

# Advanced Programming C#

## Lecture 1

dr inż. Małgorzata Janik  
[malgorzata.janik@pw.edu.pl](mailto:malgorzata.janik@pw.edu.pl)

Winter Semester 2019/2020

# Organizational issues

- **Lecture + laboratories + project:**

- dr inż. Małgorzata Janik  
Zakład Fizyki Jądrowej  
pok. 117D, Gmach Fizyki

[majanik@if.pw.edu.pl](mailto:majanik@if.pw.edu.pl)

- **Time:**

- Monday, 10:00 – 12:00
- Monday, 12:00 – 14:00

- **Webpage:**

- [www.if.pw.edu.pl/~majanik/wiki](http://www.if.pw.edu.pl/~majanik/wiki)

- **Office hours, 117D GF:**

- Monday, 14:00 – 15:00
- Wednesday 15:00-16:00

# Organizational issues

---

- **Final grades:**
  - Laboratories: 60% of the grade
  - Project: 40% of the grade
- **Laboratories:**
  - 14 classes: 1 instructional, 10 graded, 3 project-related
  - used software: Visual Studio Community
  - classes duration: 90 minutes (no break)
- **Projects:**
  - Project presentation on 6<sup>th</sup>, 10<sup>th</sup> and 14<sup>th</sup> classes

# Conditions to pass the classes (1)

- **Laboratories:**

- 10 classes of diversified level (**0-6 pkt each**)
- during classes you can use any printed materials, your own programs, as well as resources available in the Internet\*
- program can be graded at any point in time during classes
- program finished at home: up to **+3 pkt**
  - finished program must be presented in the beginning of next class

\*) it is forbidden to use mailboxes, messangers, social networks or programs written by other students, as well as phones, tablets etc. to communicate with others.

- **Absences:**

- max 2 unjustified absences are allowed (**0 pkt**)
- in case of justified absence student can finish program at home and show it to tutor during the office hours latest two weeks after return (**max 5 pkt**)

# Conditions to pass the classes (2)

---

- **Project:**
  - grading: **0-40 pkt** for the project
  - During the semester there will be **2 intermediate stages**, when the current status of the project should be presented
  - Each intermediate stage: **0-10 pkt**
  - Final project (should be shown in the last class): **0-20 pkt**
  - To pass the subject **>50% of the points from the project** should be acquired (minimal project requirements should be completed)

# Project proposals

---

- Simulation of several simple physics experiments
- Simulation of the interaction of the radiation with matter
- Main building path finder: application showing the shortest path between two rooms in the Warsaw University of Technology Main Building
- Network Messenger
- Simple RPG game
- Simple platform game

