

Faculty of Mathematics and Physics
Charles University in Prague
10th March 2016



C# Made Easy!

Programming II

Workshop 3 – Unleashing the Snake

Workshop 3

Outline

1. Test
2. Next workshop...
3. The Snake (as homework)



Test 03

Quick Warm up!

Find the test here (no-ads):

<https://goo.gl/R108VL>

Permanent link:

https://docs.google.com/forms/d/1Wp0Hw3b9B30wEI2_6IcX81yVbhtXp0Edpm0phu135RY/viewform

Time for the test:

15 min

Next workshop?

Doodle

I am not in Prague on 17.3.2016 ...

Please fill this Doodle out and mark your preferences on compensatory workshop

- <http://doodle.com/poll/pk68bn3d5mnsgn8s>

Revisiting the Homework

What is the benefit of having interfaces?

Discussing “Simple Calc” example.

- One big file
 - Not a reusable implementation
- OOPish 1
 - No interface for Calc to work with
 - Not a reusable implementation
- OOPish 2
 - Static methods are not OOP
 - Not a reusable implementation
- OOP way
 - Sort of ok...

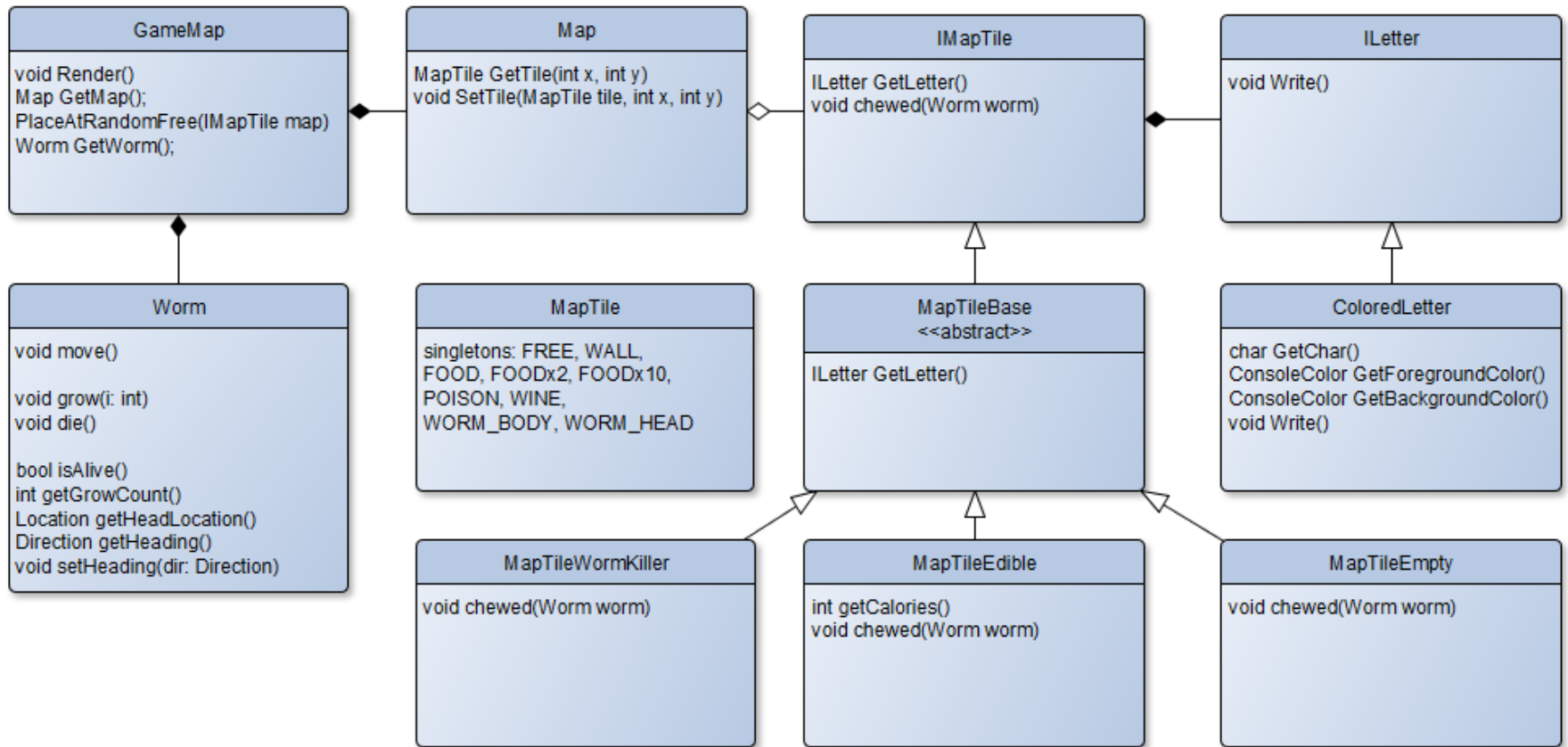
Assignment 3

Create "The Snake" aka "The Worm" game

- Console application (text-based game)
 - Snake is controlled either by WSAD, or arrows, or 8246 numpad numbers
 - You have to display "snake's length" and "time elapsed" somewhere at "bottom status line"
 - Escape key terminates the game
 - Implement tiles: Empty, Wall, Food, Posion, Wine, "Snake"
 - Snake movement speed should be increasing every N food eaten (regardless their calories)
- Provide solution that can easily change
 - Map dimensions
 - Ask me on how big map I want to play (max 80 x 40)
 - Food effects
 - Adding new food should be as easy as creating and „registering“ a new Food object
 - Both POISON and WALL kill the snake
 - Implement WINE so it reverse the controls (UP<->DOWN, LEFT<->RIGHT)
- Bonus:
 - Provide different visualization for the snake's body using
 - <, v, >, ^ as a head ... -, | as a straight body and / \ at "turning points"
