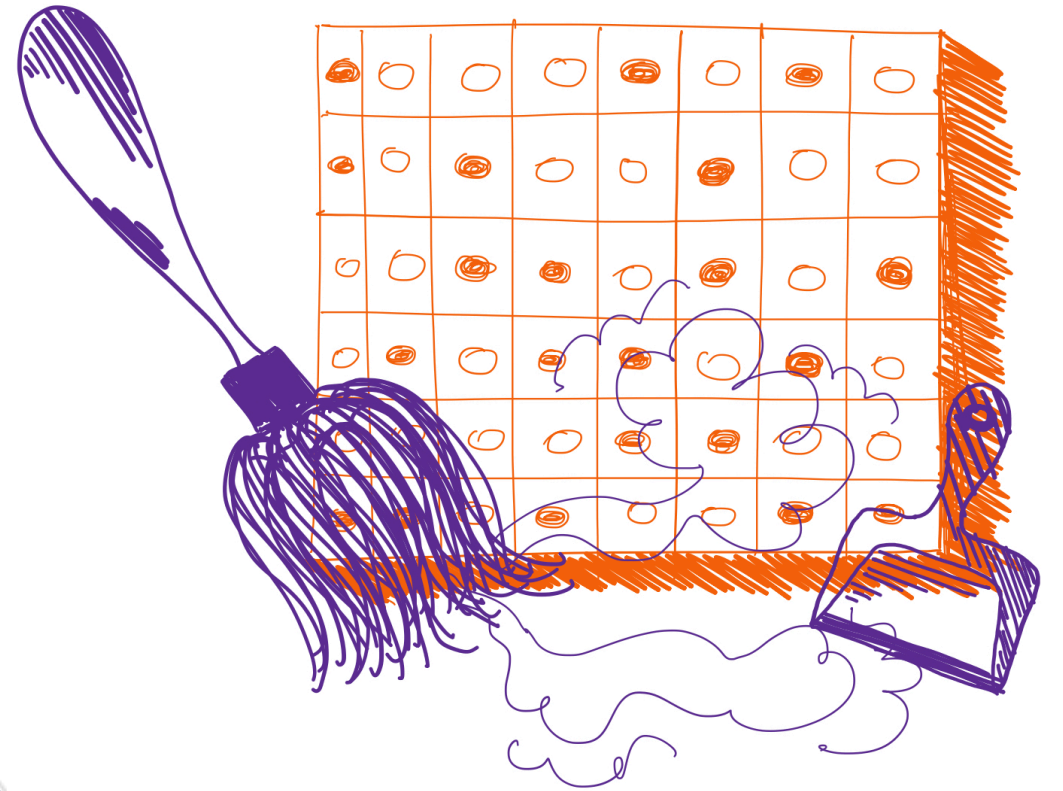


SWEEP THE BOARD

BY NOUR HAYEK, JASON CARPIO, LAUREN GRAY, PAOLA KARAPATAKI, SACHA PRITZKER, AND GAURAV VARMA

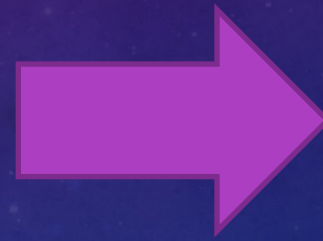


GENERAL GAME SETUP

Start of Game

○	○	●	○	●	
●	○		●	○	●
	●	●		○	○
●	○		●	○	●

Not echelon form



End Goal of Game

●				○	○
	●			○	○
		●			○
			●	○	○

Reduced Echelon Form

- Each symbol corresponds to a different value
- The code is written in Mod3 (adding two numbers takes the remainder when dividing by 3)
- The user can reduce the matrix using row addition and interchanging of rows

Number Key :