# Project proposals

- Simu
- Simu
- Main  
betwe  
Main
- Netw
- Simp
- Simp



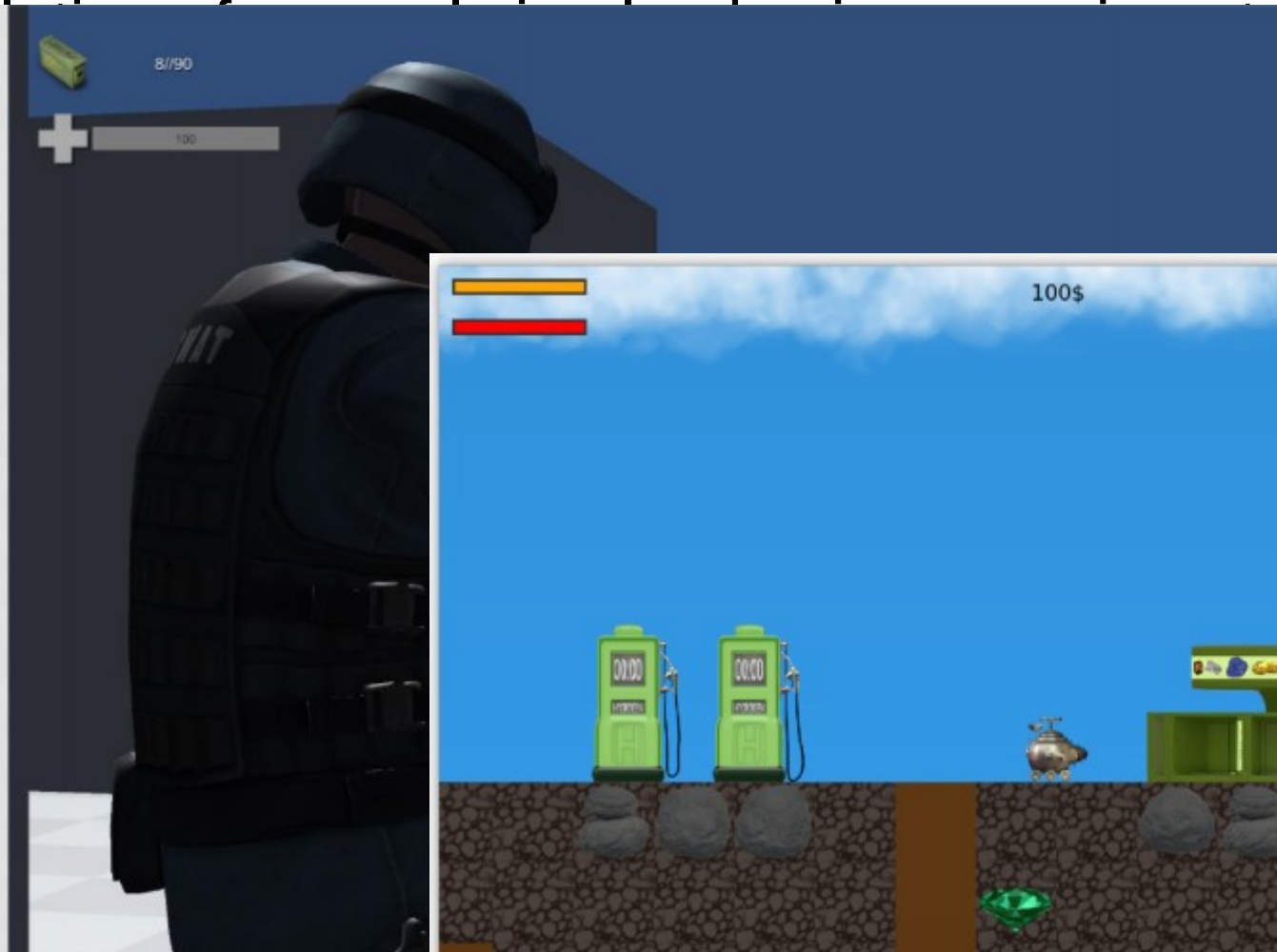
natter

shortest path  
echnology

**Shooter**  
Movement  
Shooting  
Death  
Opponents AI

# Project proposals

- Simu
- Simu
- Main
- between
- Main
- Netw
- Simp
- Simp



natter

shortest path

## MarsMiner

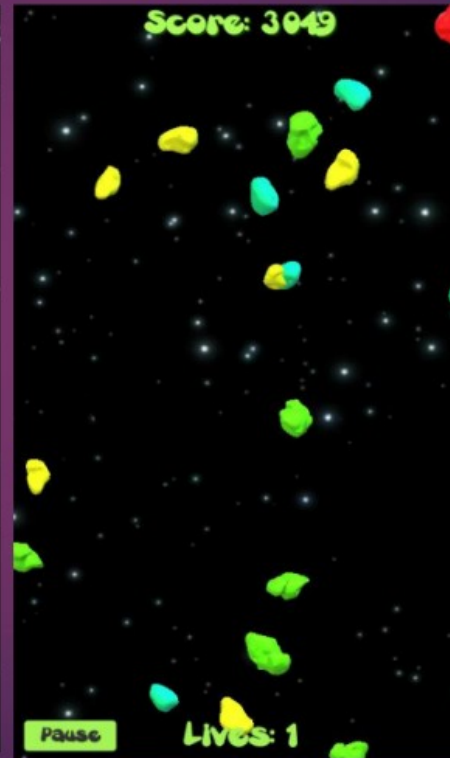
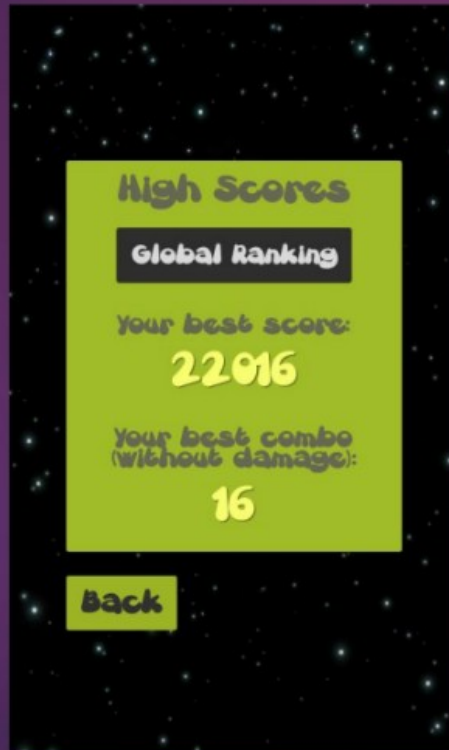
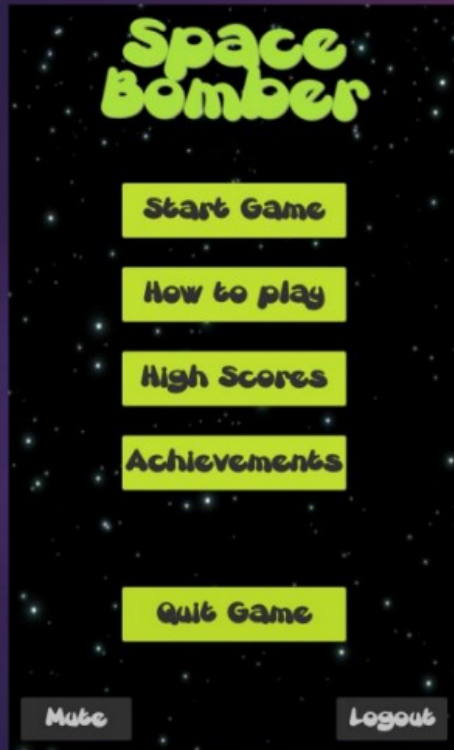
Tile generation  
Selling Minerals  
Town Buildings  
Buttons, Mouse Support



