論理演算子には***&&***,|| が用いられていました。しかしこれらは日常的に使用しない記号で、意味を汲み取ることが難しいため、**and**,**or** へ変更しました。

マルチバイト対応

従来の言語では変数名に日本語などの英語以外の言語を使用することは不可能でしたが、PeriDotでは使用可能にしました。これにより、中高生で英語に不慣れな人たちでも、直感的な記述が可能となり、プログラミング学習へのハードルが下がります。

図1-8

```
make りんご = 0
make オレンジ = 0
if りんご==0 and<<論理積>> オレンジ==0 {
        SAY(“完売” )
}
```

```
完売
```

氏名　本間大一優

1-3.独自機能

日本語でのエラー表示

ビジュアルプログラミング言語はブロックを組み合わせてプログラムを構築していくためエラーというものが存在しないのに対して、テキストプログラミングではソースコードを自由に記述できる代わりに正しい形へ矯正するエラーというものが存在します。しかし、従来の言語ではエラーメッセージは英語で記述されており、初心者には解読が難しくとてもハードルの高いものとなっていました。そこで、PeriDotではエラーメッセージの日本語化を行い、どこが間違っているのかを明記することで、エラーに抵抗なく対処することができるようにしました。以下がその例です。

間違ったプログラム	エラーメッセージ	正しいプログラム
make Array = [0,1,2] SAY(Array[3])	配列から値を取り出せませんでした。 [3]に対応する値がみつかりません。 (添字は2以下である必要があります)	make Array = [0,1,2] SAY(Array[2])
if true = true {}	'='は適切な演算子ではありません！ >>もしかして'=='?	if true == true {}
func main {}	'('となるべきところが'{'になっています！	func main() {}

実行過程の表示

プログラミング中は、常に予期せぬ値が返ってくる“バグ”という現象に悩まされます。プログラミングの経験が不十分な初心者は、バグの原因を特定するのに時間がかかることが多く、学習スピードが下がってしまいます。さらに、自分の思い通りにいかないことに苛立ちを覚え、プログラミングを嫌いになってしまう可能性もあります。それを防ぐために、PeriDotには実行過程の表示機能を組み込みました。これにより、自分が書いたプログラムがどのようにコンピュータに解釈され、動作したのかについて確認することができ、デバッグが容易になります。また、他の人が書いた優れたプログラムの実行過程を確認し理解することで、自主的に学習をすすめることができます。

入力された数字の偶奇を調べるプログラム

```
func IsEven(target) {
  return target % 2 == 0
}
make num = GETNUM()
if IsEven(num) {
  SAY(num + "は偶数です" )
} else {
  SAY(num+ "は奇数です" )
}
```

実行結果

```
4
4は偶数です
```

表示される実行過程(4を入力した場合)

```
[1] 1行目: IsEven -> func IsEven (target) { return ((target % 2) == 0}
          「関数を定義(名前:IsEven)」
[2] 4行目: GETNUM -> 実行 「関数を実行(関数:GETNUM)」
[3] 4行目:  -> 4 「組み込み関数GETNUMを実行」
[4] 4行目: num -> 4 「変数(num)を定義し(4)を代入」
[6] 5行目: IsEven -> 実行 「関数を実行(関数:IsEven)」
[7] 5行目: num -> 4 「変数(num)を参照」
[8] 1行目: target -> 4 「関数のパラメータに引数を代入」
[9] 2行目: target -> 4 「変数(target)を参照」
[10] 2行目: 4 % 2 -> 0 「計算」
[11] 2行目: 0 == 0 -> true 「評価(true or false)」
[12] 2行目: true -> 呼び出し元へ 「関数の実行結果を返す」
[13] 5行目: IsEven(num) -> true 「条件がtrueであったためif内を実行」
[14] 6行目: SAY -> 実行 「関数を実行(関数:SAY)」
[15] 6行目: num -> 4 「変数(num)を参照」
[16] 6行目: 4 + は偶数です -> 4は偶数です 「文字列結合」
```

