# Advanced Programming C#

## Lecture 1

dr inż. Małgorzata Janik

majanik@if.pw.edu.pl

*Winter Semester 2017/2018*

# Organizational issues

- **Lecture + laboratories + project:**
  - dr inż. Małgorzata Janik
    Zakład Fizyki Jądrowej
    pok. 117D, Gmach Fizyki

    majanik@if.pw.edu.pl
- **Time:**
  - Monday, 12:00 – 14:00
- **Webpage:**
  - www.if.pw.edu.pl/~majanik/wiki
- **Office hours,** 117D GF:
  - Monday, 10:00 – 11:00
  - 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 6th, 10th  and 14th 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 mailboxs, 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 pk**t)

# Conditions to pass the classes (2)

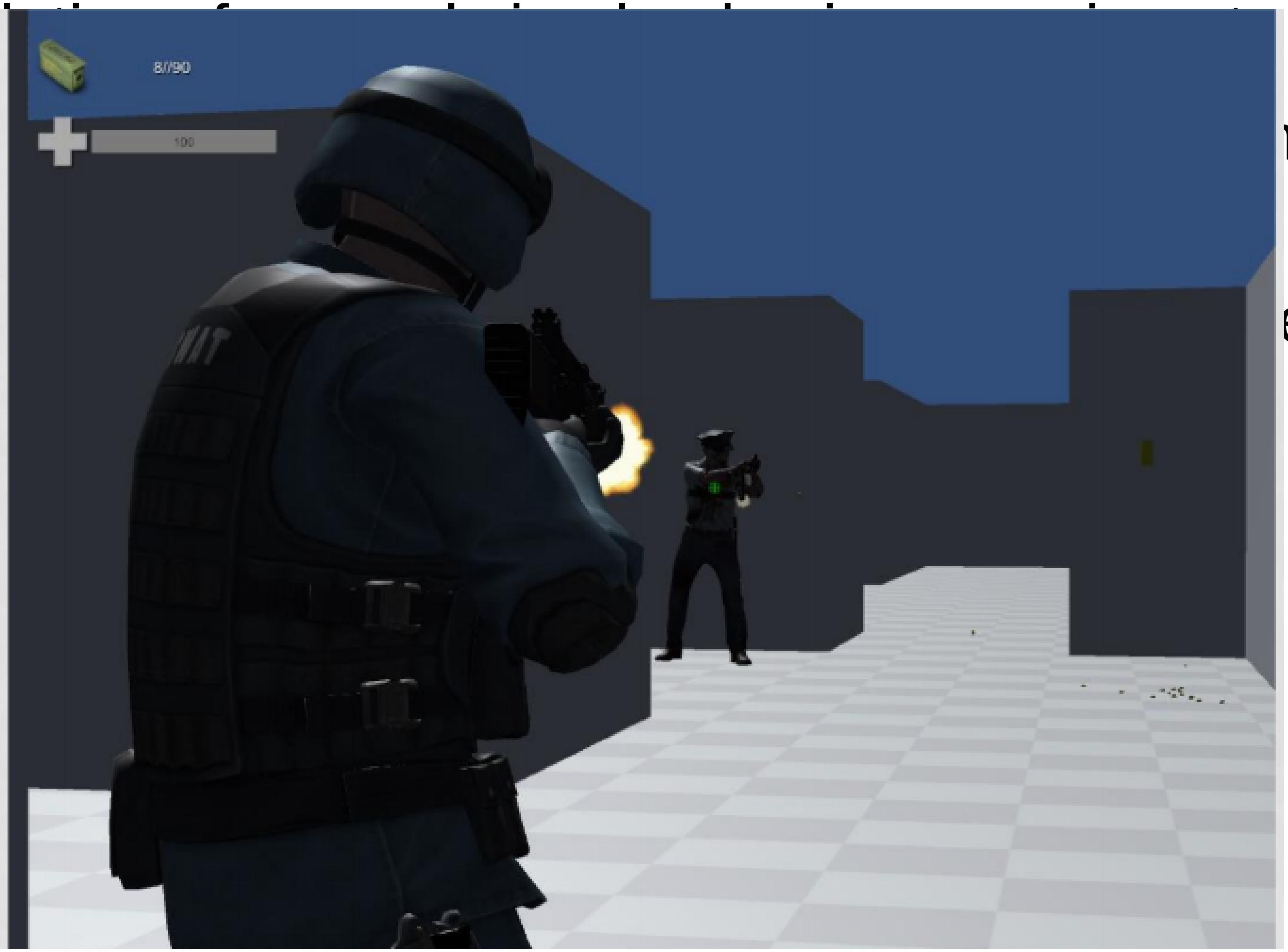
- **Project:**
  - grading: **0-40 pkt** for the project
  - During the semester there will be **2 intermediate stages,** when the current status of the project should be presented
  - Each intermediate stage: **0-10 pkt**
  - Final project (should be shown in the last class):  **0-20 pkt**
  - To pass 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