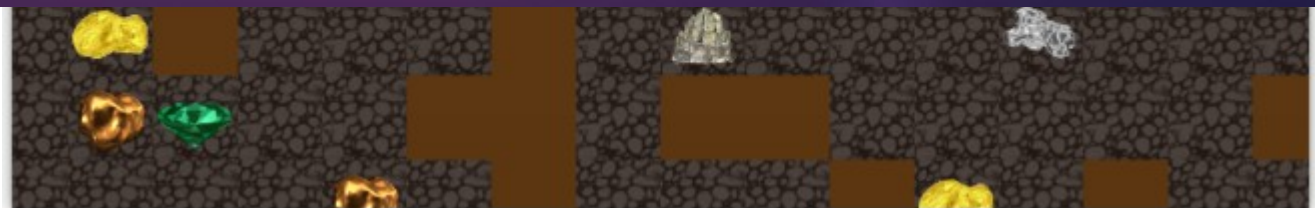


# Project proposals

- Simulation



The generation  
Selling Minerals  
Town Buildings  
Buttons, Mouse Support



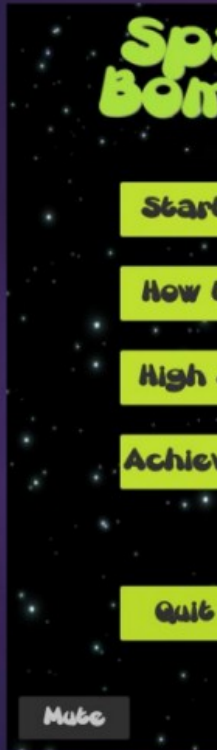
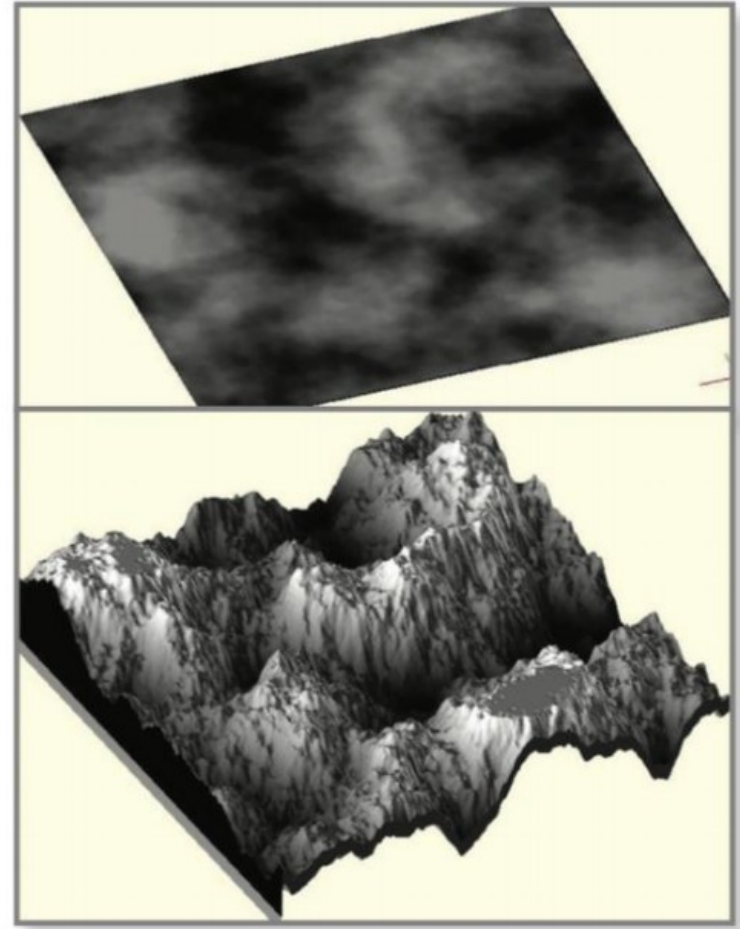
# Project proposals

- Simulation



## Topic

Convert BMP file to STL file



The game  
Selling minerals  
Town Buildings  
Buttons, Mouse Support



# Conditions to pass the classes (3)

- **Grading:**
  - Maximal number of points: **100**
    - laboratories:  **$10 \cdot 6 = 60$**
    - project:  **$2 \cdot 10 + 20 = 40$**
- **To pass the subject (% of the total number of points):**
  - **>50% - 3** (50,5 pkt. – 60,0 pkt.)
  - **>60% - 3,5** (60,5 pkt. – 70,0 pkt.)
  - **>70% - 4** (70,0 pkt. – 80,0 pkt.)
  - **>80% - 4,5** (80,5 pkt. – 90,0 pkt.)
  - **>90% - 5** (90,5 pkt. – 100,0 pkt.)
- **Warning! To pass the subject you have to deliver the project (>50% points)**

# Literature

---

- English:

1. Joseph Albahari, Ben Albahari, C# 6.0 in a Nutshell, 2016.
2. Ian Griffiths, Programming C# 5.0, O'Reilly Media, 2012.

- Polish:

1. Joseph Albahari, Ben Albahari, C# 6.0 w pigułce, Helion 2016
2. Ian Griffiths - "C# 5.0. Programowanie", Helion, 2013.
3. Andrew Troelsen - "Język C# 2010 i platforma .NET 4", PWN, 2011.
4. Jon Skeet - "C# od podszewki", Helion, 2012.
5. Jesse Liberty - "Programowanie C#", Helion 2012

# Programme

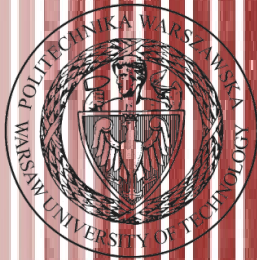
---

1. Introduction to the C# programming language and Visual Studio software.
2. Principles of C# programming language, basic information on the .NET platform. Windows Forms.
3. Classes, inheritance, virtual methods.
4. Interfaces, instruction foreach, yield iterators.
5. Standard library classes (collections, streams and files).
6. Delegations, lambda expressions.
7. Events, exceptions.
8. LINQ technology.

# Programme

---

1. Introduction to the C# programming language and Visual Studio software. Principles of C# programming language.
2. Windows Forms.
3. Windows Presentation Foundation (WPF).
4. Web Forms: ASP.NET.
5. Databases: AOD.NET.
6. PROJECT I
7. Classes, inheritance, virtual methods.
8. Delegations, lambda expressions.
9. Events, exceptions.
10. PROJECT II
11. LINQ technology.
12. Multithreading.
13. To be decided.
14. PROJECT II



# Introduction to the C# language and Visual Studio software

# C#

- C# (pronounced "C sharp") is a programming language that is designed for building a variety of applications that run on the .NET Framework.





# .NET Framework

.NET Framework (pronounced **dot net**) is a software framework developed by Microsoft.



.NET Framework includes 2 parts:

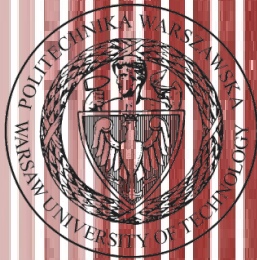
- a **large class library** known as **Framework Class Library** (FCL) and provides language interoperability (each language can use code written in other languages) **across several programming languages (C#, C++, F#, Visual Basic, and a few dozen others)**.
- programs written for .NET Framework execute in a software environment known as **Common Language Runtime (CLR)**, an **application virtual machine** that provides services such as security, memory management, and exception handling.

