

INTERNATIONAL INFORMATION
TECHNOLOGY UNIVERSITY

2048 in PYTHON language

By Abzal,Bekzat,Aibek,Anuar

2048

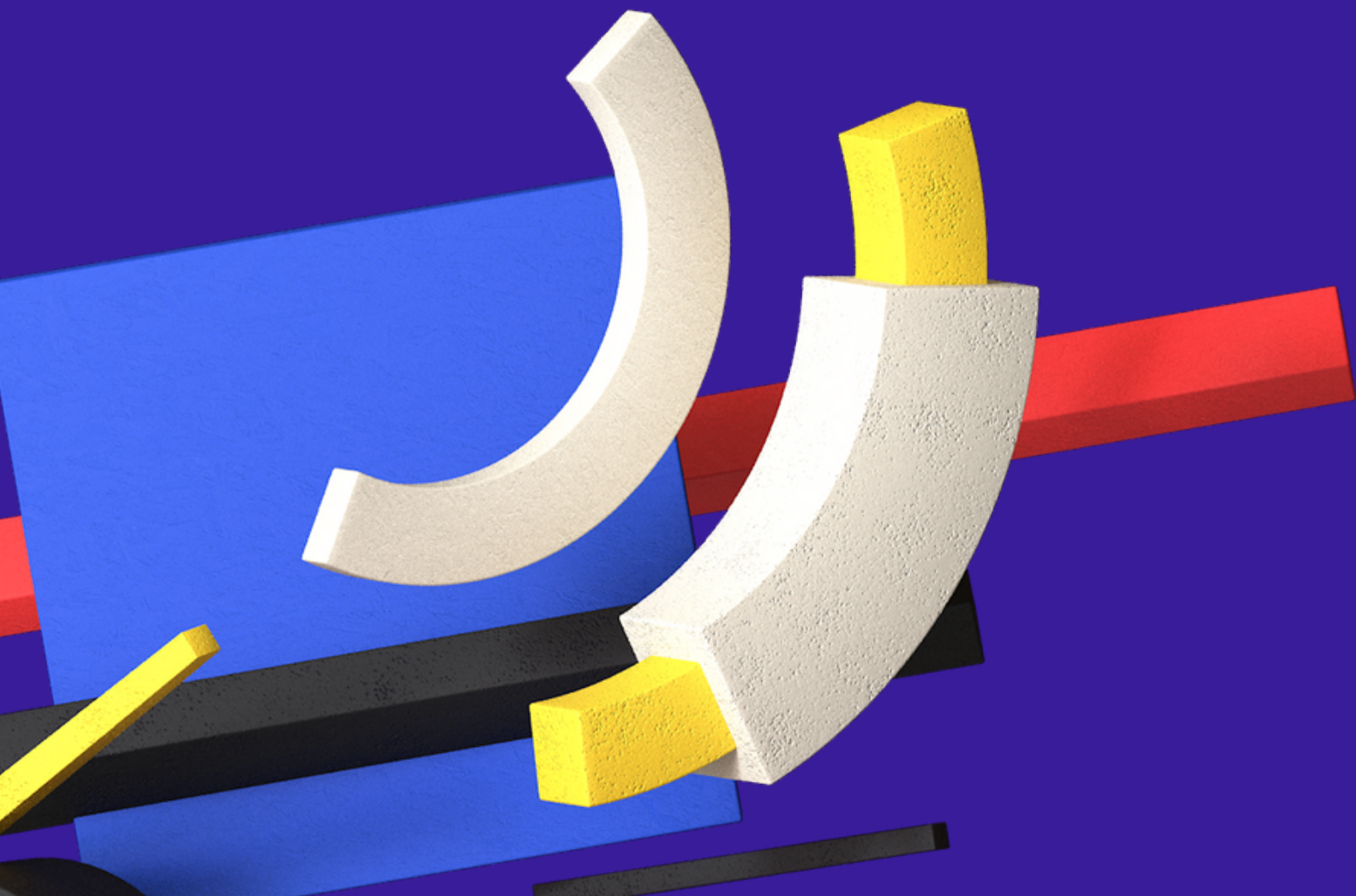


OUR PLAN