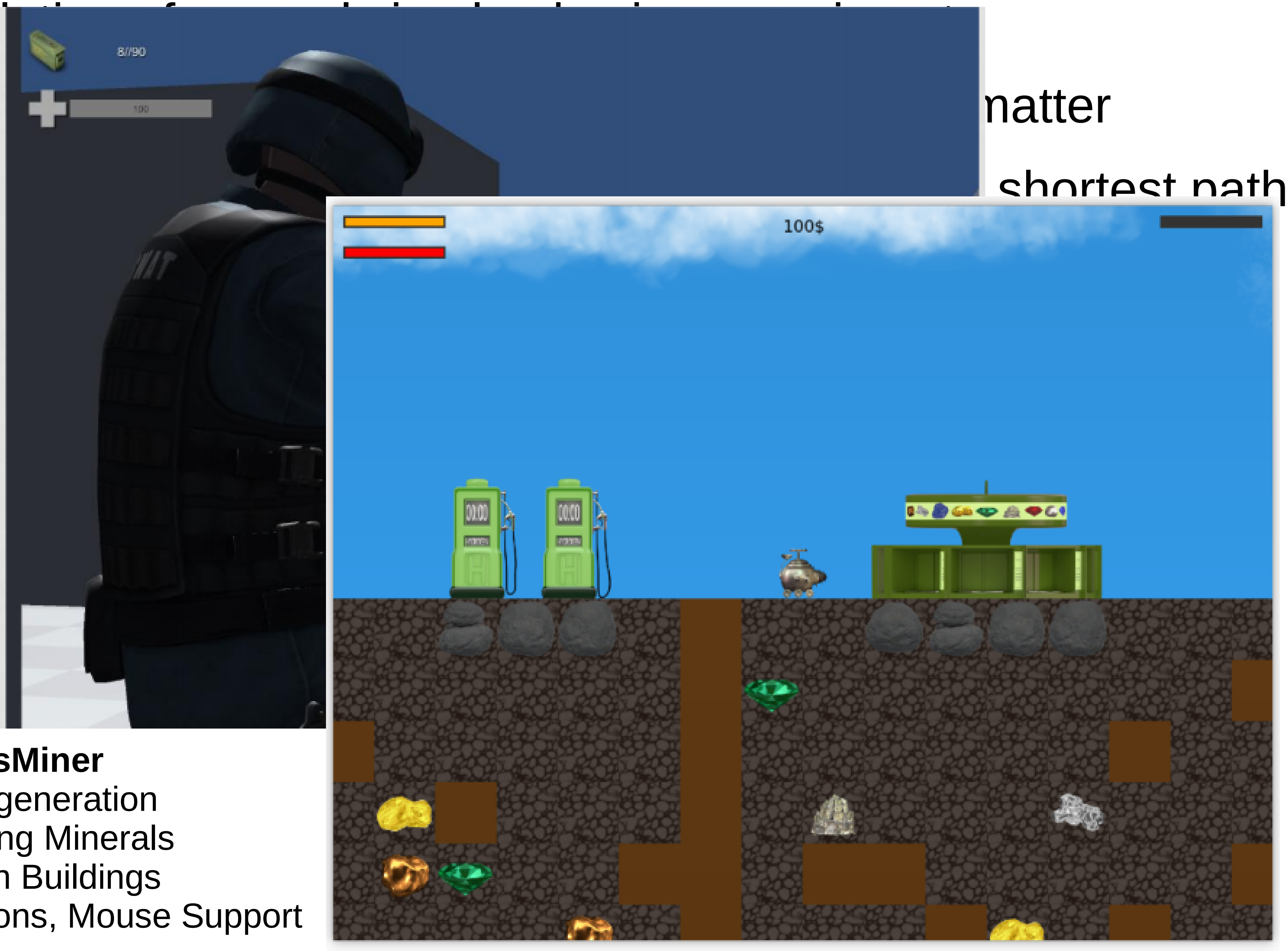
- Simu
- Simu matter
- Main shortest path betw echnology Main
- Netw
- Simp
- Simp