# Why C#?

---



- Simple and easy to learn
- Curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java → easy for people previously programming in any of those languages
- C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java.
- C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code.
- Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

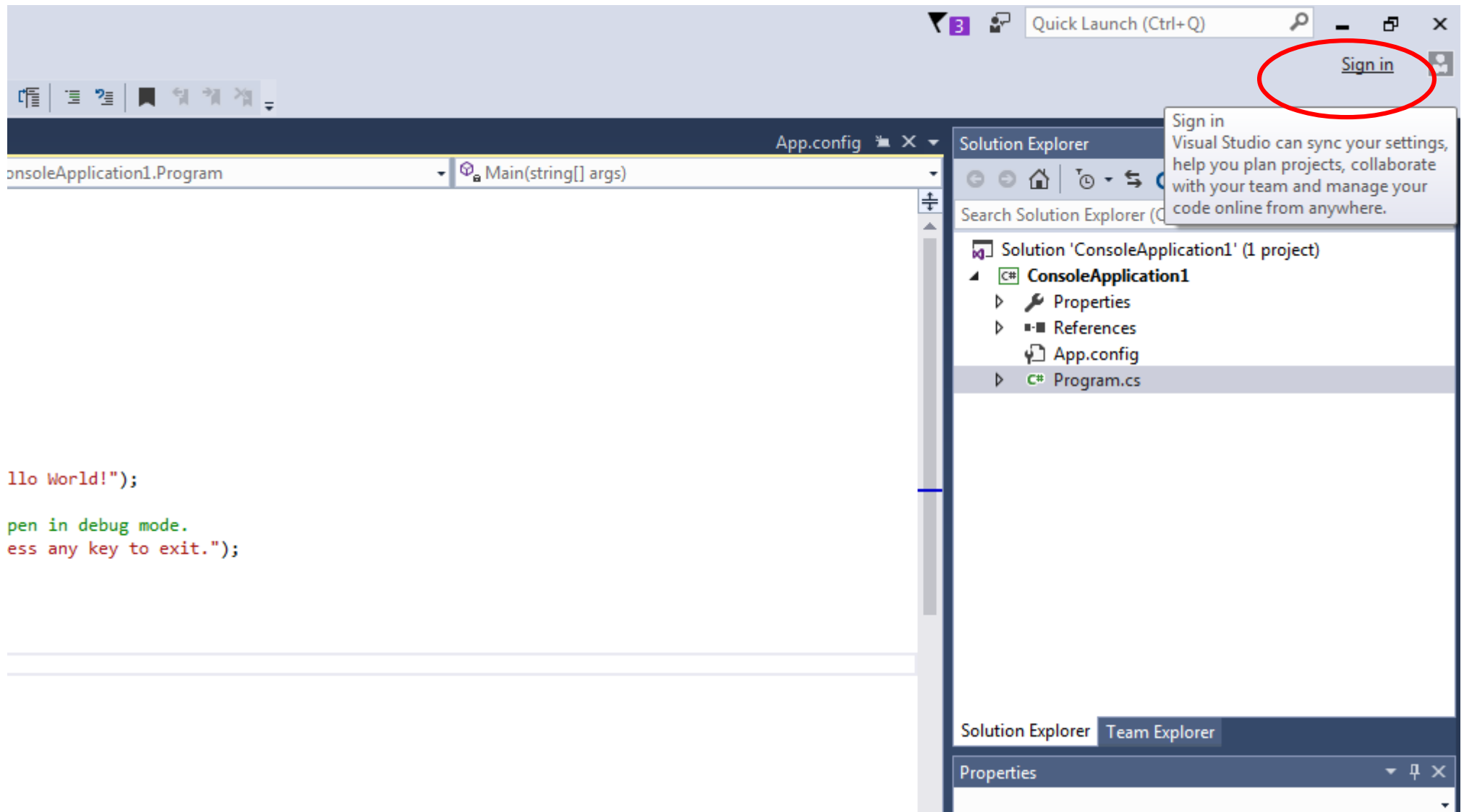


# Hands on!

## First console application

- Open Visual Studio
- File → New → Project
- Console Application  
(.NET Framework)

# Sign in into Visual Studio....



# Sign in into Visual Studio....



Wpisz adres e-mail lub numer telefonu konta, za pomocą którego chcesz się zalogować.

Kontynuuj

To samo konto, co poczta politechniczna:

