УДК 004.624

Реализация виртуального аквариума на языке программирования Python

Кизянов Антон Олегович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье описан процесс создания виртуального аквариума с виртуальными рыбами. Для создания используется язык программирования Python. Созданное приложение служит наглядным пояснением как можно создать анимацию с заданной логикой.
Ключевые слова: Python, аквариум

Implementing a virtual aquarium in the Python programming language

Kizyanov Anton Olegovich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article describes the process of creating a virtual aquarium with virtual fish. For creation, the programming language Python is used. The created application serves as a visual explanation of how you can create an animation with a given logic.
Keywords: Python, aquarium

Виртуальные рыбки в виртуальном аквариуме с пузырьками воздуха и водорослями. Каждый раз, когда запускают программу, она случайным образом генерирует рыбу, используя разные типы и цвета рыб.

Цель исследования – написать приложение, имитирующее виртуальных рыб в виртуальном аквариуме на языке программирования Python.

Ранее этим вопросом интересовались М.М. Белолюбский развивал тему «Возможности модуля pygame языка программирования python для детей» [1] в которой рассматриваются особенности модуля Pygame языка программирования Python. Рассматриваемый модуль предназначен для создания игр и позволяет выработать навыки по основам создания игр. Проведен анализ основных команд, возможностей и перспектив данного модуля. Он отлично подойдет для детей, которые знакомятся со структурой программирования компьютерных игр. А.М. Петров, О.А. Лобастова с темой «Программирование на python» [2], а подробнее про возможность языка Python быстро разрабатывать программы различного уровня