**Shooter**
Movement
Shooting
Death
Opponents AI

# Project proposals

- Simu
- Simu matter
- Main shortest path betw Main
- Netw
- Simp
- Simp



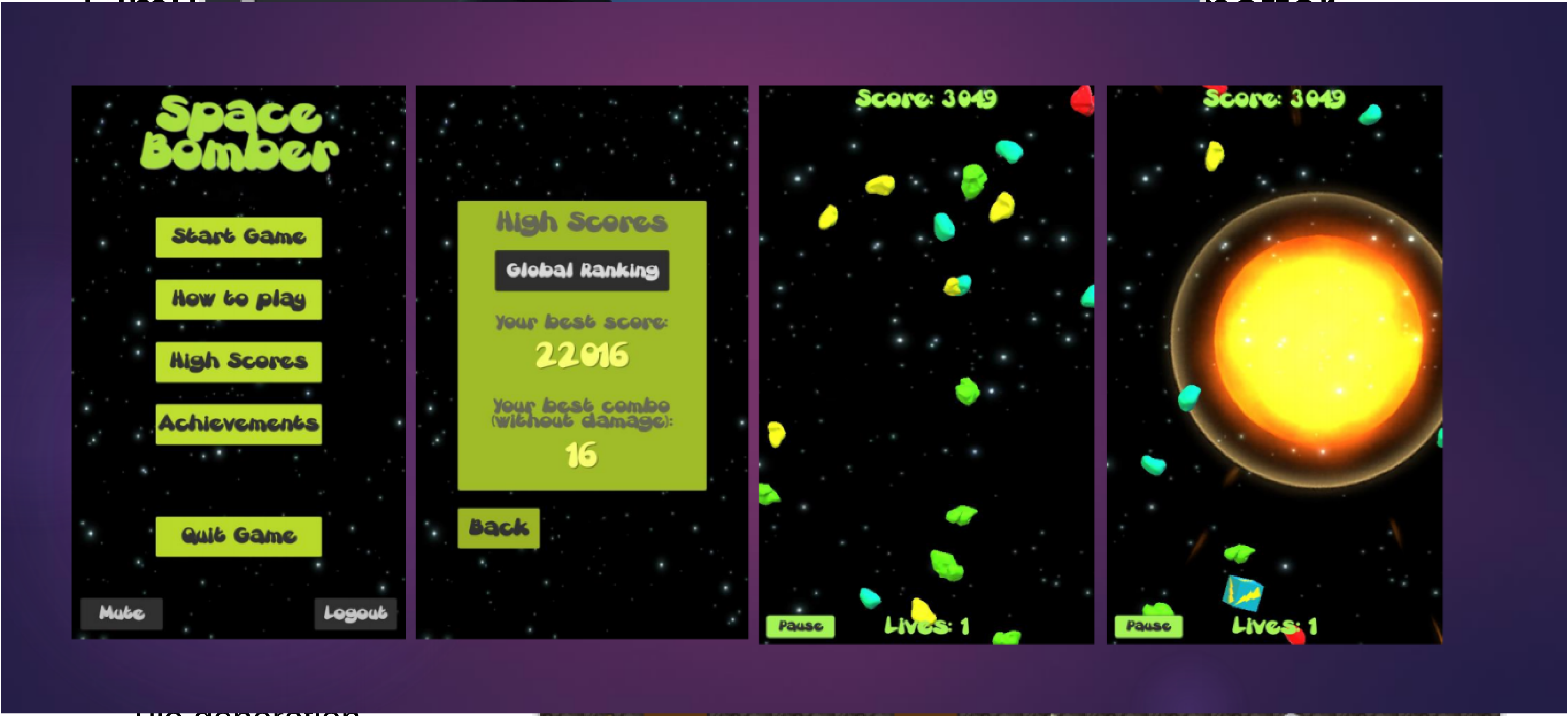
**MarsMiner**
Tile generation
Selling Minerals
Town Buildings
Buttons, Mouse Support

# Project proposals

- Simu

Simu

natter



Tile generation
Selling Minerals
Town Buildings
Buttons, Mouse Support

# Conditions to pass the classes (3)

- **Grading:**
  - Maximal number of points: **100**
    - laboratories: **10*6 = 60**
    - project: **2*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. „Factory” 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

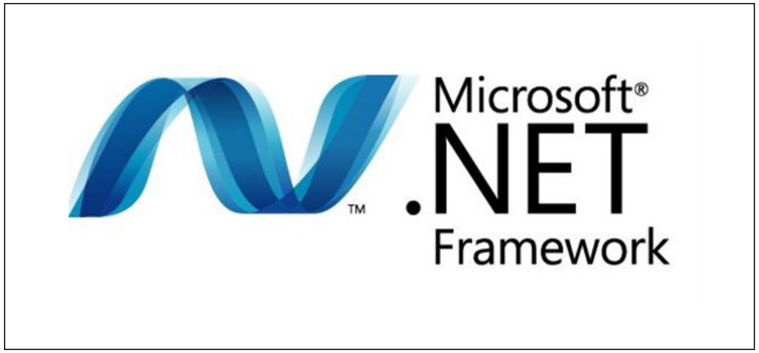
# Introduction to the C# language and Visual Studio software

# 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.

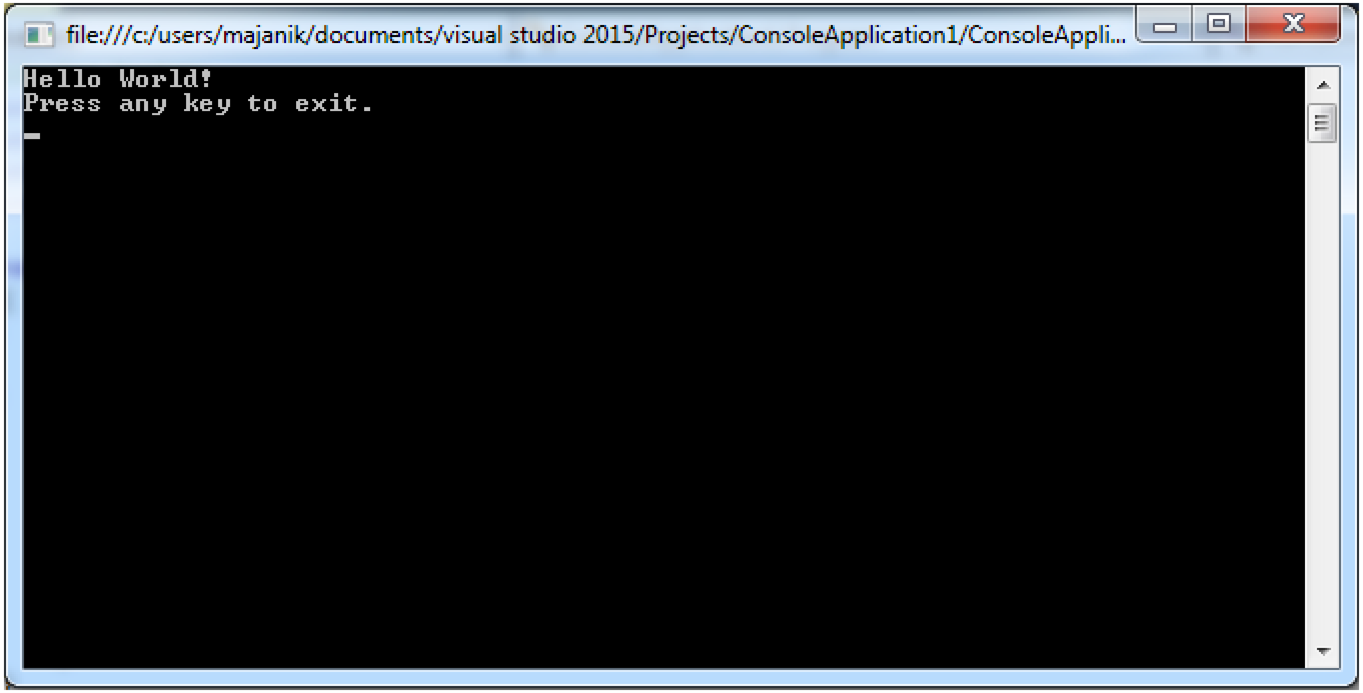
# Hands on!