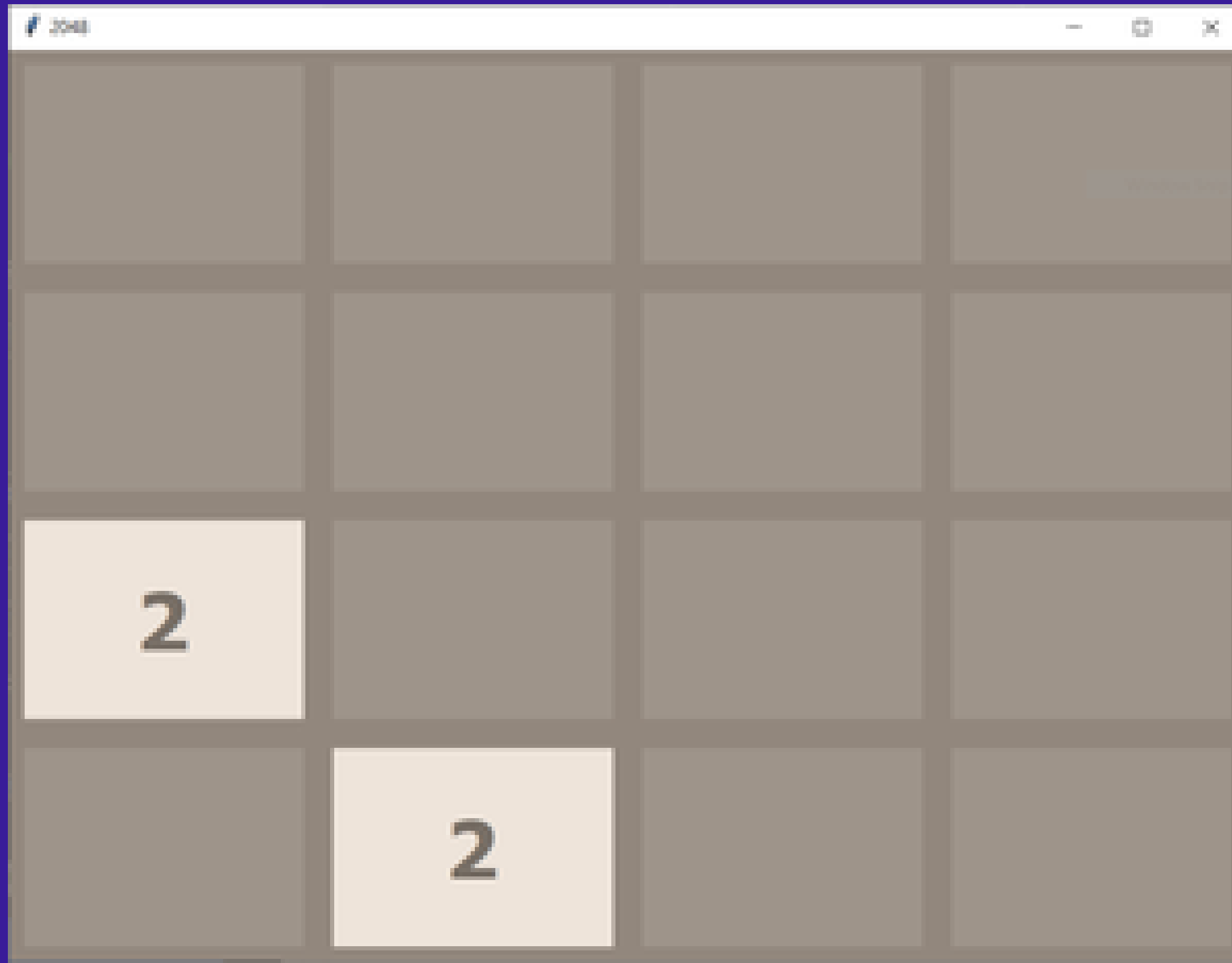


HISTORY <<2048>> GAME

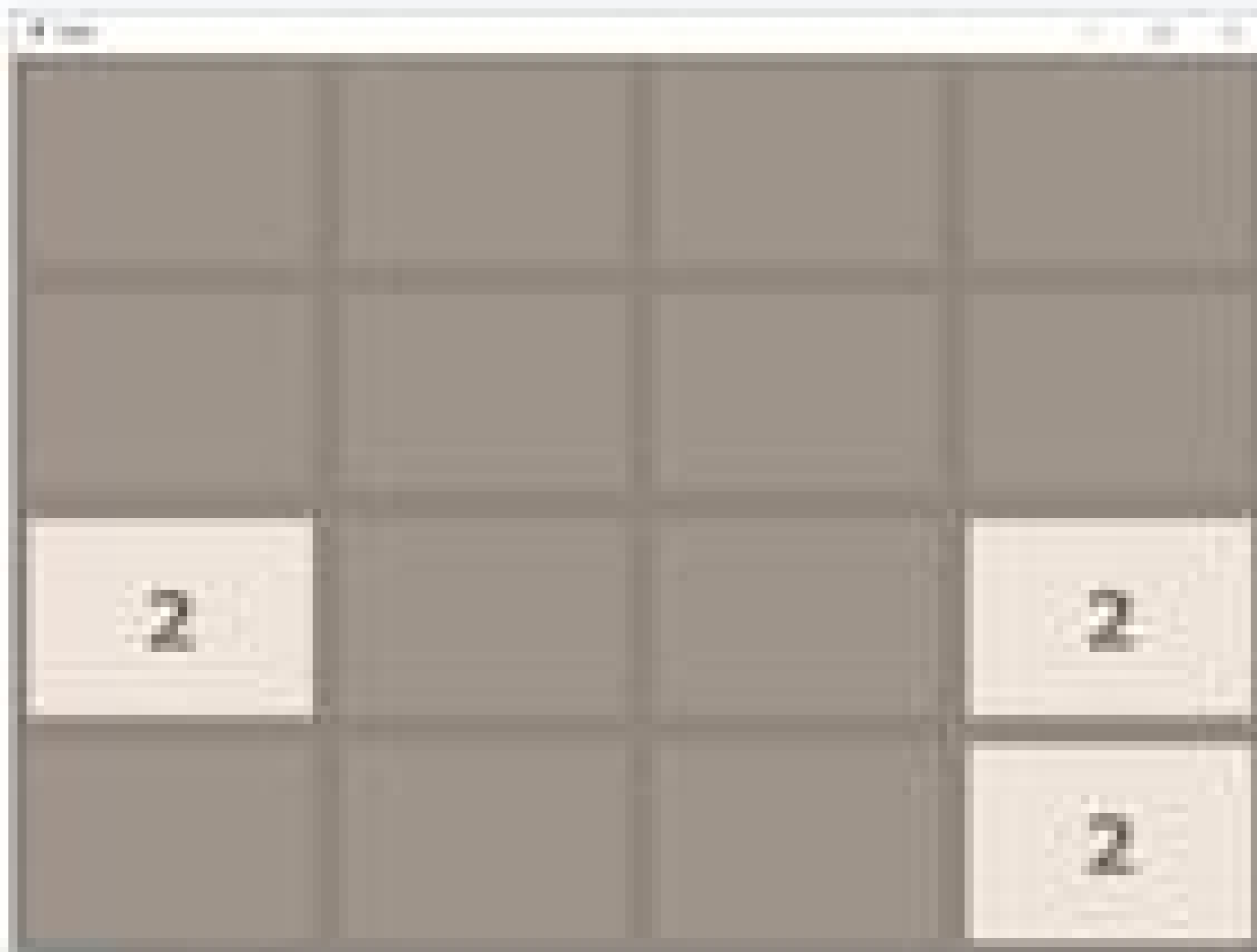
- 2048 is a browser game written by 19-year-old Italian developer Gabriele Cirulli in the JavaScript programming language. The game "2048" was written in less than two days as an exercise in programming according to the author.