сложности. В работе рассмотрен пример написания игры «Отгадай число» в среде IDLE. Приведены алгоритм, код на языке Python и результат выполнения программы. А.Г. Сиденко опубликовал статью «Использование игровых сред для обучения программированию» [3] в статье демонстрируется использование подходов, связанных с геймификацией являются одним из перспективных направлений развития методики преподавания информатики. В основе такого подхода лежат принципы игровой механики и возможностей виртуального мира в рамках игры по определенному сценарию. В статье рассмотрены основные подходы для создания игрового мира и обучения с помощью изучения языка программирования Python основных возможностей для его исследования. В качестве примера приведены основы работы в игровом мире Minecraft и элементы обучения программированию на языке Python.

```
import random
import sys
import time

import bext

span, peak = bext.size()
span -= 1
num_seaweed = 2
num_chum = 10
num_ballon = 1
fps = 30
fish_category = [{'right': ['><>'], 'left': ['<><']}, {'right': ['>||>'],
'left': ['<||<']},
                 {'right': ['>))>'], 'left': ['<[[<']}, {'right': ['>||o',
'>||.'], 'left': ['o||<', '.||<']},
                 {'right': ['>))o', '>)).'], 'left': ['o[[<', '.[[<']},
{'right': ['>-==>'], 'left': ['<==-<']},
                 {'right': [r'>\\>'], 'left': ['<//<']}, {'right':
['><)))*>'], 'left': ['<*(((><']},
                 {'right': ['}-[[[*>'], 'left': ['<*]]]-{']}, {'right': [']-
<)))b>'], 'left': ['<d(((>-[']},
                 {'right': ['><XXX*>'], 'left': ['<*XXX><']},
                 {'right': ['_.-._.-^=>', '.-._.-.^=>', '-._.-._^=>', '._.-
._.^=>'],
                  'left': ['<=^-._.-._', '<=^.-._.-.', '<=^_.-._.-', '<=^._.-
._.']}, ]

high_len_fish = 10
left_end = 0
right_end = span - 1 - high_len_fish
top_end = 0
bottom_end = peak - 2


def main():
    global chums, ballons, sparkles, seaweeds, stride
    bext.bg('black')
    bext.clear()
    chums = []

    for i in range(num_chum):
        chums.append(generateFish())
```

```
    ballons = []
    for i in range(num_ballon):
        ballons.append(random.randint(left_end, right_end))
    sparkles = []

    seaweeds = []
    for i in range(num_seaweed):
        kelpx = random.randint(left_end, right_end)
        kelp = {'x': kelpx, 'segments': []}
        for i in range(random.randint(6, peak - 1)):
            kelp['segments'].append(random.choice(['(', ')']))
        seaweeds.append(kelp)

    stride = 1
    while True:
        simulateAquarium()
        drawAquarium()
        time.sleep(1 / fps)
        clearAquarium()
        stride += 1


def getRandomColor():
    return random.choice(('black', 'red', 'green', 'yellow', 'blue',
'purple', 'cyan', 'white'))


def generateFish():
    fishType = random.choice(fish_category)
    colorPattern = random.choice(('random', 'head-tail', 'single'))
    fishLength = len(fishType['right'][0])
    if colorPattern == 'random':
        colors = []

        for i in range(fishLength):
            colors.append(getRandomColor())
    if colorPattern == 'single' or colorPattern == 'head-tail':
        colors = [getRandomColor()] * fishLength

    if colorPattern == 'head-tail':
        headTailColor = getRandomColor()

        colors[0] = headTailColor
        colors[-1] = headTailColor
    fish = {'right': fishType['right'], 'left': fishType['left'], 'colors':
colors, 'hSpeed': random.randint(1, 6),
            'vSpeed': random.randint(5, 15), 'timeToHDirChange':
random.randint(10, 60),
            'timeToVDirChange': random.randint(2, 20), 'goingRight':
random.choice([True, False]),
            'goingDown': random.choice([True, False])}

    fish['x'] = random.randint(0, span - 1 - high_len_fish)
    fish['y'] = random.randint(0, peak - 2)

    return fish


def simulateAquarium():
    global chums, ballons, sparkles, KELP, stride
    for fish in chums:
        if stride % fish['hSpeed'] == 0:
            if fish['goingRight']:
                if fish['x'] != right_end:
```

```
                    fish['x'] += 1
                else:
                    fish['goingRight'] = False
                    fish['colors'].reverse()
            else:
                if fish['x'] != left_end:
                    fish['x'] -= 1
                else:
                    fish['goingRight'] = True
                    fish['colors'].reverse()
        fish['timeToHDirChange'] -= 1
        if fish['timeToHDirChange'] == 0:
            fish['timeToHDirChange'] = random.randint(10, 60)
            fish['goingRight'] = not fish['goingRight']

        if stride % fish['vSpeed'] == 0:
            if fish['goingDown']:
                if fish['y'] != bottom_end:
                    fish['y'] += 1
                else:
                    fish['goingDown'] = False
            else:
                if fish['y'] != top_end:
                    fish['y'] -= 1
                else:
                    fish['goingDown'] = True

        fish['timeToVDirChange'] -= 1
        if fish['timeToVDirChange'] == 0:
            fish['timeToVDirChange'] = random.randint(2, 20)
            fish['goingDown'] = not fish['goingDown']

    for bubbler in ballons:
        if random.randint(1, 5) == 1:
            sparkles.append({'x': bubbler, 'y': peak - 2})

    for bubble in sparkles:
        diceRoll = random.randint(1, 6)
        if (diceRoll == 1) and (bubble['x'] != left_end):
            bubble['x'] -= 1
        elif (diceRoll == 2) and (bubble['x'] != right_end):
            bubble['x'] += 1

        bubble['y'] -= 1

    for i in range(len(sparkles) - 1, -1, -1):
        if sparkles[i]['y'] == top_end:
            del sparkles[i]

    for kelp in seaweeds:
        for i, kelpSegment in enumerate(kelp['segments']):
            if random.randint(1, 20) == 1:
                if kelpSegment == '(':
                    kelp['segments'][i] = ')'
                elif kelpSegment == ')':
                    kelp['segments'][i] = '('


def drawAquarium():
    global chums, ballons, sparkles, KELP, stride
    bext.fg('white')
    bext.goto(0, 0)
    bext.fg('white')
    for bubble in sparkles:
```

```
        bext.goto(bubble['x'], bubble['y'])
        print(random.choice(('o', 'O')), end='')

    for fish in chums:
        bext.goto(fish['x'], fish['y'])
        if fish['goingRight']:
            fishText = fish['right'][stride % len(fish['right'])]
        else:
            fishText = fish['left'][stride % len(fish['left'])]
        for i, fishPart in enumerate(fishText):
            bext.fg(fish['colors'][i])
            print(fishPart, end='')

    bext.fg('green')
    for kelp in seaweeds:
        for i, kelpSegment in enumerate(kelp['segments']):
            if kelpSegment == '(':
                bext.goto(kelp['x'], bottom_end - i)
            elif kelpSegment == ')':
                bext.goto(kelp['x'] + 1, bottom_end - i)
            print(kelpSegment, end='')

    bext.fg('yellow')
    bext.goto(0, peak - 1)
    print(chr(9617) * (span - 1), end='')
    sys.stdout.flush()


def clearAquarium():
    global chums, ballons, sparkles, KELP
    for bubble in sparkles:
        bext.goto(bubble['x'], bubble['y'])
        print(' ', end='')

    for fish in chums:
        bext.goto(fish['x'], fish['y'])
        print(' ' * len(fish['left'][0]), end='')

    for kelp in seaweeds:
        for i, kelpSegment in enumerate(kelp['segments']):
            bext.goto(kelp['x'], peak - 2 - i)
            print(' ', end='')

    sys.stdout.flush()


if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        sys.exit()
```

Результат запуска кода выше можно увидеть на рисунке 1.

