

С#. Подведение итогов, try-catch, Files, JSON. Игра

Лектор: Юрий Коноплев

01.08.19

01.08.19 расписание дня

Время	Что делаем?
10:00 – 12:15	Лекция <ul style="list-style-type: none">▪ подведение итогов▪ Исключения▪ Работа с файлами▪ JSON
12:15 – 13:00	Постановка задачи на день Лектор общается с каждой командой и вместе пропишите потенциальные задачи на день
13:00 – 14:00	Обед
14:00 – 16:45	Работаем над задачами
16:45 – 17:00	Подведение итогов

Игра "Охотники за Вампусом"

```
xxxxx
.....
xxxxx
.....
xxPxx
.....
xxxxx
.....
xxxxx
.....
```

x - клетки которые игрок видит

x1 .. x24

// $2^5 = 32$

x1 x2 x3 x4 x5

x6 x7

.. .. P

..

.. x24

x_i = 1 если в x_i находится монстр (m)

Игра "Охотники за Вампусом"

```
=====  
часть 2  
=====
```

```
игрок имеет
```

- число жизней
- ориентацию / направление - т.е. куда он смотрит

```
игрок на каждом ходу может делать одно из действий:
```

- a : идти на влево (команда - нажатие клавиши a)
- w : идти на вверх (команда - нажатие клавиши w)
- d : идти на право (команда - нажатие клавиши d)
- s : идти на вниз (команда - нажатие клавиши s)
- f : стрелять (f - fire)
- n : none - не делать ничего

```
всего 6 команд
```

```
//  $2^3 = 8$ 
```

```
* в начале реализации можно без fire
```

Игра "Охотники за Вампусом"

=====

часть 3

.....

=====

игрок имеет byte[] dna = [1,0,1,0,0,1,0, ...,] - битовая последовательность
.....
описывающая его действия
.....

формат:

.....

- b1 b2 b3 b4 b5 b6 b7 b8 b9...
- [b1 b2 b3] [b4 b5 b6] [b7 b8 b9] ...
- A1 A2 A3 ...

если в x1 монстор то игрок делает действие A1

.....

если в x2 монстор то игрок делает действие A2

.....

если в x3 монстор то игрок делает действие A3

.....

...

.....

Игра "Охотники за Вампусом"

=====

часть 3

.....

=====

игрок имеет byte[] dna = [1,0,1,0,0,1,0, ...,] - битовая последовательность
описывающая его действия
.....

формат:

.....

- b1 b2 b3 b4 b5 b6 b7 b8 b9...
- [b1 b2 b3] [b4 b5 b6] [b7 b8 b9] ...
- A1 A2 A3 ...

если в x1 монстор то игрок делает действие A1

.....

если в x2 монстор то игрок делает действие A2

.....

если в x3 монстор то игрок делает действие A3

.....

...

.....

Игра "Охотники за Вампусом"

3 бит хватит чтобы закодировать действия {a,w,d,s,f,n}

например:

a = 0 000

w = 1 001

d = 2 010

s = 3 011

f = 4 100

n = 5 101 // не делать ничего

n = 6 110 // не делать ничего

n = 7 111 // не делать ничего

размер dna $24 \times 3 + 6 \times 3$ битов, т.е. 90 битов

первые 24×3 отвечают за действия в случае если игрок видит монстра

последние 6×4 за действия того как игрок ходит по карте если монстра не видит

Игра "Охотники за Вампусом"

3 бит хватит чтобы закодировать действия {a,w,d,s,f,n}

например:

a = 0 000

w = 1 001

d = 2 010

s = 3 011

f = 4 100

n = 5 101 // не делать ничего

n = 6 110 // не делать ничего

n = 7 111 // не делать ничего

размер dna $24 \times 3 + 6 \times 3$ битов, т.е. 90 битов

первые 24×3 отвечают за действия в случае если игрок видит монстра

последние 6×4 за действия того как игрок ходит по карте если монстра не видит

Игра “Охотники за Вампусом”

=====

часть 5

=====

- фитнес функция $fitness = (\text{количество убитых монстров}) * 2 + \text{количество сделанных ходов}$
- игроку даётся 100 ходов
- размер популяции - 1000 особей
- особи сортируются по значению от фитнес функции и далее выбираются первые 100
- лучшие особи скрещиваются между собой так чтобы стало 1000 особей
- для каждой особи вызывается функция мутация которая с вероятностью p изменяет каждый бит dna

Игра “Охотники за Вампусом”

=====

часть 6

.....

=====

запуск игры для любой особи

.....

ССЫЛКИ

- try-catch

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/try-catch>

- <https://www.tutorialsteacher.com/csharp/csharp-tutorials>

- <https://repl.it/languages/csharp>

- Func and delegate

<https://www.growingwiththeweb.com/2012/08/func-and-action-basics-in-c.html>

- https://github.com/j0k/it_school_weeks

try catch

```
try
{
    // code that may raise exceptions
}
catch(Exception ex)
{
    // handle exception
}
finally
{
    // final cleanup code
}
```

try catch

```
class TryFinallyTest
{
    static void ProcessString(string s)
    {
        if (s == null)
        {
            throw new ArgumentNullException();
        }
    }

    static void Main()
    {
        string s = null; // For demonstration purposes.

        try
        {
            ProcessString(s);
        }
        catch (Exception e)
        {
            Console.WriteLine("{0} Exception caught.", e);
        }
    }
}
```

Files

- <https://support.microsoft.com/en-us/help/304430/how-to-do-basic-file-i-o-in-visual-c>
- <https://docs.microsoft.com/en-us/dotnet/api/system.io.file?view=netframework-4.8>
- http://mycsharp.ru/post/21/2013_06_12_rabota_s_fajlami_v_si-sharp_klassy_streamreader_i_streamwriter.html

JSON

- <http://streletzcoder.ru/rabotaem-s-json-v-c-serializatsiya-i-deserializatsiya/>
- <https://progtask.ru/json-c-sharp/>
- <https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/how-to-serialize-and-deserialize-json-data>