 - 5 points

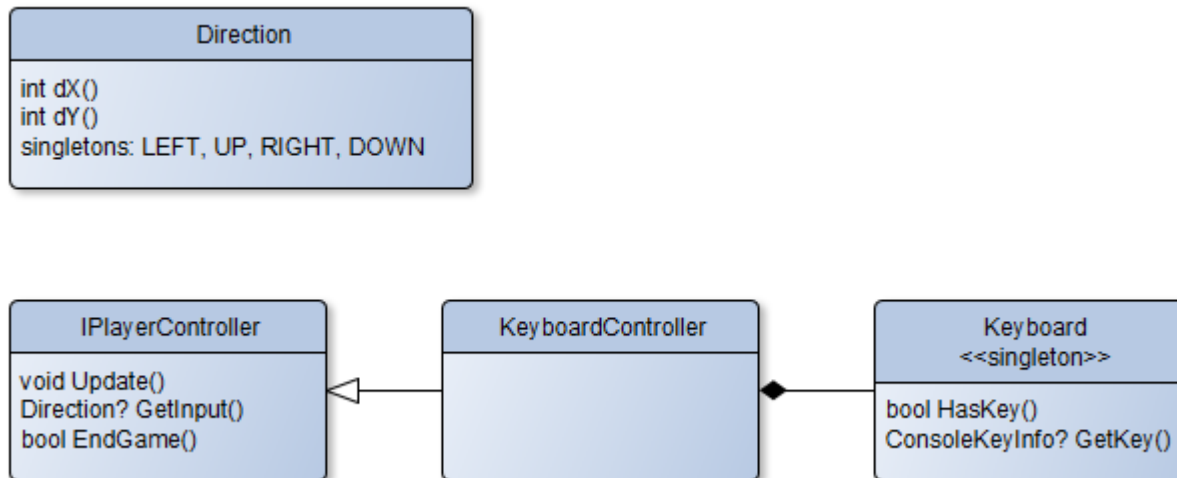
Assignment 3

Representing the Game Model + View



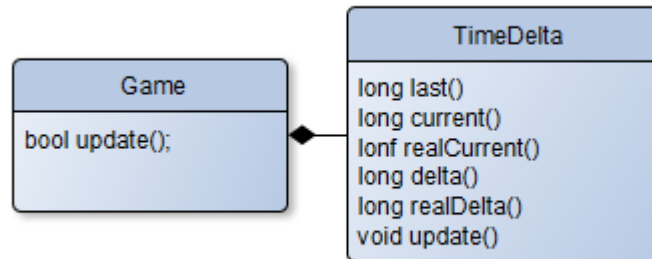
Assignment 3

Getting Player's Input



Assignment 3

Timing the Game



CheatSheet

Reading Inputs from Keyboard

```
1 reference  
public bool hasKey()  
{  
    return Console.KeyAvailable;  
}
```

```
1 reference  
public ConsoleKey? getKey()  
{  
    if (Console.KeyAvailable)  
    {  
        return Console.ReadKey(true).Key;  
    }  
    return null;  
}
```

CheatSheet

Changing Console Output Color

- Google: C# Console Colors
 - Google – The Best Programmer's Friend
 - Keep in mind the limit of "Googling" for "Code"

CheatSheet

Timing your Snake

```
private static readonly DateTime Jan1st1970 = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
```

2 references

```
public static long CurrentTimeMillis()  
{  
    return (long)(DateTime.UtcNow - Jan1st1970).TotalMilliseconds;  
}
```

Assignment 3

Start coding the Snake

- Follow the UML
- Create STUB classes (each class within own file)
 - STUB == fields and methods declared, constructors prepared, but w/o implementation
- If you feel like it, try to implement something, but this assignment is actually about reading UML, declaring classes and interfaces in C#
- 10 points

Assignment 3

Send me an email

- Email: jakub.gemrot@gmail.com
- Subject: **Programming II – 2016 – Assignment 03**
- Zip up the whole project and send it
 - If Google mumbles something about invalid files within zip, just rename the extension to .zi_ ... this would fool it ;-)
- You WILL NOT find the assignment in CoDex!
- Deadline: **17.3.2016 23:59**

Questions?

I sense a soul in search of answers...

- In case of doubts about the assignment or some other problems don't hesitate to contact me!
 - Jakub Gemrot
 - jakub.gemrot@gmail.com