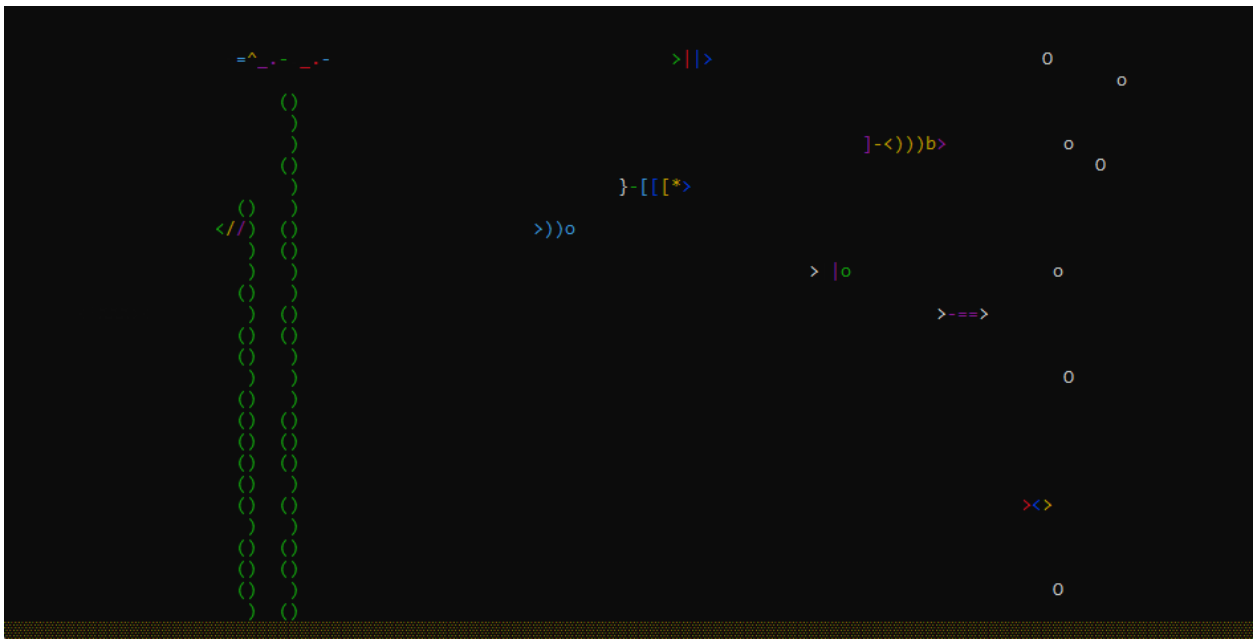

Рис. 1 Аквариум на Python

Вывод

В этой статье был реализован виртуальный аквариум с виртуальными рыбами. Приложение позволяет бесконечно смотреть за жизнью виртуальных рыб и их поведением. Благодаря этому приложению можно разобраться в построении анимации в простом консольном окне.

Библиографический список

1. Белолюбский М.М. Возможности модуля pygame языка программирования python для детей // В сборнике: Наука современности: проблемы и решения. Сборник научных статей. Москва, 2021. С. 24-27. URL: https://www.elibrary.ru/item.asp?id=47113254 (Дата обращения: 05.01.2022)
2. Петров А.М., Лобастова О.А. Программирование на python // В сборнике: Информатика и вычислительная техника. сборник научных трудов. Б.М. Калмыков (гл. редактор); Н.В. Первова (отв. редактор). Чебоксары, 2015. С. 91-94. URL: https://www.elibrary.ru/item.asp?id=24670280 (Дата обращения: 05.01.2022)
3. Сиденко А.Г. Использование игровых сред для обучения программированию // В сборнике: Информатизация непрерывного образования - 2018. материалы Международной научной конференции: в 2 т.омах. Под общей редакцией В. В. Гриншкуна. 2018. С. 491-496. URL: https://www.elibrary.ru/item.asp?id=40651541 (Дата обращения: 05.01.2022)