

Convay Game Of Life

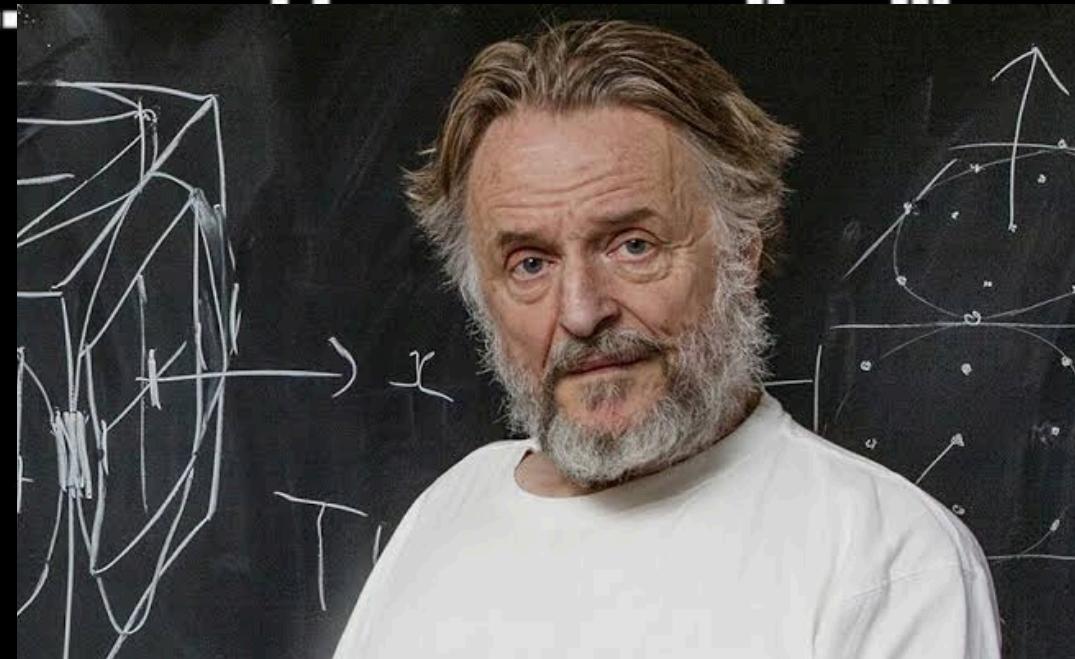
Skład zespołu:
Natalia Zagórska

11.06.2024

Koordynator:
Jakub Susoł



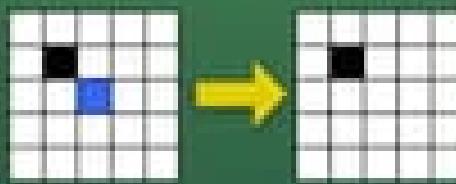
SciByteHub





Basic Rules of Conway's Game of Life

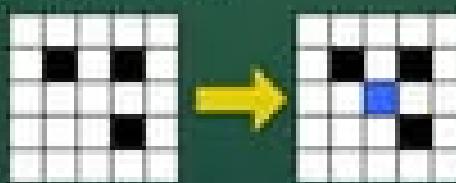
1. Living cells die if they have fewer than 2 neighbors (underpopulation/loneliness)



2. Living cells die if they have more than 3 neighbors (overpopulation)



3. Dead cells that have 3 neighbors become alive (reproduction)



4. Otherwise, there is no change (whether cell is alive or dead)



Cel

Celem projektu było zaimplementowanie słynnej gry Conway Game Of Life w języku Python



Pipeline

1. Import bibliotek
2. Stworzenie klasy Cell i Board
3. Zdefiniowanie odpowiednich metod
4. Stworzenie silnika gry
5. Obsługa błędów

Metody i narzędzia

Biblioteki:

- random
- time

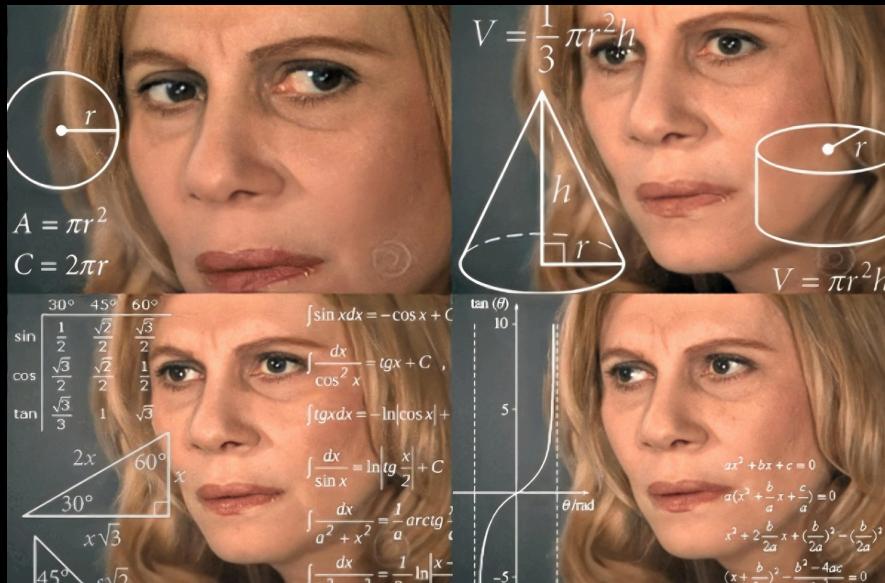
Programowanie obiektowe ale też funkcyjne

Inne narzędzia:





Jak było...



wrote a while(true) loop



With break statement right?