# First console application

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

# 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();
        }
    }
}
```



file:///c:/users/majanik/documents/visual studio 2015/Projects/ConsoleApplication1/ConsoleAppli...

```
Hello World!
Press any key to exit.
_
```

# Console Application - Variables

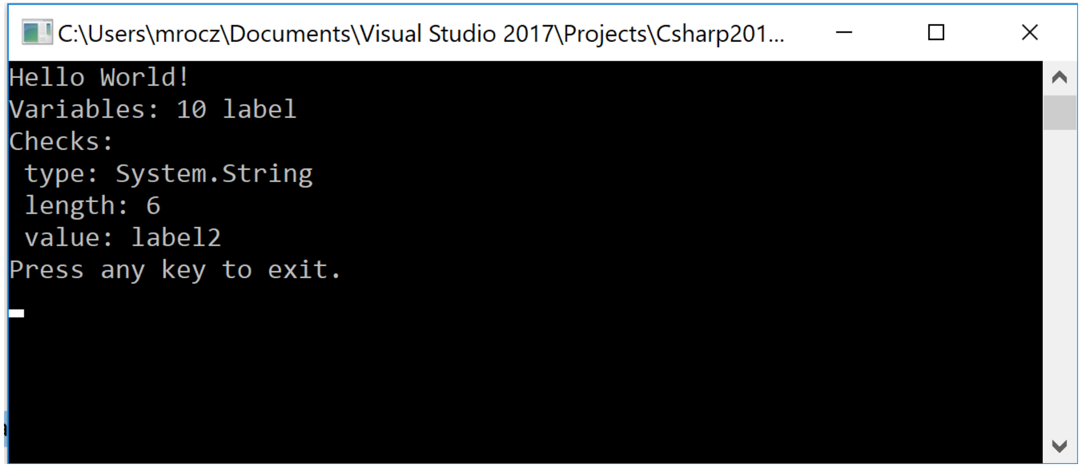
```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");

            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();
        }
    }
}
```

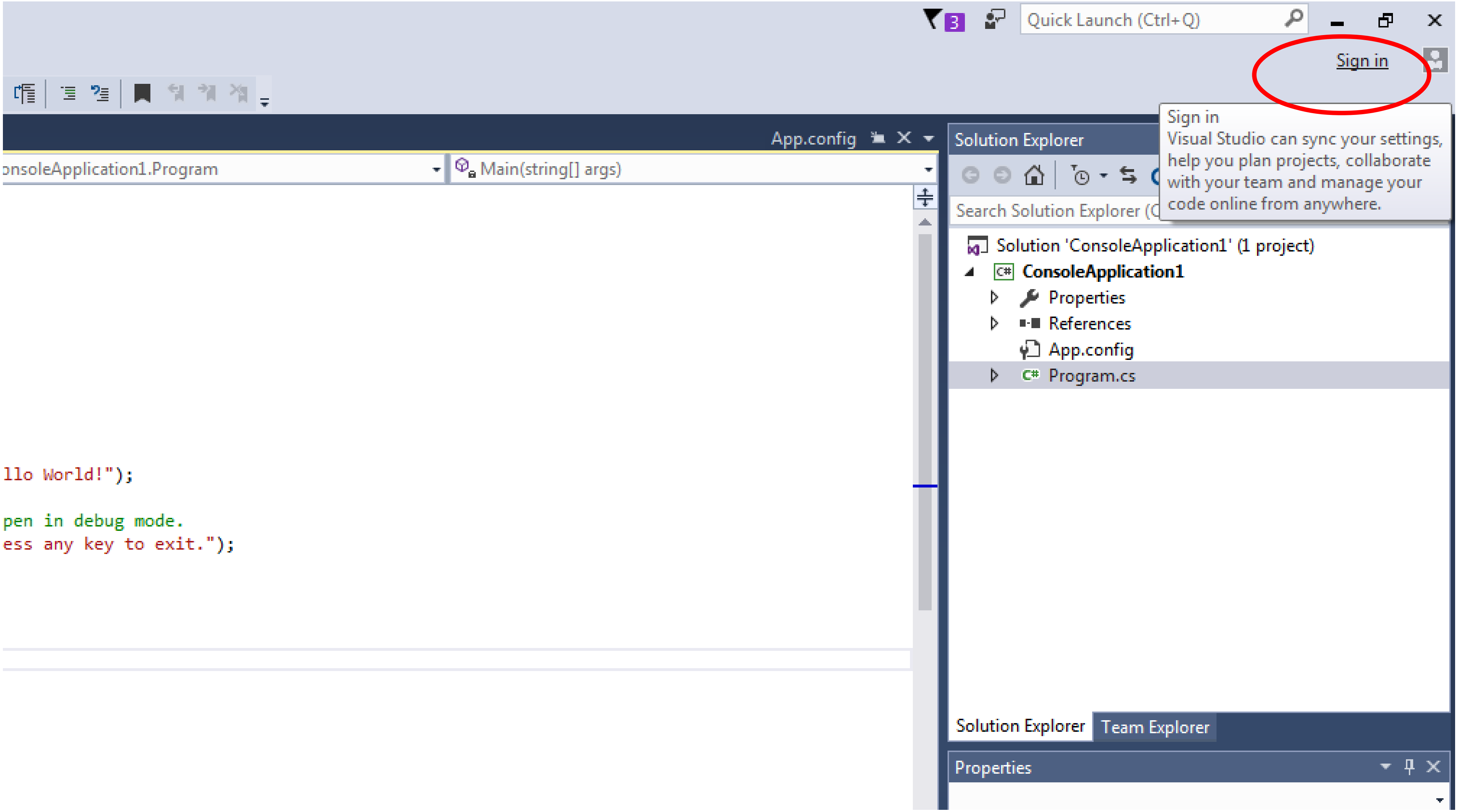
Type “c.” and wait for the list of possible methods and properties appear.

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.
```