There is a 4*4 grid that can be filled with at least some number. Initially, two random cells are filled with the digits 2. The other cells are empty.

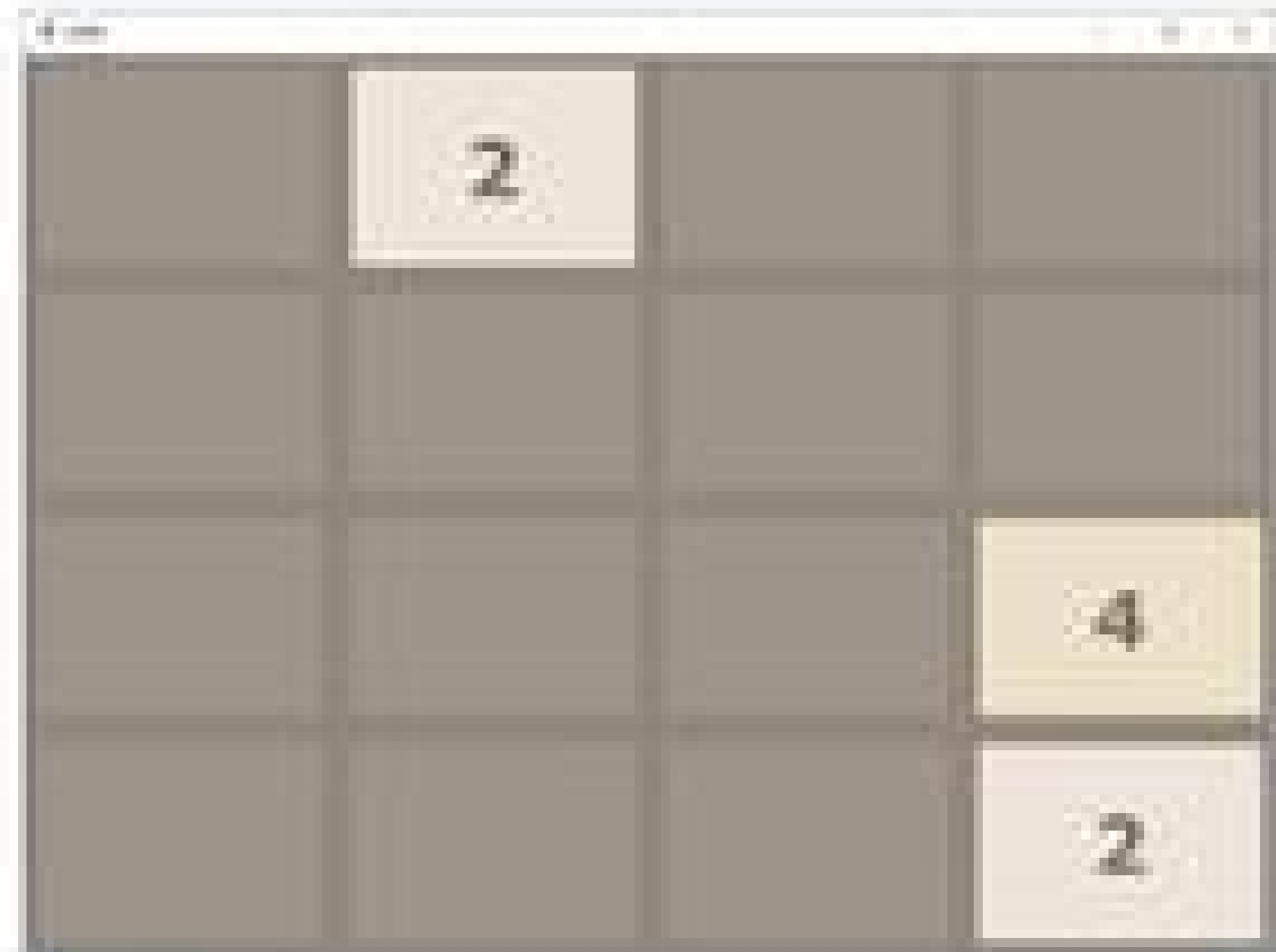


We have to press any of the four keys to move up, down, left or right. .We press any button, the elements of the cell move in this direction so that if any two similar numbers are contained in a given specific row or column, they get the addition and the last cell in this direction will be filled with this number.