氏名　本間大一優

2. Peree

URL:https://hmwri.com/peridot/peree/editor.php

環境構築というカベ

一般のプログラミング言語は使う前にまず実行環境を構築する必要があります。しかしその工程の中には、環境変数の設定など初心者には難しい作業が多く含まれます。そのため、初心者がプログラミングを始める上で環境構築は大きな障壁になります。そこで、PeriDotをweb上で実行可能にしたPereeというサービスを開発しました。

すべてブラウザで完結

Pereeの最大の特徴は、プログラムの編集、実行、保存、管理、共有がすべてブラウザで完結するということです。web上で動く統合開発環境といえます。そのためユーザーはwebページにアクセスするだけでプログラミングの全ての準備を整えることができ、エディタのインストールや環境変数の設定などももちろん必要ありません。初心者でもブラウザさえあれば気軽にプログラミング学習を始めることができます。

シングルページアプリケーション

Pereeではユーザビリティ向上のため、Ajaxを用いることで、プログラムの編集から実行、その結果表示まで一つの画面で行えることを可能にしています(図2-1)。これによりユーザーは実行毎の画面遷移のストレスなくスムーズに開発をすすめることができます。

図2-1:画面レイアウト

セキュリティ対策

Pereeではユーザーのアカウント情報を扱うため、暗号化通信(SSL)、プリペアドステートメントによるSQLインジェクション対策、OSコマンドインジェクション対策など基本的なセキュリティ対策を施しています。また、サーバーのリソース圧迫を防ぐために、実行時間の長いプログラムは途中終了する仕組みも埋め込まれています。

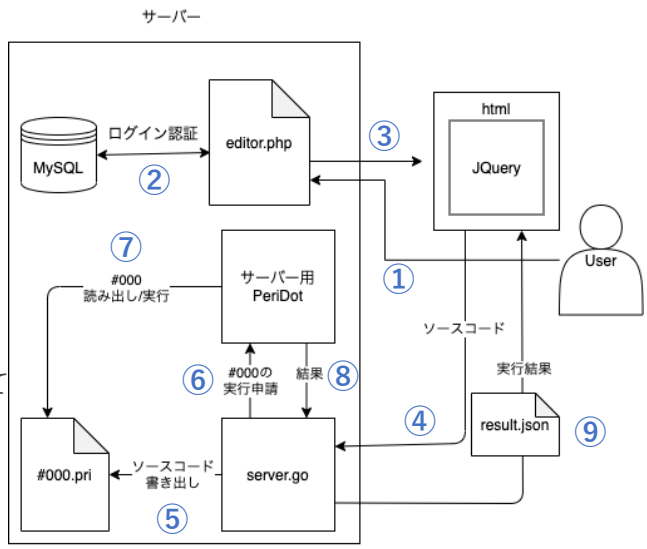
技術仕様

UI構成:**JavaScript[JQuery],HTML,CSS**
実行/結果返却、ソースの保存/読み出し:**GO言語**
ユーザーのサインイン/サインアップ:**PHP**
データベース:**MySQL**

基本的な動作の流れ(図2-2)

① ユーザーがeditor.php(図2-1)にアクセス
② MySQLにアクセスしログイン認証
③ JQueryを含むHTMLファイルを返却
④ ソースコードをserver.go にPOST(Jquery)
⑤ ソースコードにIDを付与しテキストファイルとして保存(OSコマンドインジェクション対策)
⑥ IDとともにサーバー用PeriDotに実行申請
⑦ 送られたIDを元にプログラムを実行
⑧ 結果をserver.goへ返却
⑨ JQueryへresult.jsonとして結果を返却

図2-2:基本的な動作の流れ



氏名　本間大一優

2-1.Peree 操作の流れ

アカウント登録/ログイン

Pereeのソースコード保存,共有機能を利用するにはアカウント登録(図2-3)が必要です。パスワードはハッシュ化されてデータベースに保存されるため、安心して利用できます。また、ログイン毎にトークンを発行し、実行や保存の際にトークン認証を行うためセキュアな通信が可能です。