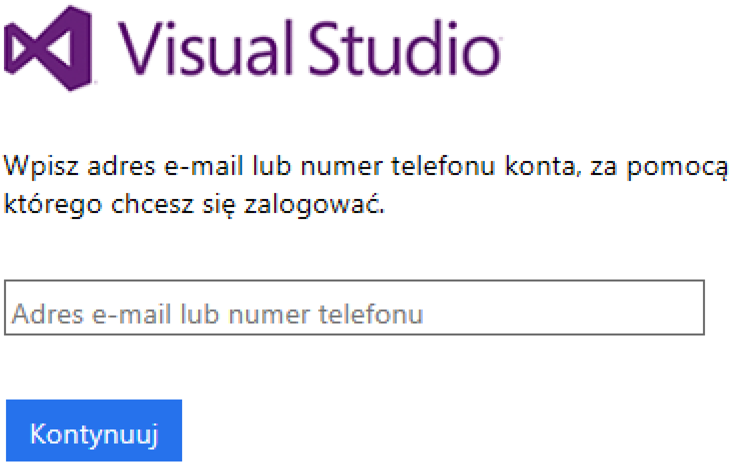
# Sign in into Visual Studio....



Sign in
Visual Studio can sync your settings, help you plan projects, collaborate with your team and manage your code online from anywhere.

```
llo World!");

pen in debug mode.
ess any key to exit.");
```