<https://portal.office.com>

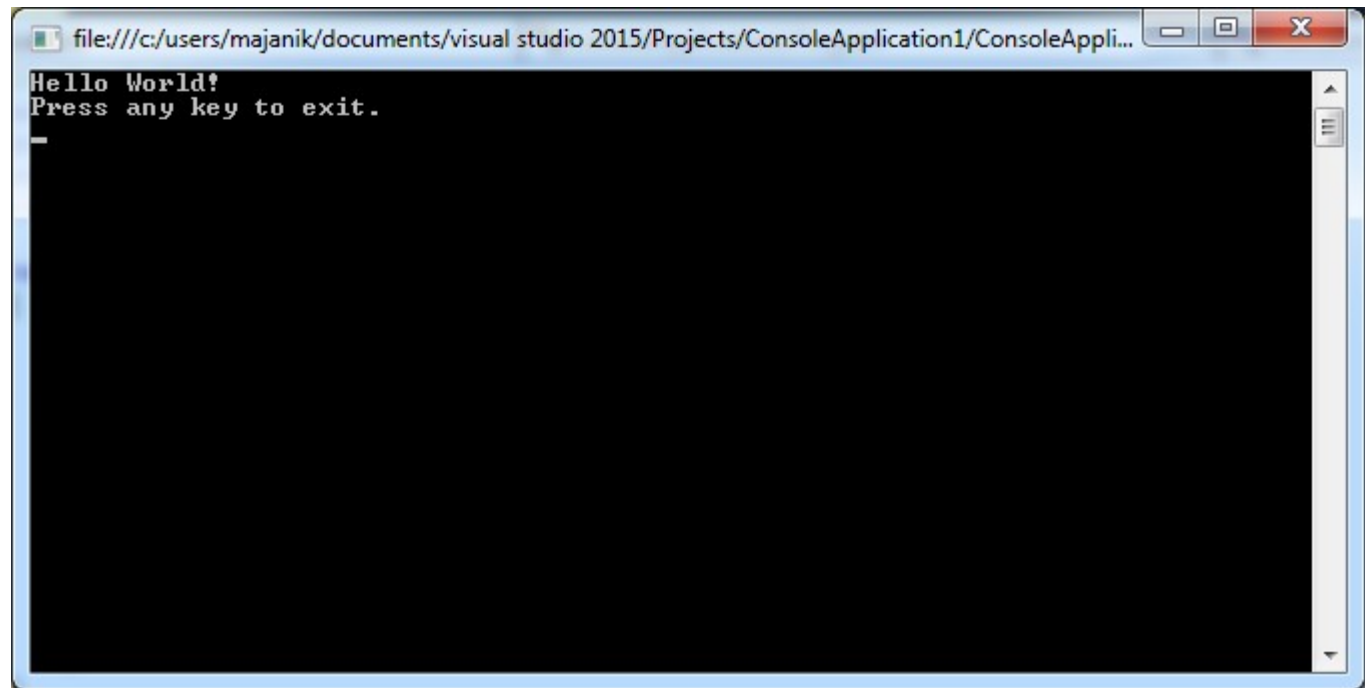
[numer]@pw.edu.pl  
111111111@pw.edu.pl

Można również stworzyć nowe konto.

# Console Application - Printing

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");

            // Keep the console window open in debug mode.
            System.Console.WriteLine("Press any key to exit.");
            System.Console.ReadKey();
        }
    }
}
```



The screenshot shows a Windows-style console window titled "file:///c:/users/majanik/documents/visual studio 2015/Projects/ConsoleApplication1/ConsoleAppli...". The window has a black background and white text. The text displayed is "Hello World!" on the first line and "Press any key to exit." on the second line. A small white cursor is visible on the third line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

# Console Application - Variables

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");

            int a = 10;
            string b = "label";
            System.Console.WriteLine("Variables: {0} {1}", a, b);

            var c = "label2";
            // var d; // NOT POSSIBLE
            // Keep the console window open in debug mode.
            System.Console.WriteLine("Press any key to exit.");
            System.Console.ReadKey();
        }
    }
}
```

# Console Application - Task

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");

            int a = 10;
            string b = "label";
            System.Console.WriteLine("Variables: {0} {1}", a, b);

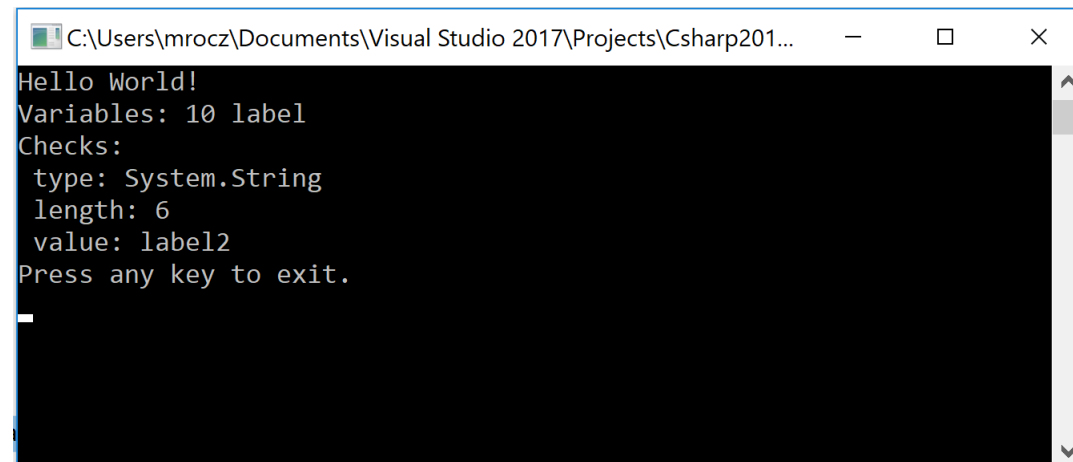
            var c = "label2";

            // TASK
            // Print: type of c, length of c and value of c

            // Keep the console window open in debug mode.
            System.Console.WriteLine("Press any key to exit.");
            System.Console.ReadKey();
        }
    }
}
```

Type "c." and wait for the list of possible methods and properties appear.

Browse through them and try to find the ones requested.

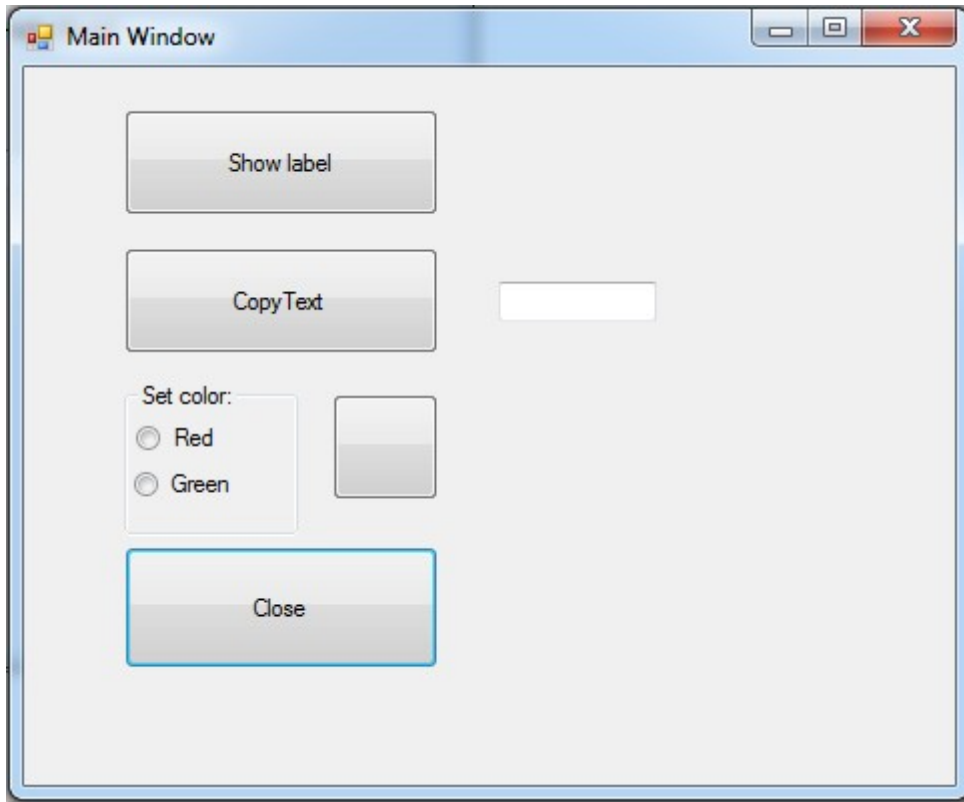


