

可逆アルゴリズムのアルゴリズム図鑑

2018SE016 平工真基 2018SE076 佐々木龍之介

2018SE088 高見雄大 2018SE091 谷崎海良

指導教員 横山哲郎

1 はじめに

可逆アルゴリズムのオンラインでの実行環境は整っているが、現在は実行結果しか表示されずどのようにしてそのような結果が実行されたのか理解するのが難しいと感じた。そこで、アルゴリズムの実行過程を可視化することにした。

1.1 アルゴリズムとは

アルゴリズムとは、与えられた入力から必要な出力を得るために定められた明確な手順の事である。この手順は、たいてい、コンピューターがプログラムとして実行することである。[1]

1.2 可逆とは

可逆であるとは、ある状態から別の状態に変化したときに元の状態に戻ることができることをいう。[2] 可逆でないということは非可逆であるということである。

1.3 可逆アルゴリズムとは

可逆アルゴリズムとは、出力されたものから入力されたものを一意に定めることができるアルゴリズムのことである。[2]

2 解決策の立案

解決策として Unity を使用してパソコン上で動かせるアルゴリズムの動きを一つ一つ解説するアプリを作成する。今回は、可逆アルゴリズムの中でも、可逆線形探索の解説を作成することにした。解説を作成する時、可逆線形探索のソースコードは [3] を参考にした。

3 代替案の立案

代替案として、

- 1.web ページで作成 (html/css) する。
- 2.powerpoint を用いてスライド形式で解説を作成する。の 2 案を立案した。

4 実装

unity を用いてアルゴリズムの解説を作成する際、主に以下の 3 種類のスクリプトを c# を用いて作成した。

1. ボタンを押すと画面が切り替わるようにする
2. 解説文に合わせて図 (矢印やゴミ箱の中身) が変わるようにする
3. 画面を押すと解説文が進んでいくようにする

4.1 ボタンを用いた画面の切り替え

ボタンを押すことで画面を切り替えるために、画面ごとにシーンを作成し、SceneManager.LoadScene() を使用することで、ボタンを押したときに画面が切り替わるようにした。例えばタイトル画面から目次の画面に移動するプログラムのソースコードは図 1 のようになる。

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class StartButton : MonoBehaviour
7 {
8     // Start is called before the first frame update
9     void Start()
10     {
11
12     }
13
14     // Update is called once per frame
15     void Update()
16     {
17
18     }
19     public void Click(){
20         SceneManager.LoadScene("contents");
21     }
22 }
```

図 1 タイトルから目次に移動するプログラム

4.2 解説文に合わせた図の変更

解説文に合わせて図を切り替えるために Input.GetMouseButtonDown(0) を用いて左クリックされたかを感知するようにし、左クリックされたら変数 cntclick が増加するようにした。そして、cntclick の値に応じて図が移動したり現れたりするようにした。例えば解説文に合わせて矢印が移動するプログラムのソースコードは図 2 のようになる。

4.3 クリックによる解説文の遷移

画面をクリックすることで、解説文が切り替わるようにするために、事前に解説文を string 型の配列の中に入れておき、その後、this.GetComponent<Text>().text を使用して配列の中身を 1 文ずつ表示していく。GetMouseButtonDown(0) を用いて左クリックを感知した際、変数 tnum の値を増やし、配列を texts[tnum] と参照することによって表示する文章を切り替える。配列の中身をすべて表示した後に目次に戻るようにするために、予め変数 n に配列の長さを記録しておき、tnum と n が一致した状態で左クリックされた時に、SceneManager.LoadScene() を用いて目次へ移動させる。解説文を表示するプログラムの

ソースコードの一部は図 3 のようになる。

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class MoveArrow : MonoBehaviour
{
    public int cntclick=0;
    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        if(Input.GetMouseButtonDown(0)){
            cntclick++;
            if(1 < cntclick && cntclick < 5){
                this.transform.position += new Vector3(2, 0, 0);
            }
        }
    }
}
```

図 2 解説文に合わせて矢印が移動するプログラム

```
void Update()
{
    this.GetComponent<Text>().text = texts[tnum];
    if(Input.GetMouseButtonDown(0) && tnum<n) tnum++;
    if(Input.GetMouseButtonDown(0) && tnum==n) SceneManager.LoadScene("contents");
}
```

図 3 クリックによって解説文が遷移するプログラムの一部

5 おわりに

今回は Unity を使用して可逆線形探索の一般解法のアロゴリズムの動きを解説するアプリを作成した。今後の課題として、別のアロゴリズムの動きについても解説していけるようにしたり配列やキーなどの入力を自由に変更していけるようにアプリを拡張していきたい。

6 役割分担

レポート作成 谷崎, 高見
アルゴリズム図鑑作成 平工, 佐々木

参考文献

- [1] 結城浩.「数学ガール/乱択アルゴリズム」, SBCreative, 2015 年, pp.21-54.
- [2] 家崎雄太, 水野竣太郎,「可逆線形探索」, 閲覧日 2021-01-25,
<http://www.st.nanzan-u.ac.jp/info/gr-thesis/2017/yokoyama/pdf/14se024.pdf>

- [3] 「reversible-algorithms/Lsearch1.j」, github, 閲覧日 2021-01-25,
https://github.com/yokoyama-lab/reversible-algorithms/blob/master/linear_search/Lsearch1.j