# Sign in into Visual Studio....



Visual Studio

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

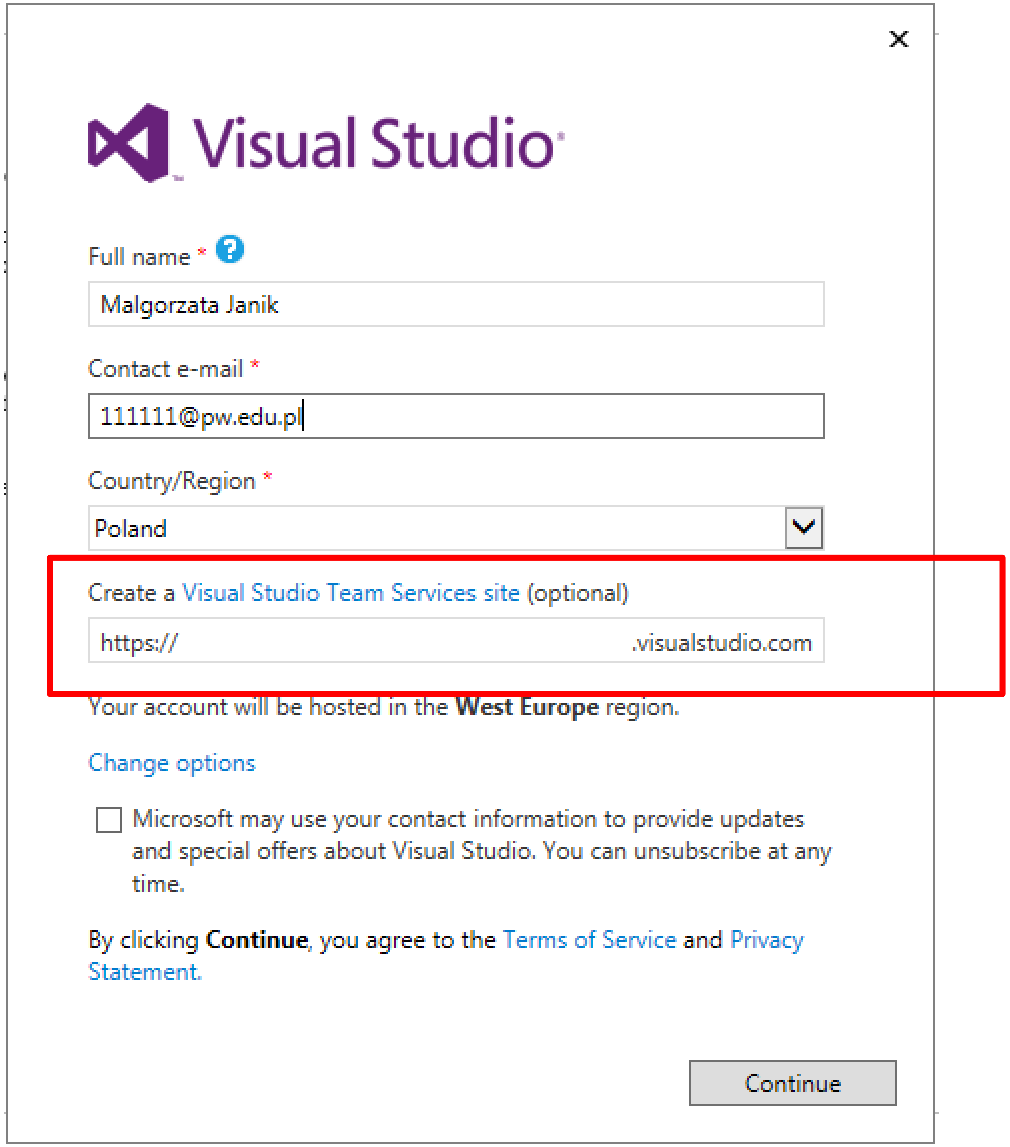
Adres e-mail lub numer telefonu

To samo konto, co poczta politechniczna:

https://portal.office.com

[numer albumu]@pw.edu.pl
111111@pw.edu.pl

# Sign in into Visual Studio….



Full name *

Malgorzata Janik

Contact e-mail *

111111@pw.edu.pl

Country/Region *

Poland

Create a Visual Studio Team Services site (optional)

https:// .visualstudio.com

Your account will be hosted in the **West Europe** region.

Change options

□ Microsoft may use your contact information to provide updates and special offers about Visual Studio. You can unsubscribe at any time.

By clicking **Continue**, you agree to the Terms of Service and Privacy Statement.

Visual Studio Team Services site:

**<u>NazwiskoCsharp2017</u>.visualstudio.com**

# Visual Studio Team Services

- Open in webbrowser
  **NazwiskoCsharp2017.visualstudio.com**

→ **New project**

- Lab1WindowsFormsProject

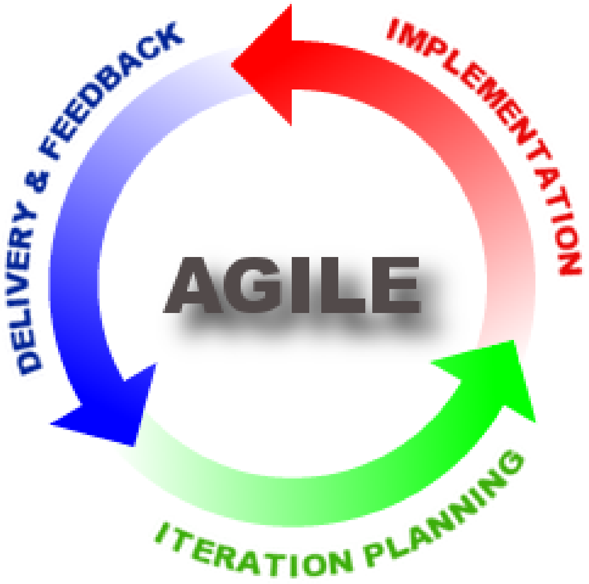
- Git

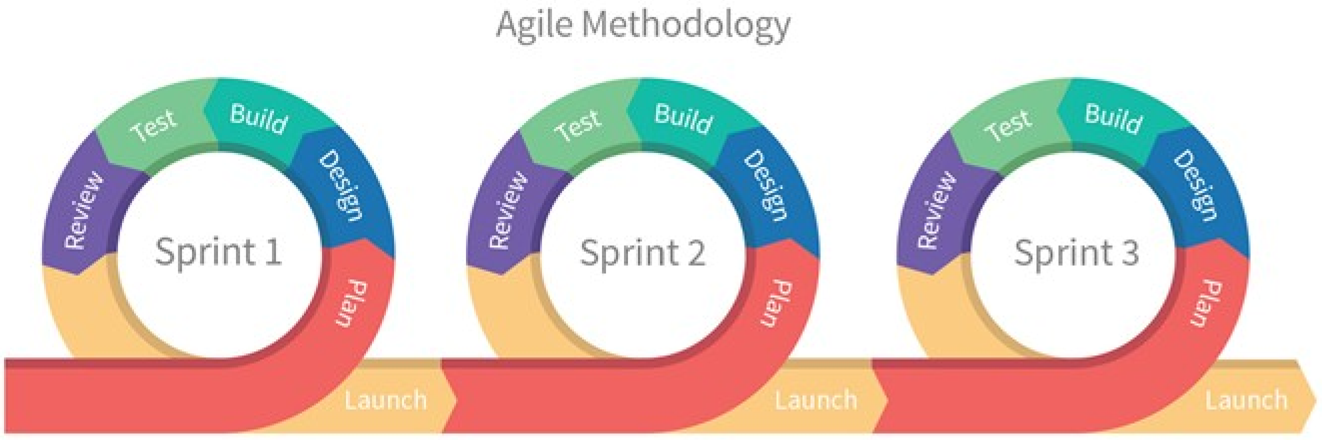
- Agile

# Agile software developement

Agile is a term used to describe a general approach to software development.

