コンピュータサイエンス第一
課題 1「四則演算でアニメーション」

学籍番号******** ****

概要

課題として、CUI 上で表現される四則演算を利用したアニメーションの作成に取り組んだ。Ruby という言語を用いて作成し、プログラムは anime.rb というファイルに保存した。

1　実行方法とアニメーションの説明

1.1　実行方法

ターミナルを開き、anime.rb が存在するディレクトリ内で、以下のコマンドを実行することにより、本プログラムを実行できる。

```
ruby anime.rb
```

1.2　アニメーションの説明

道路上を車が奥に進んでいくというアニメーションを作成した。実際には、車を固定することで、相対的に道路が手前に動くように描いている。コマは全部で 100 コマで、0.3 秒ごとに新しいコマが表示される。

2　計算の仕組み

表示する内容は行ごとに 25 桁の整数で表される。各桁は 0 または 1 である。

道路の描画には剰余を利用し、5 行ごとに 3 行分の縦線と 2 行分の空白が描画されるようにした。行に番号を振り、その数字を 5 で割った時の余りが 1 以下か 2 以上かを評価することでこの分岐を実装している。

道路のレイヤーと車のレイヤーを合成するために、和を用いている。どちらのレイヤーでも 1 になるところは無いため、素直に 2 つのレイヤーの数を足すことで、合成した行のデータができる。

3　工夫した点

制御文字を出力することによって、毎コマごとターミナルの表示をリセットし、画面の遷移がわかりやすくなるようにした。プログラムの見通しがつきやすいよう、役割ごと関数に分け、適切なコメントを付与した。配列と for 文を利用することで、繰り返し行う処理を短く記述した。

4　ソースコード

以下に anime.rb のソースコードを示す。

```
# anime.rb
# Author: Kazuki Asakura

# 予約用の行
$reserveLines = []
# 表示用の行
$displayLines = Array.new(10)
# 車表示用のレイヤーデータ
$carLines = [
  0,
  0,
  0,
  0,
  0,
  1000000000000000000,
  1111100000000000000000,
  1000100000000000000000,
  1000100000000000000000,
  0
]

# 表示する行を追加
def addDisplayLine(newLine)
  $displayLines.pop                # 最後の要素を削除
  $displayLines.unshift(newLine) # 新しい要素を先頭に追加
end

# displayLinesの内容を表示
def printLines
  puts "\e[H\e[2J" # clear
  for i in 0..9
    puts $displayLines[i] + $carLines[i] # 道路レイヤーに車を合成
```

```
  end
  sleep(0.3)
end

# 道路を作成、予約
def createRoad(length)
  for i in 0..length
    if i % 5 <= 1
      $reserveLines.unshift(1000000000000000000000001)
    else
      $reserveLines.unshift(1000000000001000000000001)
    end
  end
end

# init
createRoad(100)
for i in 0..9
  $displayLines.unshift($reserveLines.pop)
end
# reserveLinesが空になるまでコマ送り
while ! $reserveLines.empty? do
  addDisplayLine($reserveLines.pop)
  printLines()
end
```