マイページ

アカウント登録をしたユーザーにはマイページ(図2-4)が付与されます。ここではユーザーが作成したプロジェクト一覧を確認することができ、さらにプロジェクトの新規作成、変更、削除が可能です。

図2-3:アカウント作成画面

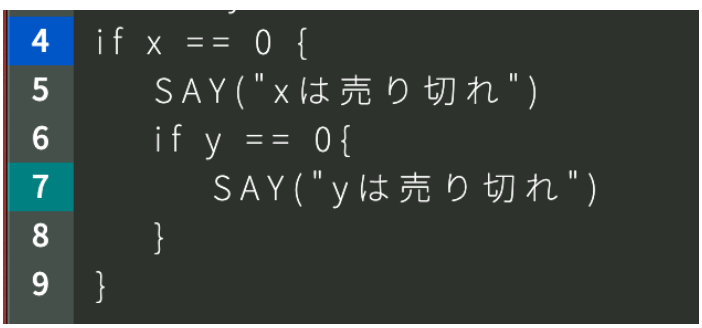
図2-4:マイページ

プログラムの編集/実行

マイページでプロジェクトを開くと図2-5の画面に遷移します。編集エリアにてプログラムを記述します。ここでは、自動インデント補完機能によりプログラムは見やすい形に整形されます(図2-6)。 画面下部にあるツールバー上の実行ボタンを押すことで、サーバー側でプログラムが実行されます。

図2-5:編集/実行画面

図2-6:自動インデント補完機能



氏名　本間大一優

2-1.Peree 操作の流れ

実行結果/過程の確認

サーバーで実行が終わると返却された実行結果と実行過程が画面右側に表示されます(図2-7)。最新の実行結果は上から追加される形で表示されるため、以前の結果との比較が可能です。また、ツールバーの矢印ボタンを押すと実行過程を工程ごとに追うことができます。また、編集エリアにて選択している工程の行番号とエラーのある行番号はハイライト表示されます。

図 2 -7:結果と過程の表示

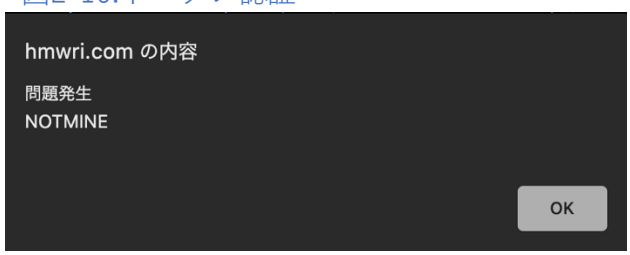
プログラムの保存/共有/リミックス

“上書き保存”ボタンを押すと、開いているプロジェクトに編集したコードが直接上書き保存されます。また、”別名で保存”ボタンを押すことで、任意の名前をつけて新しいプロジェクトとして保存ができます。共有ボタンを押すとURLが表示され、そのURLを共有することで、プロジェクトをシェアできます(図2-8)。また、他人のプロジェクトを別名で保存することで、自分のプロジェクトとしてリミックスすることもできます。ここで、上書き保存ボタンはプロジェクトを保護するため、プロジェクト作成者以外には非表示になっています(図2-9)。悪意のある第三者が不正な上書き保存リクエストをしたとしても、前述したトークンの認証によって上書きできないようになっています(図2-10)。