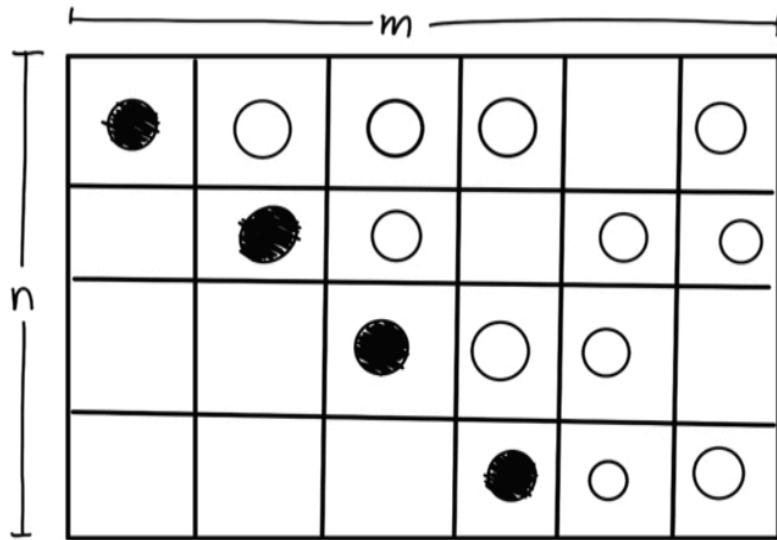
$$\square = 0$$

$$\blacksquare = 1$$

$$\circ = 2$$

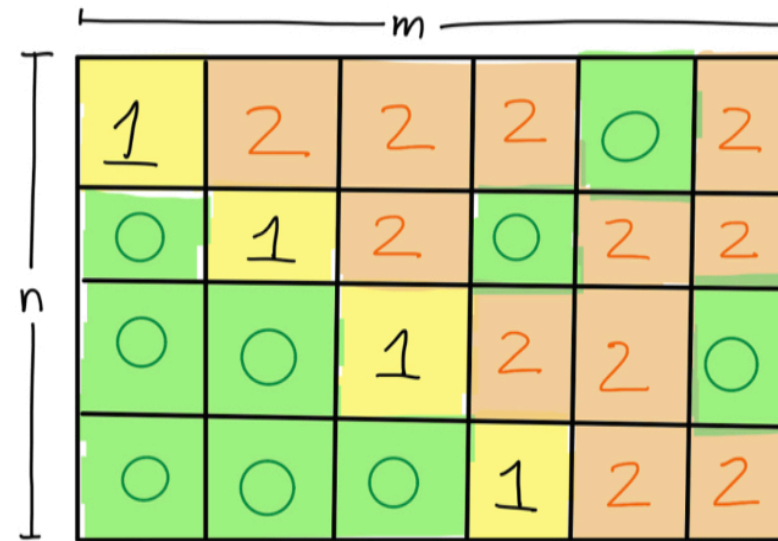
$$\blacksquare + \blacksquare = \circ$$

$$\blacksquare + \circ = \square$$



pivot columns
(basic variables)

non-pivot
columns
(free variables)



pivot columns
(basic variables)

non-pivot
columns
(free variables)

THE FOLLOWING MATRICES ARE EQUIVALENT

```
1  """
2  Author: Qirong Li
3  Date: February 2019
4  Project: Row reduction game
5  """
6  from Game import Game
7  import sys
8
9  new_game = Game(board_size = int(sys.argv[1]))
10
11  #new_game = Game(board_size = 6)
12  new_game.start()
```

Hello from the pygame community. <https://www.pygame.org/contribute.html>

```
(venv) Jasons-MacBook-Pro-5:LinearGames_sweep_the_board-3 jasoncarpio$ python3 start_game.new.py 3
```