I wrote a while(true) loop

With break statement right?



I wrote a while(true) loop

With break statement right?



Kod



```
#libs
import random
import time
```

```
#defining classes
You, 1 godzinę temu | 1 author (You)
class Cell:
    def __init__(self, alive):
        self.alive = alive

    def update_status(self, alive_neighbors):
        if self.alive:
            if alive_neighbors < 2 or alive_neighbors > 3:
                self.alive = False
            else:
                if alive_neighbors == 3:
                    self.alive = True

    def __str__(self):
        return "■" if self.alive else " "
```



```
File: C:\Users\Arun\OneDrive\PyCharm\PyCharm 2020.1.3\scibytehub\src\main\java\com\scibytehub\cell\Board.java
class Board:
    def __init__(self, width, height):
        self.width = width
        self.height = height
        self.grid = [[Cell(random.choice([True, False])) for x in range(self.width)] for y in range(self.height)]

    def count_alive_neighbors(self, x, y):
        alive_neighbors = 0

        for cell_in_x in range(x - 1, x + 2):
            for cell_in_y in range(y - 1, y + 2):
                if (cell_in_x == x and cell_in_y == y) or cell_in_x < 0 or cell_in_y < 0 or cell_in_x >= self.height or cell_in_y >= self.height:
                    continue
                if self.grid[cell_in_x][cell_in_y].alive:
                    alive_neighbors += 1
        return alive_neighbors
```



```
def update_board(self):
    new_grid = [[Cell(random.choice([False])) for x in range(self.width)] for y in range(self.height)]
    for x in range(self.height):
        for y in range(self.width):
            alive_neighbors = self.count_alive_neighbors(x, y)
            cell = self.grid[x][y]
            new_cell = Cell(cell.alive)
            new_cell.update_status(alive_neighbors)
            new_grid[x][y] = new_cell
    self.grid = new_grid

def display(self):
    for row in self.grid:
        print(" ".join(str(cell) for cell in row))

def __str__(self):
    return '\n'.join(' '.join(str(cell) for cell in row) for row in self.grid)
```



```
#engine
if __name__ == "__main__":
    print(f"Hello user ! Welcome to the Conway Game Of Life")
    while True:
        try:
            width = int(input(f"Enter width of game board: "))
            height = int(input(f"now enter height of game board: "))
            if width <= 0 or height <= 0:
                raise ValueError("Width and height must be positive integers")
            break
        except ValueError as error:
            print(f"Invalid input: {error}. Please enter positive integer values")

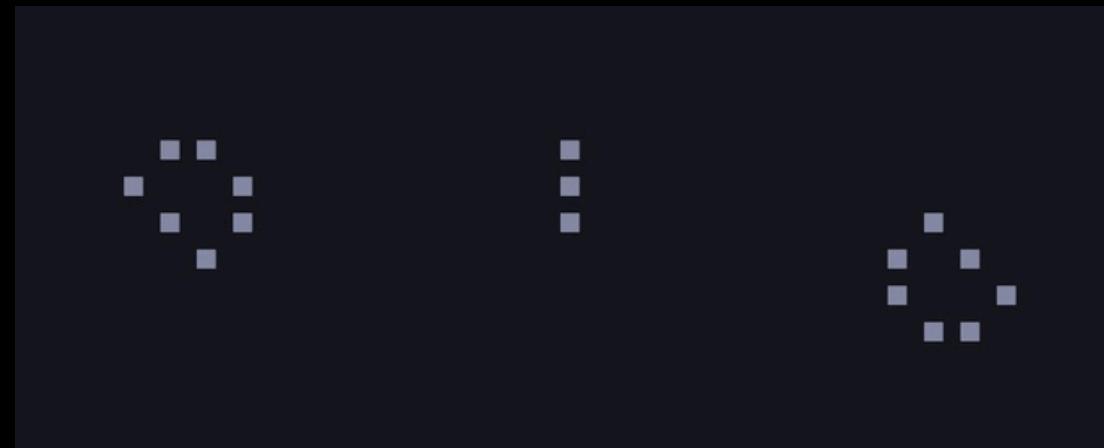
    board = Board(width, height)
    print("Initial status of cells: ")
    board.display()
```

```
while True:
    command = input((f"In a moment game of life will begin, if you would like to quit the game press ctrl + c at once. Begin ? (yes/no):"))
    if command.lower() == "yes":
        while True:
            print(f"\n next generation: ")
            board.update_board()
            board.display()
            time.sleep(2)
            update = input(f"Press ENTER if you wish to continue or q to quit the game: ")
            if update == 'q':
                print("Exiting the game. Goodbye!")
                exit(0)
    elif command == 'q' or command == 'no':
        print("Exiting the game. Goodbye!")
        exit(0)
    else:
        print("Invalid input. Please type 'yes' to start or 'q' to quit.")
```



SciByteHub

Demo



Wnioski

Kod działa prawidłowo, jest to uproszczona wersja
która można by zwizualizować dla lepszego UX np.
za pomocą matplotlib lub turtle



DZIĘKUJE ZA UWAGĘ!



SciByteHub