図2-8:共有機能

図2-9:上書き保存ボタンの非表示

図2-10:トークン認証



氏名　本間大一優

3.PeriWorld

URL:https://hmwri.com/peridot/periWorld/

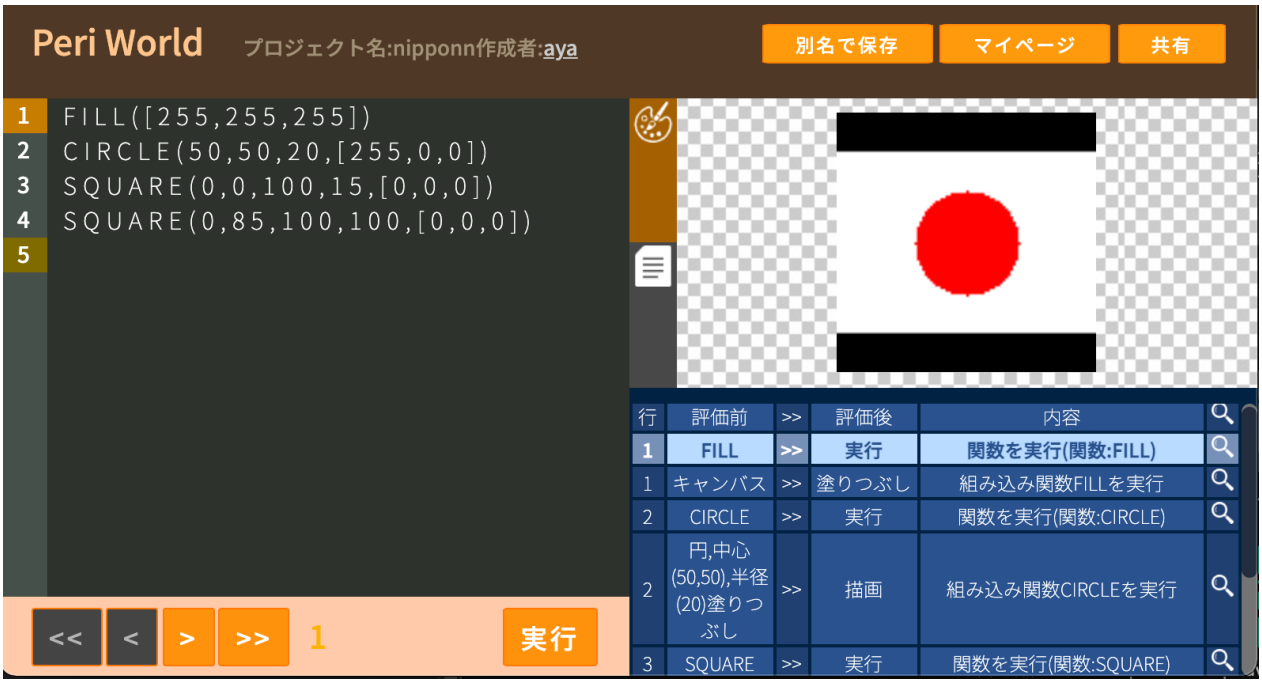
成果が目に見える

プログラミングを学ぶことの楽しさは、”できることの広がり”にあると私は思います。しかし、テキストプログラミングは結果の形式がテキストであるため、それを実感するのが難しく、それ故途中で挫折してしまう人が多いのではないかと考えました。そこで、プログラミングの楽しさをより多くの人に実感してもらえるように、成果が図形として目に見え、学べば学ぶほど多様な図形が描けるようになるPeriWorldを開発しました。

誰でも簡単にデザイン

PeriWorldはPereeと同じようにwebサービスであるため、ブラウザで作業が可能です(図3-1)。また、画像をデザインするための関数も直感的に設計されているため、誰でも簡単にPeriDotで画像をデザインできます。

図3-1:PeriWorld実行画面



描画関数一覧

- **FILL([R,G,B,(A)])**
  単色でキャンバス全体を埋める
- **LINE(x1 , y1 ,x2 , y2 ,[R,G,B,(A)])**
  始点(x1,y1),終点(x2,y2)の線を描く
- **DOT(x , y ,[R,G,B,(A)])**
  (x,y)に点を打つ
- **CIRCLE(x ,y ,r ,[R,G,B,(A)])**
  中心(x,y),半径rの円を塗り潰す
- **BCIRCLE(x ,y ,r ,[R,G,B,(A)])**
  中心(x,y),半径rの円の輪郭を描く
- **SQUARE(x1 , y1 ,x2 ,y2 ,[R,G,B,(A)])**
  始点(x1,y1),終点(x2,y2)の長方形を塗り潰す
- **SQUARE(x1 , y1 ,x2 ,y2 ,[R,G,B,(A)])**
  始点(x1,y1),終点(x2,y2)の長方形の輪郭を描く