A screenshot of a 4x4 grid game state. The grid is mostly empty, with three cells containing the number 2. These cells are located at row 3, column 1; row 3, column 4; and row 4, column 4. The cells are highlighted in a light beige color.

| | | | |
|---|--|--|---|
| | | | |
| | | | |
| 2 | | | 2 |
| | | | 2 |



A screenshot of a 4x4 grid game state after a move. The grid now contains four cells with numbers: a 2 at row 1, column 2; a 4 at row 3, column 4; and a 2 at row 4, column 4. The cell with the 4 is highlighted in a yellow color, while the others are light beige.

| | | | |
|--|---|--|---|
| | 2 | | |
| | | | |
| | | | 4 |
| | | | 2 |

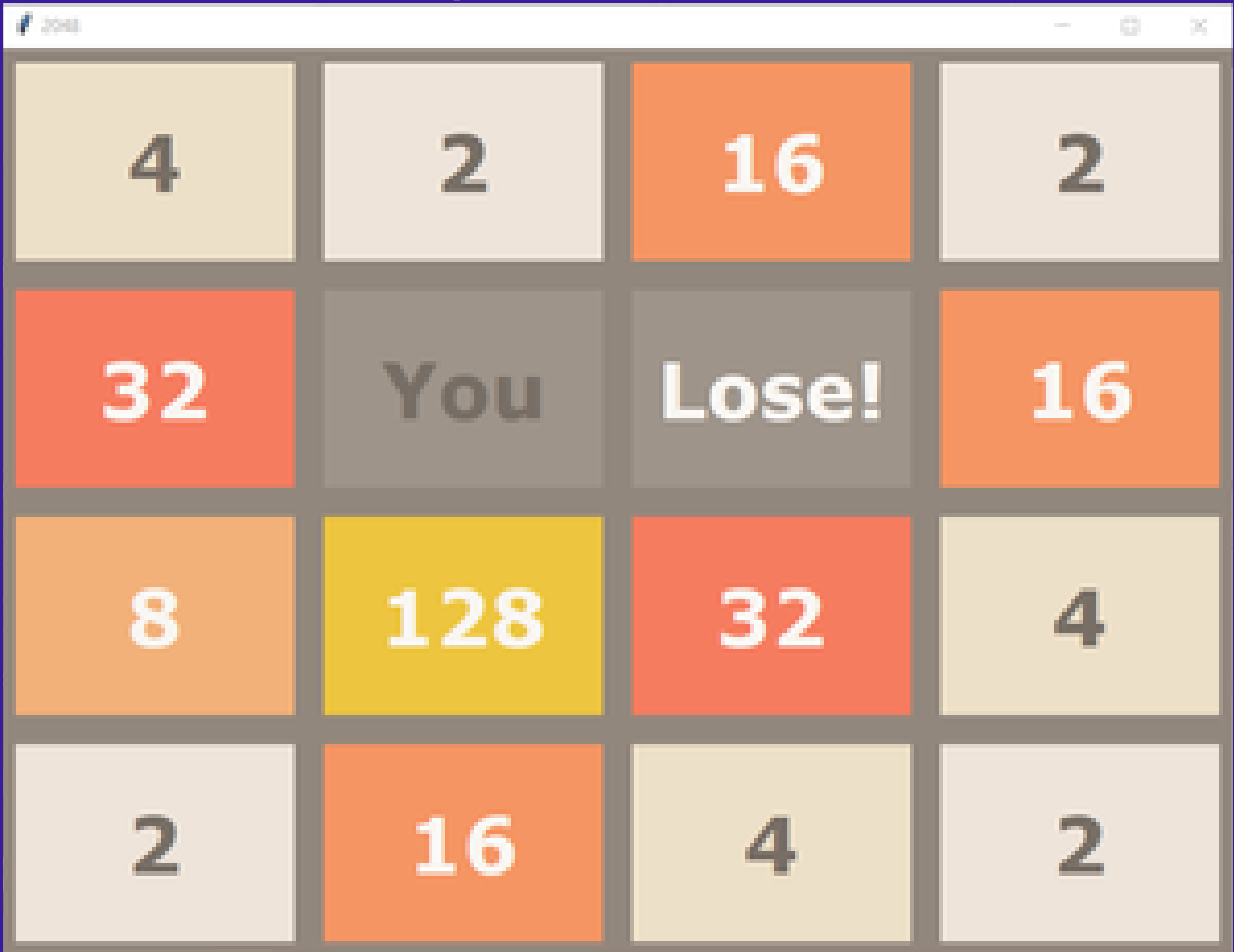


After this
grid
exhibition,
any random
empty cell
is
accumulate
d by the
number 2.

Following the process described above, we must double the elements by adding and getting 2048 in any of the cells. If we can do this, we will win.



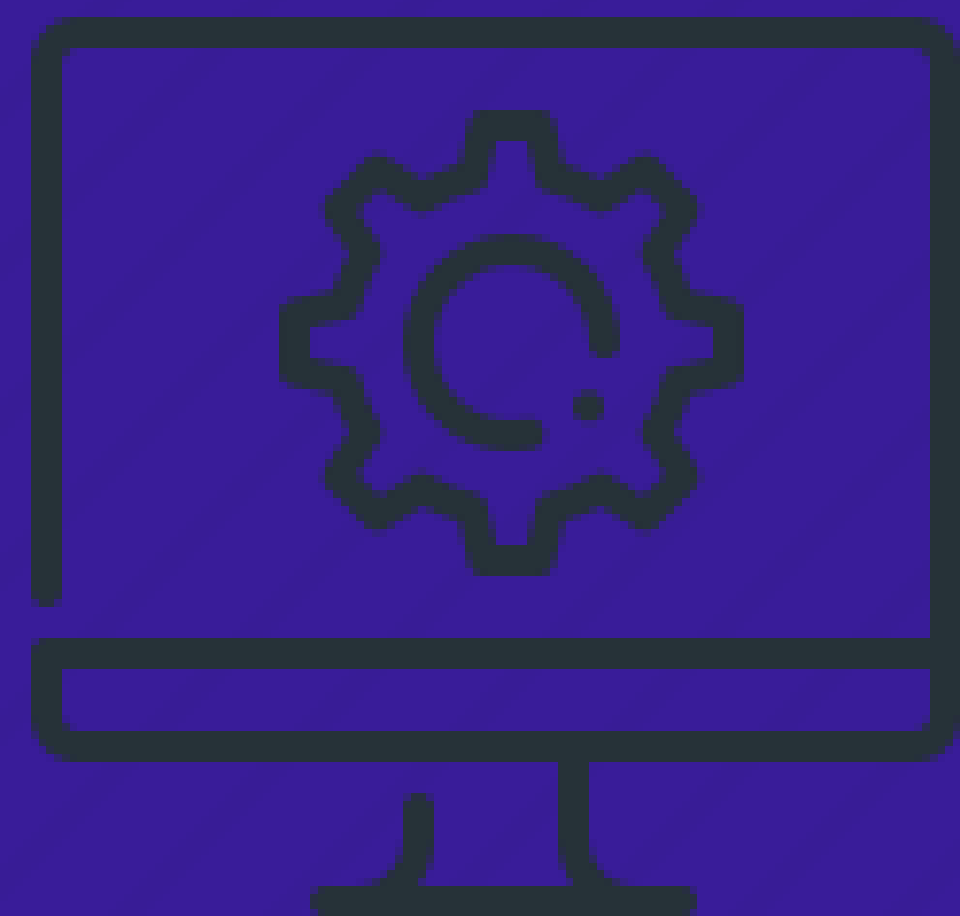
But if during the game there is no empty cell left to fill in the new 2, then you will lose.