All agile methods, emphasize teamwork, frequent deliveries of working software, close customer collaboration, and the ability to respond quickly to change.





# Visual Studio Team Services: Work

**WORK tab: Board**

- It's new (New)

- I'm working on it (Active)

- I think I've finished  / fixed it (Resolved)

- Yes, it's finished / fixed, never look at it again (Closed)

Create three tasks:

(1) „Learn basic features of VS Team Services”

(2) „Create new project and synchronize it with repository”

(3) “Write demo program in Windows Forms.”

Move „Learn basic features of VS Team Services” to Active.

# Dashboards: other functionalities of VSTS

**Team Members**

→ Invite a friend

**Dashboad**

→ Add Team Room

**Rooms**

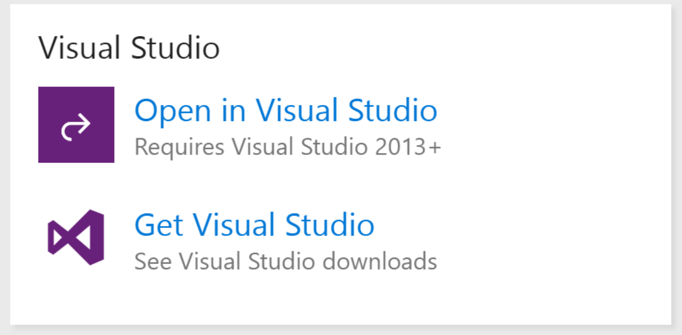
→ See rooms

→ Create events

„Learn basic features of VS Team Services” → Resolved

“Create new project and synchronize it with repository” → Active

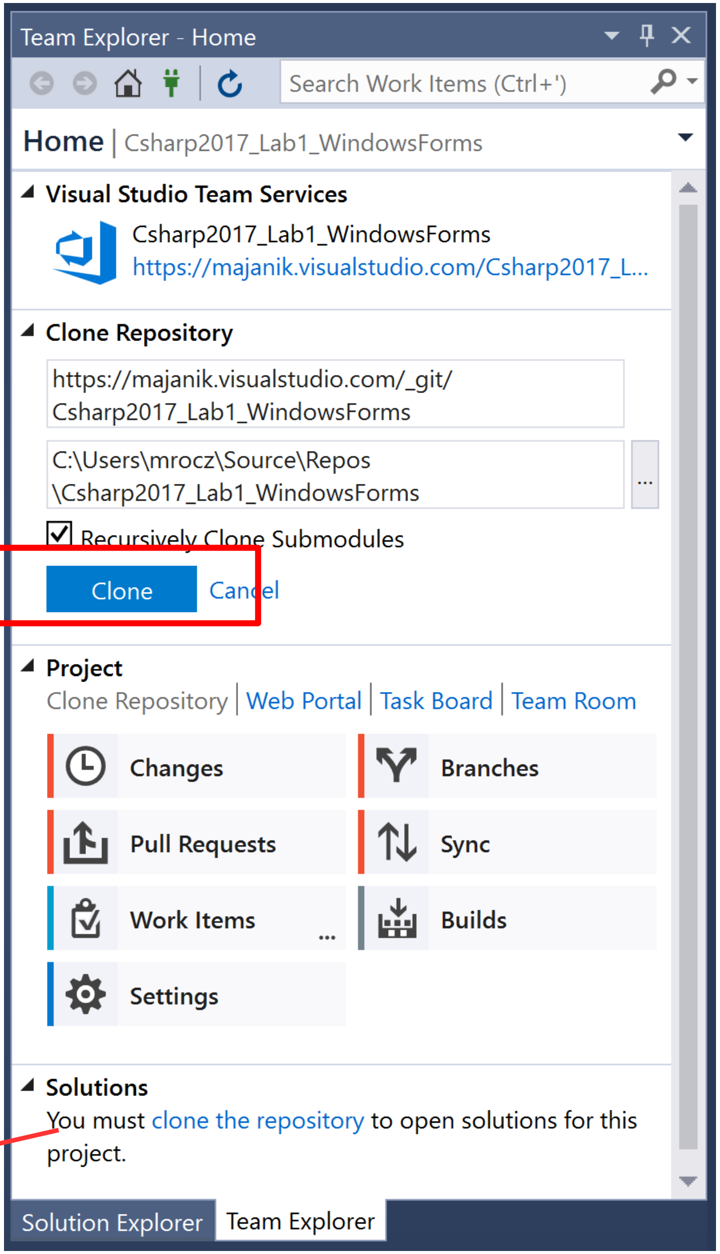
# Create new project and synchronize it with repository



Visual Studio

Open in Visual Studio
Requires Visual Studio 2013+

Get Visual Studio
See Visual Studio downloads



Team Explorer - Home

Search Work Items (Ctrl+')

Home | Csharp2017_Lab1_WindowsForms

Visual Studio Team Services

Csharp2017_Lab1_WindowsForms
https://majanik.visualstudio.com/Csharp2017_L...