ALTERING THE DIMENSIONS OF THE MATRIX THE ORIGINAL CODE

```
math_22 | LinearGames_sweep_the_board-3 | start_game.new.py
Project | README.txt | start_game.py | setup.py | start_game.new.py | Game.py | Game2.py | constants.py | Game_prev.py
math_22 ~/PycharmProjects/math_22
  LinearGames_sweep_the_board-3
    constants.py
    Game.py
    Game2.py
    Game_prev.py
    README.txt
    setup.py
    start_game.new.py
    start_game.py
    venv
  External Libraries
  Scratches and Consoles
1  """
2  Author: Qirong Li
3  Date: February 2019
4  Project: Row reduction game
5  Modified by: Lauren
6
7  from Game import Game
8  import sys
9
10 new_game = Game(row_size = int(sys.argv[1]), col_size = int(sys.argv[2]))
11 # new_game = Game(board_size = 6)
12 new_game.start()
13
14
out of the if statements, col is
3
(venv) Jasons-MacBook-Pro-5:LinearGames_sweep_the_board-3 jasoncarpio$ python3 start_game.new.py 3 4
pygame 2.0.0.dev4 (SDL 2.0.10, python 3.7.4)
```

ALTERING THE DIMENSIONS OF THE MATRIX OUR MODIFIED CODE

Original Code

```
228 def init_grid(self):
229     grid = []
230
231     for y in range(self.board_size):
232         grid.append([])
233         for x in range(self.board_size):
234             if x == y: # diagonal: black
235                 color = 1
236             elif x > y:
237                 rand = random.uniform(0, 1)
238
239                 if rand <= 1/4:
240                     color = 1
241                 elif rand <= 2/3:
242                     color = 0
243                 else:
244                     color = -1
245             else:
246                 color = -1
247
248             grid[y].append([cs.grid_left + x*(self.block_size + 2), \
249                             cs.grid_top + y*(self.block_size + 2), color])
250         # print(grid)
251     return grid
252
253 def rand_grid(self):
254     l = list(range(self.board_size))
255
256     for i in range(2*self.board_size):
257         rand = random.uniform(0, 1)
258         random.shuffle(l)
259         row1, row2 = l[0], l[1]
260         # print(row1, row2)
261         if rand <= 0.5:
262             self.add_init(row1, row2)
263         else:
264             self.swap_init(row1, row2)
265
266
267
```

Modified Code

```
221 # [x, y, color]
222 # -1: empty, 0: white, 1: black
223 def init_grid(self):
224
225     grid = []
226
227     for y in range(self.row_size):
228         grid.append([])
229         for x in range(self.col_size):
230
231             rand = random.uniform(0, 1)
232
233             if rand <= 1/4:
234                 color = 1
235             elif rand <= 2/3:
236                 color = 0
237             else:
238                 color = -1
239
```

RANDOMIZING
ALL POSITIONS
OF THE
MATRIX

Original Code

```
74
75 def is_goal(self):
76
77     count = 0
78
79     for row in range(self.board_size):
80         for col in range(self.board_size):
81
82             color = self.grid[row][col][2]
83             # the original goal
84             if color == 0:
85                 return False
86             elif color == 1:
87                 count += 1
88
89             # if color != -1:
90                 count += 1
91
92     return (count <= self.board_size)
```

Modified Code

```
79 def is_goal(self):
80
81     color = 0
82     hasPivot = False
83     emptyRow = False
84     for row in range(self.row_size):
85         col = 0
86         while col < self.col_size and hasPivot == False:
87             color = self.grid[row][col][2]
88             if emptyRow == True:
89                 if color != -1:
90                     return False
91             if color == 1:
92                 hasPivot = True
93                 for r in range(self.row_size):
94                     color = self.grid[r][col][2]
95                     if color != -1 and r != row:
96                         return False
97             elif color == 0:
98                 return False
99             col += 1
100         if hasPivot == False:
101             emptyRow = True
102             hasPivot = False
103
104     return True
```

CHANGING THE IS_GOAL FUNCTION

1. The Python code of the original version of the game “Sweep the Board”
https://math.dartmouth.edu/~m22f19/LinearGames_sweep_the_board.zip
2. An online version of Linear Matrix Games www.doob.dartmouth.edu:5000
3. A file explaining the mathematics behind the matrix games
https://math.dartmouth.edu/~m22f19/LinearGames_math_behind.pdf

LIST OF REFERENCES



"That's all, Folks!"