```
C:\Users\mrocz\Documents\Visual Studio 2017\Projects\Csharp201...
Hello World!
Variables: 10 label
Checks:
  type: System.String
  length: 6
  value: label2
Press any key to exit.
_
```



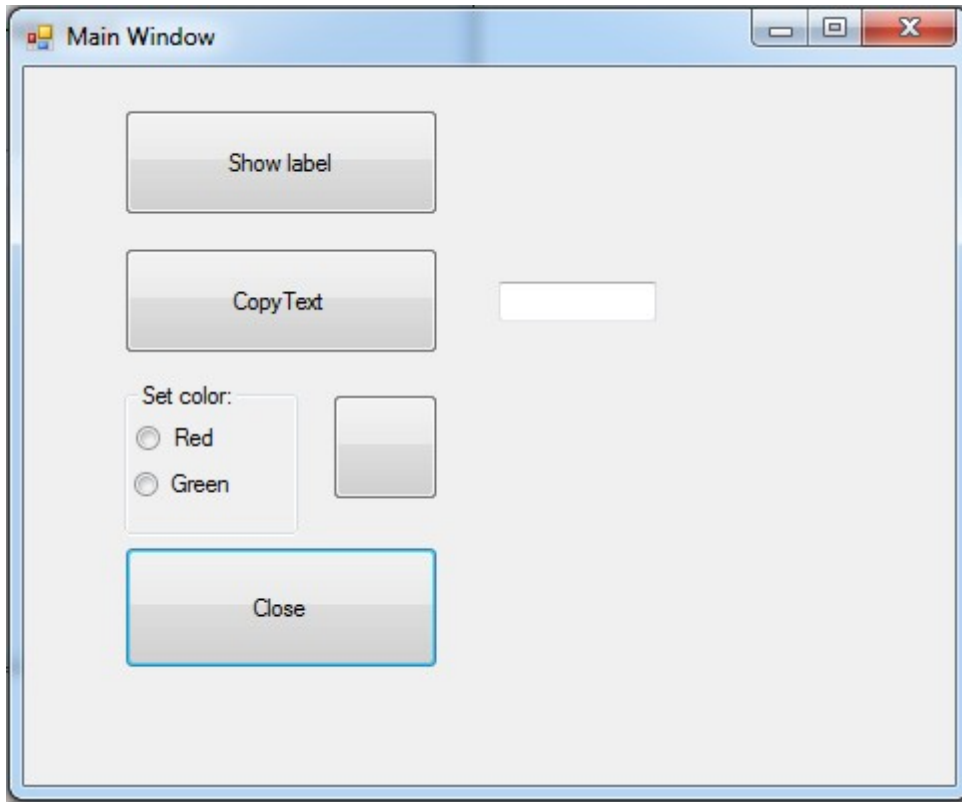
# Planned application

Initial window:

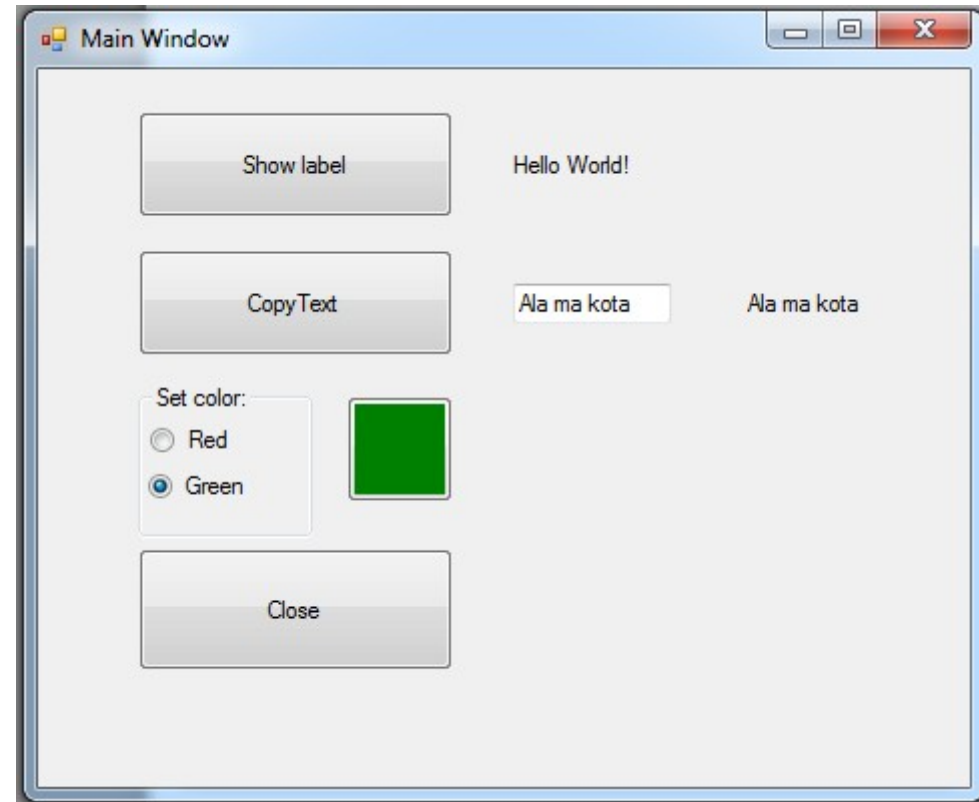


# Planned application

Initial window:



Used functionalities:

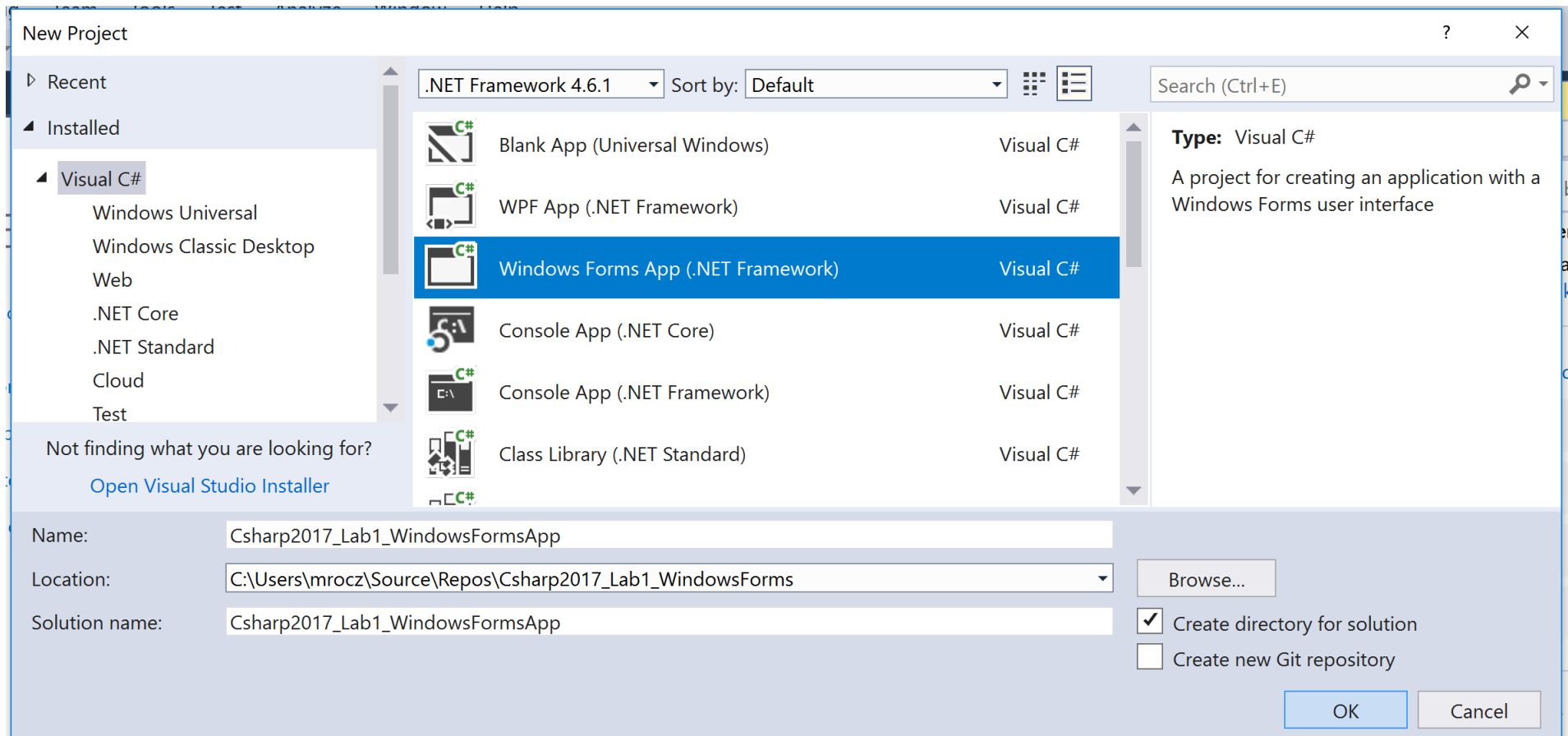
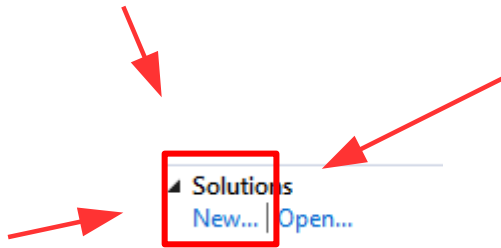


# Create new project

New...

Project

**Windows Forms Application**



# Create new project and synchronize it with repository

The screenshot displays the Microsoft Visual Studio IDE with a new Windows Forms application named 'WindowsFormsApplication1'. The main window, 'Form1.cs [Design]', is in design mode and contains a 'Main Window' with the following elements:

- A 'Show label' button at the top left.
- A 'Hello World!' label to the right of the 'Show label' button.
- A 'CopyText' button below the 'Show label' button.
- A text input field to the right of the 'CopyText' button.
- A 'Set color:' section with two radio buttons: 'Red' (selected) and 'Green'.
- A color selection box to the right of the radio buttons.
