

Программирование на Python

Урок №15-16

План урока:

Самостоятельный проект:

Мы приступаем к созданию итогового проекта, который подытожит все темы, которые мы прошли за эти 4 месяца. Основой нашей будущей игры будут два двумерных списка. На одном будет отображаться игровое поле, а на другой будет виден только компьютеру.

Из данного проекта можно сделать морской бой, поиск клада, раскопку сокровищ и вообще любую игру, где с помощью выбор двух координат нужно что-то обнаружить на карте. Мы будем рассматривать для примера морской бой.

Точки в первой таблице означают, что это пока еще не открытая клетка. А вторая таблица показывает что под каждой клеткой скрыто.

	0	1	2	3	4
0
1
2
3
4

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Давайте расположим корабль в 4 клетки в строке номер 1.

	0	1	2	3	4
0
1
2
3
4

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Теперь попробуем представить как будет выглядеть выстрел в клетку 3 строки 0 столбца.

	0	1	2	3	4
0
1
2
3	X
4

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	1
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Попробуем выстрелить в строку 1 столбец 2.

	0	1	2	3	4
0
1	.	.	O	.	.
2
3	X
4

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	-1	1	1
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Клетка с единицей меняется на -1. Это будет означать, что наш корабль подбит.

Игровой цикл:

Мы можем заметить, что какие-то изменения в нашей игре происходят только после того, как пользователь решит куда-то выстрелить. Пусть у пользователя будет 10 выстрелов, чтобы потопить корабль. Попробуем написать программу, которая будет создавать два вложенных списка, один из которых заполнен нулями, а другой точками.

```

pole = []
karta = []
i = 0
j = 0
razmer = 5
while j < razmer:
    new = []
    i = 0
    while i < razmer:
        new.append(0)
        i = i + 1
    pole.append(new)
    j = j + 1
while j < razmer:
    new = []
    i = 0
    while i < razmer:
        new.append(".")
        i = i + 1
    karta.append(new)
    j = j + 1

```

Данные циклы очень похожи и отличаются только тем, чем мы заполняем связанный список. Вынесем это в отдельную функцию.

```

def zapolnit(spisok, simbol, razmer):
    i = 0
    j = 0
    while j < razmer:
        new = []
        i = 0
        while i < razmer:
            new.append(simbol)
            i = i + 1
        spisok.append(new)
        j = j + 1

pole = []
karta = []
razmer = 5
zapolnit(pole, 0, razmer)
zapolnit(karta, ".", razmer)

```

Теперь создадим функцию, которая будет отображать список на экран в виде таблицы.

```

def zapolnit(spisok, simbol, razmer):
    i = 0
    j = 0
    while j < razmer:
        new = []
        i = 0
        while i < razmer:
            new.append(simbol)
            i = i + 1
        spisok.append(new)
        j = j + 1

def pokazat(spisok, razmer):
    i = 0
    while i < razmer:
        print(spisok[i])
        print()
        i = i + 1

pole = []
karta = []
razmer = 5
zapolnit(pole, 0, razmer)
zapolnit(karta, ".", razmer)
pokazat(pole, razmer)
pokazat(karta, razmer)

```

Теперь нам необходимо ввести игровой цикл. Наш цикл while будет просить пользователя ввести число, пока у него есть попытки и проверять с помощью новой функции попал пользователь или нет.

```

def popal(pole, karta, coord1, coord2):
    if pole[coord1][coord2] == 1:
        karta[coord1][coord2] = "O"
    if pole[coord1][coord2] == 0:
        karta[coord1][coord2] = "X"

pole = []
karta = []
razmer = 5
zapolnit(pole, 0, razmer)
zapolnit(karta, ".", razmer)
pokazat(karta, razmer)
popitki = 0
while popitki < 10:
    print("Введите строчку")
    coord1 = int(input())
    print("Введите столбец")
    coord2 = int(input())
    popal(pole, karta, coord1, coord2)
    popitki = popitki + 1
    pokazat(karta, razmer)

```

Теперь осталось добавить только функцию размещения кораблей и условия победы.

Наша программа выполнена!