氏名　本間大一優

3.PeriWorld

みんなのデザインから学べる

PeriWorldでは、各自の作品は図3-2のような地図上に個々の割り当てられた土地のデザイン(図形)として自動的に共有されます。地図上では自由に移動することができ、他の人の作品をみることができます。また、画像をクリックすることでソースコードを見ることができる(図3-3)ため、主体的に他の人のコードから学習することが可能です。また全体俯瞰できることにより、視覚的に個々の土地(図形)の複雑さがわかるため、教育者による評価も容易です。

図3-2:地図表示機能

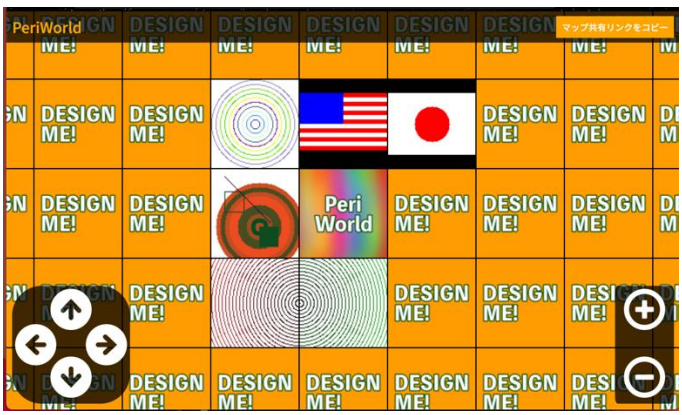
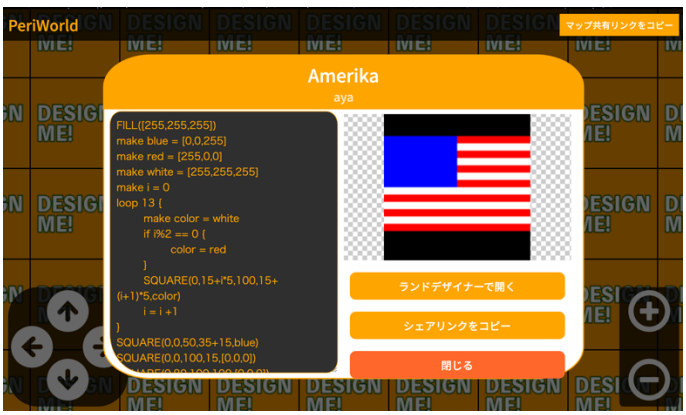


図3-3:ソースコード閲覧機能



みんなでデザインする

複数人の土地(図形)を合体させ、大きな土地をみんなでデザインする”グループデザイン”も可能になっています(図3-4)。グループデザインを通して他人と協力しあい楽しみながらプログラミングを学ぶことができます。

技術仕様

実行の仕組みは基本的にPereeを継承していますが、地図表示機能のために以下のようなプログラムを新たに構築しました。

① 新規プロジェクトを作成する際にデーターベースにプロジェクトIDとそれに対する適切な座標 (座標は図3-5右下のように定義され、渦巻状に使用していく)をインサートし、プロジェクトIDを付与した画像を作成するプログラム

② 地図表示画面からの座標リクエストに対して①のデータベースと照合し、適切なプロジェクトIDを返却するプログラム

また、一度に複数の座標のリクエストを可能とすることでサーバーとの通信回数を減らし表示高速化を図る工夫を実施しました。

図3-4:グループデザイン機能

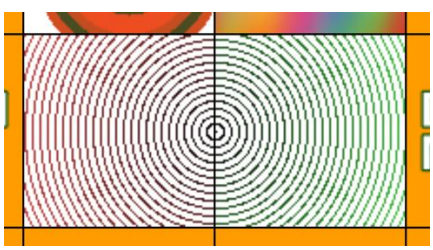


図3-5:処理の流れ