

# 条件最適化席替えアプリ“めえめえ2号”の設計・開発

兵庫県立星陵高校 2年 浦尻 雄斗 1年 織田 竜輝 1年 堀田 綾大

## 1. 研究動機・目的

昨年度私たちの先輩方が、本校で現在も活躍中の席替えAI“めえめえ”を作成した。しかしながら、次の3つの課題を残していた。

### 課題

1. 前列希望者は前から2列目以外の列を希望列にできない。
2. 普通教室より大きい特別教室の席替えには非対応。
3. 2回以上前に隣り合った生徒と隣り合うことがある。

そこで、上記の課題全てを克服した席替えアプリの開発を目的として研究を行った。

## 2. 研究方法

席替えAI“めえめえ”を解析し、アルゴリズムを把握した。これを基本としつつ3つの課題を解決するために、アルゴリズムの拡張・発展や新たな数式モデルの考案をして、研究を行った。

開発環境は次のとおりである。

開発言語	Python	3.9.6 (64-bit)
使用ライブラリ	NumPy	1.21.1
	openpyxl	3.0.7
	PySimpleGUI	4.46.0

## 3. 研究結果

先に挙げた3つの課題全てを克服した席替えアプリ“めえめえ2号”を開発した。本アプリは様々な条件を付けて席替えを実行することができ、加えてGUIを採用しているため、ボタンなどによる視覚的操作が可能である(図2、図5、図6、図7)。また、課題の克服により座席表サイズを自由に設定できる。

本アプリは通常、後述の“隣条件”、“座席条件”を満たすように席替えを実行する。このとき、さらに“希望列条件”や“名列順条件”を付け加えることもできる。

	A	B	C	D
1	出席番号	名前		
2		1 青山		
3		2 石見		
4		3 伊藤		
5		4 井上		
6		5 上田		
7		6 梅原		
8		7 太田		
9		8 大塚		

図1 生徒の名列



図2 メイン画面

初期設定

行数を入力してください

列数を入力してください

次へ

図5 行数列数設定用画面

前列配慮

前の席を希望する人を選択します

その希望列を選択 **選択**

同じ列を希望する生徒の出席番号をすべて入力

セーブ 入力終了(セーブしてから押してください)

図7 前列希望者登録用画面

※座席表の名前はすべて架空のものです。

	A	B	C	D	E	F	G	H	I	J	K	L
1					教	卓						
2												
3	1 青山	7 太田	14 北野	21 関	28 長谷川	35 前田						
4	2 石見	8 大塚	15 木下	22 辻	29 林	36 松本						
5	3 伊藤	9 岡崎	16 木村	23 寺田	30 平井	37 向井						
6	4 井上	10 小野	17 河本	24 中井	31 福島	38 森本						
7	5 上田	11 梶川	18 坂本	25 中村	32 藤井	39 山口						
8	6 梅原	12 加藤	19 佐々木	26 丹羽	33 藤本	40 米田						
9		13 川口	20 杉本	27 橋口	34 細川							
10												

図3 名列順座席表

	A	B	C	D	E	F	G	H	I	J	K	L	M
1					教	卓							
2													
3	8 大塚	18 坂本	3 伊藤	23 寺田	15 木下	33 藤本							
4	13 川口	22 辻	17 河本	32 藤井	26 丹羽	7 太田							
5	37 向井	19 佐々木	25 中村	38 森本	24 中井	4 井上							
6	29 林	28 長谷川	39 山口	31 福島	6 梅原	34 細川							
7	16 木村	27 橋口	10 小野	40 米田	14 北野	30 平井							
8	2 石見	35 前田	36 松本	20 杉本	1 青山	5 上田							
9		21 関	12 加藤	11 梶川	9 岡崎								
10													

図4 ランダム座席表

### 3.1 “隣条件”と“座席条件”

“隣条件”を付与すると二度と同じ生徒と隣り合わないように、“座席条件”を付与すると二度と同じ席に着かないように席替えを実行できる。

### 3.2 “希望列条件”

“希望列条件”を付与すると前列希望者は希望列を指定できるようになり、その希望列以内の座席に着くように席替えを実行できる。課題の克服によりすべての列を希望列に指定できる。

### 3.3 “名列順条件”

“名列順条件”を付与することで名列順の座席表を作成することができる。図3は名列順座席表、図4は通常のランダム座席表である。“名列順条件”は初回のみ適用される。

## 4. 考察

座席数が多くなるにつれて、“隣条件”を満たす座席表の作成に時間がかかることが分かった。例えば2回目の席替え時、普通教室(8行6列)のときは約0.5秒、100行100列のときは約6.0秒かかる。

解決策として、AI化による条件の緩和や、処理方法の見直しで軽量化することを検討している。

## 5. 参考文献

- Pythonの文法の学習 (2021/7/23 最終閲覧)  
Qiita 作者: [@Fendo181\(GMOペパボ株式会社\)](https://qiita.com/Fendo181/items/a934e4f94021115efb2e)  
<https://qiita.com/Fendo181/items/a934e4f94021115efb2e>
- NumPyの利用 (2021/8/30 最終閲覧)  
Qiita 作者: [@jyori112\(Studio Ousia\)](https://qiita.com/jyori112/items/a15658d1dd17c421e1e2)  
<https://qiita.com/jyori112/items/a15658d1dd17c421e1e2>