A screenshot of a 2048 game window. The window has a title bar with a small icon and the text '2048'. The game board is a 4x4 grid of colored squares. The colors range from light beige for lower values to dark orange and yellow for higher values. The grid contains the following numbers: Row 1: 4, 2, 16, 2; Row 2: 32, 'You Lose!', 16; Row 3: 8, 128, 32, 4; Row 4: 2, 16, 4, 2. The 'You Lose!' message is in a grey square, indicating a game over state where no empty cells are available to place a new tile.

| | | | |
|----|-----------|----|---|
| 4 | 2 | 16 | 2 |
| 32 | You Lose! | 16 | |
| 8 | 128 | 32 | 4 |
| 2 | 16 | 4 | 2 |

Your screens show the part of the code that is responsible for the movement of blocks in the game. There are 4 directions here: right, left, up, and down.



```
92 # Логика движения
93
94 def up(game):
95     game = transpose(game)
96     game, done = cover_up(game)
97     game, done = merge(game, done)
98     game = cover_up(game)[0]
99     game = transpose(game)
100     return game, done
101
102
103 def down(game):
104     game = reverse(transpose(game))
105     game, done = cover_up(game)
106     game, done = merge(game, done)
107     game = cover_up(game)[0]
108     game = transpose(reverse(game))
109     return game, done
110
111
112 def left(game):
113     game, done = cover_up(game)
114     game, done = merge(game, done)
115     game = cover_up(game)[0]
116     return game, done
117
118
119 def right(game):
120     game = reverse(game)
121     game, done = cover_up(game)
122     game, done = merge(game, done)
123     game = cover_up(game)[0]
124     game = reverse(game)

# Команды движения
game.commands = {
    binds.up: up,
    binds.down: down,
    binds.left: left,
    binds.right: right,
    binds.w: up,
    binds.s: down,
    binds.a: left,
    binds.d: right,
    binds.num8_up: up,
    binds.num5_down: down,
    binds.num4_left: left,
    binds.num6_right: right,
}

game.grid_cells = []
game.cell_size()
game.matrix = new(4)
game.step = []
game.new_cells()

game.mainloop()
```

And now you can see the design of our game. There were separate colors written for each number. In order for the game to be easy to distinguish and for the game to be beautiful.



```
3  number_color = {
4      2:      "#7a7167",
5      4:      "#7a7167",
6      8:      "#f7f4f0",
7      16:     "#f7f4f0",
8      32:     "#f7f4f0",
9      64:     "#f7f4f0",
10     128:    "#f7f4f0",
11     256:    "#f7f4f0",
12     512:    "#f7f4f0",
13     1024:   "#f7f4f0",
14     2048:   "#f7f4f0",
15     4096:   "#756d64",
16     8192:   "#f7f4f0",
17     16384:  "#756d64",
18     32768:  "#756d64",
19     65536:  "#f7f4f0",
20 }
21
22  cell_color = {
23      2:      "#ebe0d5",
24      4:      "#ebddc5",
25      8:      "#f0af78",
26      16:     "#f59462",
27      32:     "#f57a5d",
28      64:     "#f55e3b",
29      128:    "#ebcd71",
30      256:    "#ebca60",
31      512:    "#ebc650",
32      1024:   "#ebc33f",
33      2048:   "#edc22f",
```