- A 'Close' button at the bottom.

The Solution Explorer on the right shows the project structure:

- Solution 'WindowsFormsApplication1' (1 project)
  - WindowsFormsApplication1
    - Properties
    - References
    - App.config
    - Form1.cs
      - Form1.Designer.cs
      - Form1.resx
      - FormMainWindow
      - Program.cs

Solution Explorer

Team Explorer

Properties

buttonHello System.Windows.Forms.Button

Margin	3; 3; 3; 3
MaximumSize	0; 0
MinimumSize	0; 0
Modifiers	Private
Padding	0; 0; 0; 0
RightToLeft	No
Size	157; 53
TabIndex	1
TabStop	True
Tag	
Text	Show label

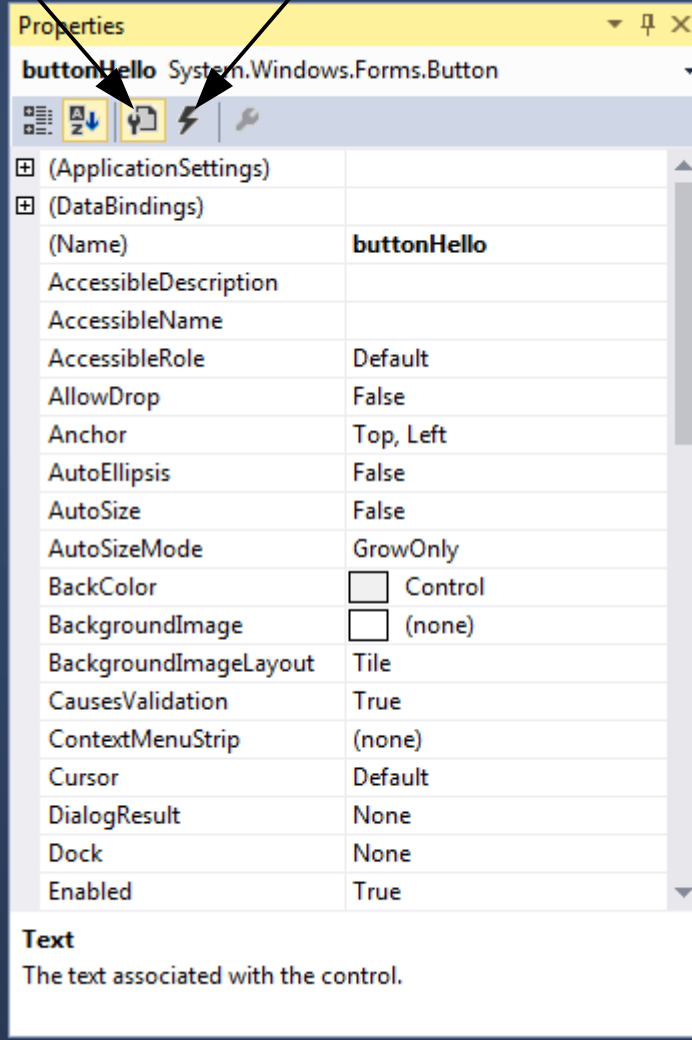
Text

The text associated with the control.

# Properties and Events

Properties

Events



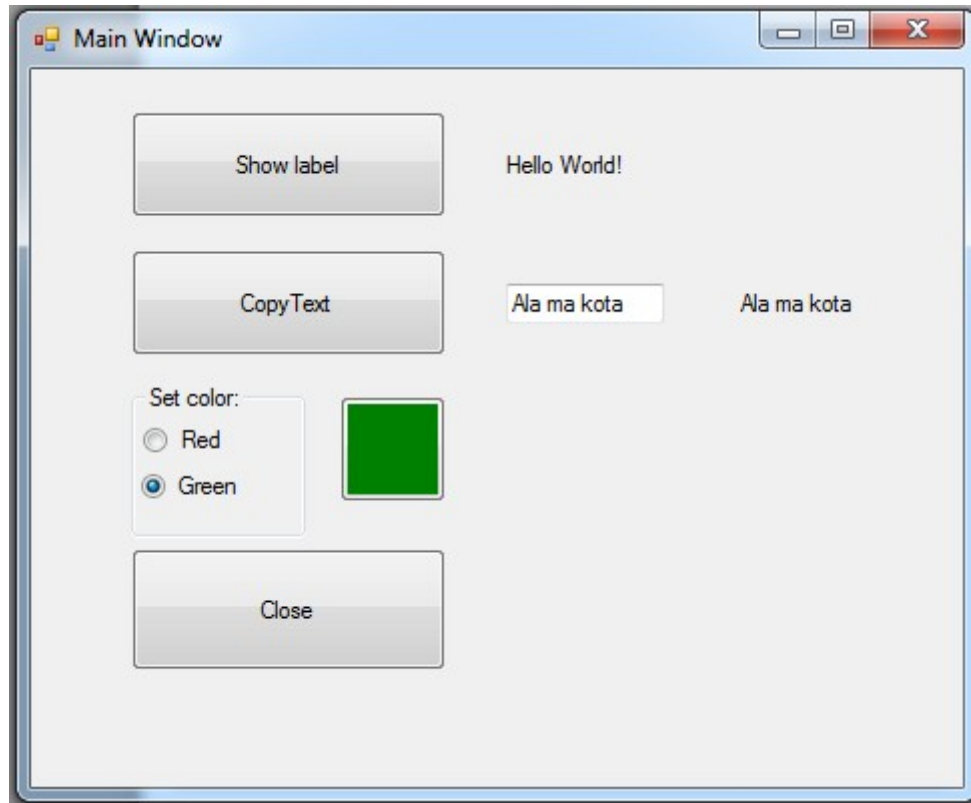
The screenshot shows the Visual Studio Properties window for a control named `buttonHello` of type `System.Windows.Forms.Button`. The window is divided into two main sections: Properties and Events. The Properties section is currently expanded to show a list of properties and their values. The Events section is collapsed. The Properties section includes a toolbar with icons for sorting, zooming, and refreshing. The Properties list includes:

(ApplicationSettings)	
(DataBindings)	
(Name)	<b>buttonHello</b>
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoEllipsis	False
AutoSize	False
AutoSizeMode	GrowOnly
BackColor	<input type="color"/> Control
BackgroundImage	<input type="image"/> (none)
BackgroundImageLayout	Tile
CausesValidation	True
ContextMenuStrip	(none)
Cursor	Default
DialogResult	None
Dock	None
Enabled	True

**Text**  
The text associated with the control.

# Build your application

## Used functionalities:



Remember to:

- give readable names to all controls
- commit changes after each part

## **Names:**

- Always change default names!
- Each team can have its own naming convention.
- Common thing: names are readable!

## **This classes:**

- Always keep the control name  
+ readable part.

e.g. formMainWindow  
labelHelloWorld

# Additional links

---

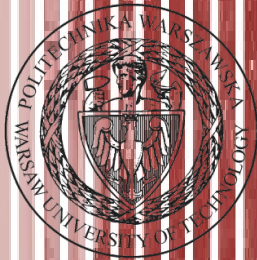
Visual Studio Team Services Agile youtube tutorials:

<https://www.youtube.com/watch?v=Vj7DmdO4-Fg>

<https://www.youtube.com/watch?v=fTGv3BAgCiM>

GIT:

<https://docs.microsoft.com/en-us/vsts/git/share-your-code-in-git-vs-2017>



**THE END**