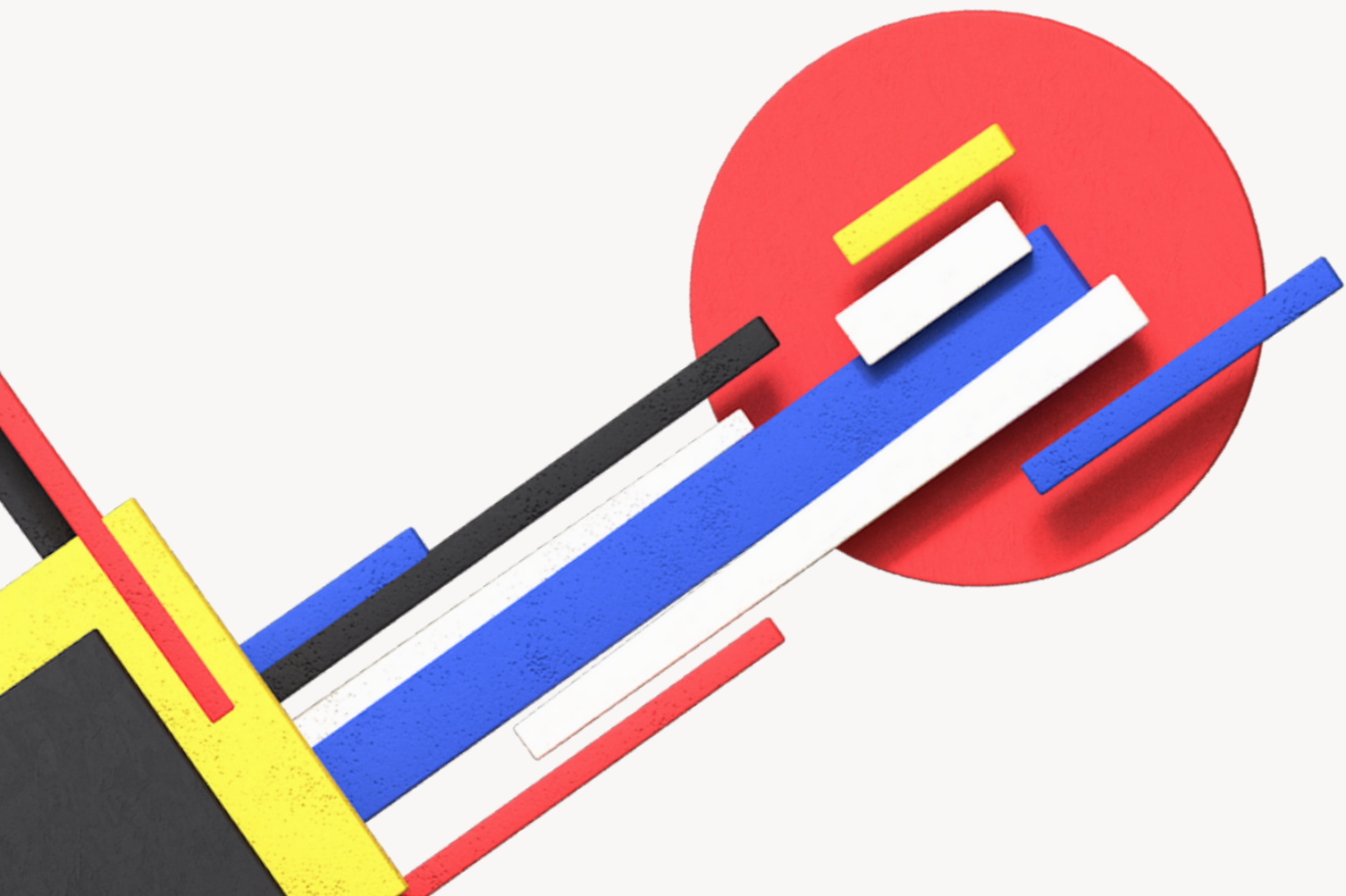
Here it was written when pressing the key, an action was performed. The player can play with 3 keyboard shortcuts. The first is the arrows. The second letter is a, b, c, D. Note these are the numbers 4, 6, 8, 5.



```
1 # Клавиши для выполнения задач
2
3 escape = "Escape"
4 back = "b"
5
6 up = "Up"
7 down = "Down"
8 left = "Left"
9 right = "Right"
10
11 w = "w"
12 s = "s"
13 a = "a"
14 d = "d"
15
16 num8_up = "8"
17 num5_down = "5"
18 num4_left = "4"
19 num6_right = "6"
```

THE IMPORTANCE OF THIS GAME

- 1 logical thinking
- 2 strategic planning skills
- 3 spatial imagination



Abstract geometric shapes in the top left corner, including a green parallelogram, a red triangle, a black rectangle, a blue rectangle, a white line, and a yellow rectangle.

**THANK YOU FOR
LISTENING**