Clone Repository

https://majanik.visualstudio.com/_git/Csharp2017_Lab1_WindowsForms

C:\Users\mrocz\Source\Repos\Csharp2017_Lab1_WindowsForms

☑ Recursively Clone Submodules

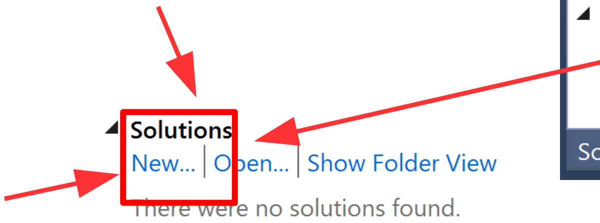
Clone Cancel

Project

Clone Repository | Web Portal | Task Board | Team Room

Solutions

You must clone the repository to open solutions for this project.

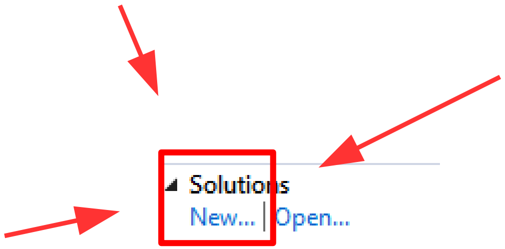


Solutions

New... | Open... | Show Folder View

There were no solutions found.

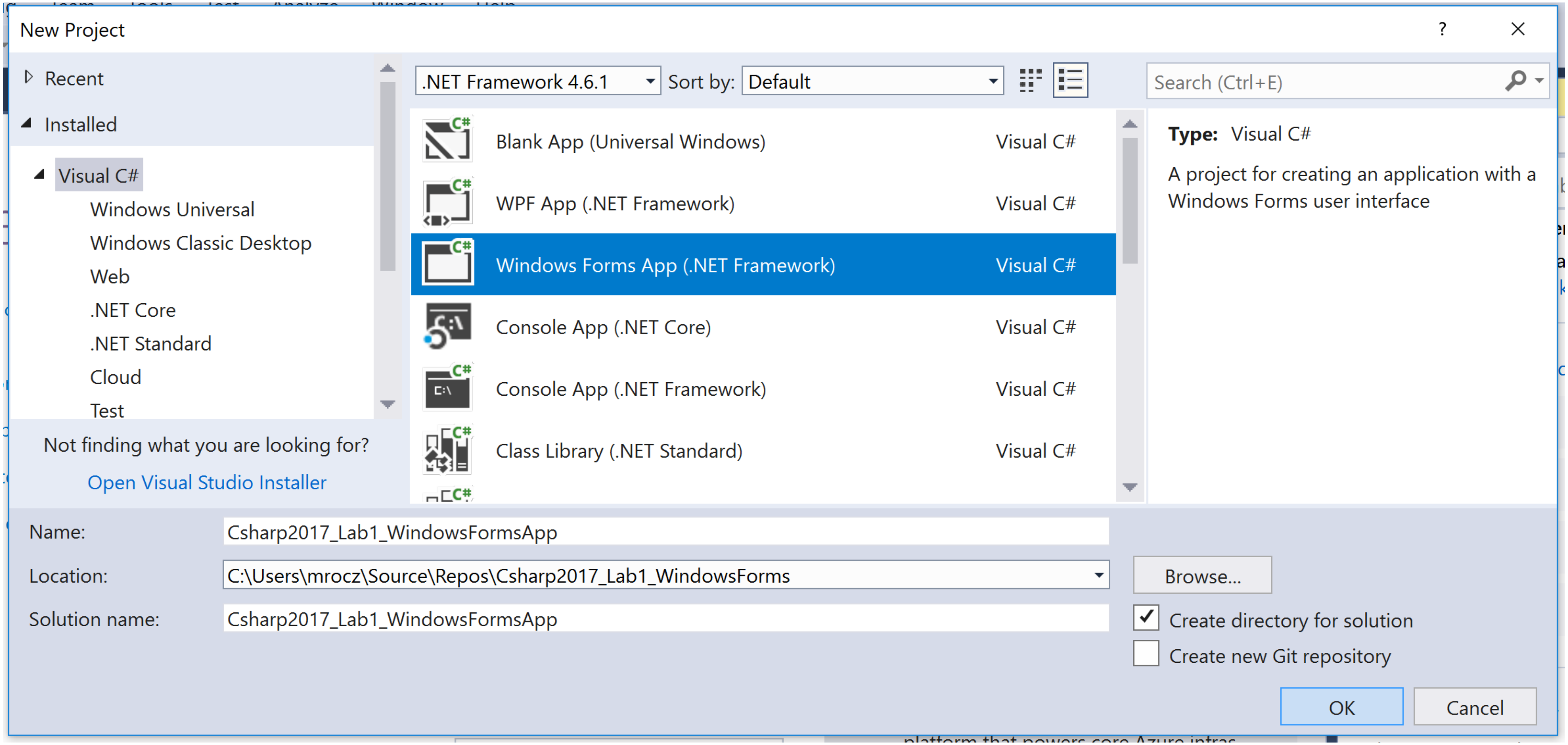
# Create new project and synchronize it with repository



New...

Project

**<u>Windows Forms Application</u>**


New Project
?
Recent
Installed
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Cloud
Test
Not finding what you are looking for?
Open Visual Studio Installer
Search (Ctrl+E)
Blank App (Universal Windows) Visual C#
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Standard) Visual C#
Browse...
OK
Cancel

.NET Framework 4.6.1 — Sort by: Default

**Type:** Visual C#

A project for creating an application with a Windows Forms user interface

Name: Csharp2017_Lab1_WindowsFormsApp

Location: C:\Users\mrocz\Source\Repos\Csharp2017_Lab1_WindowsForms

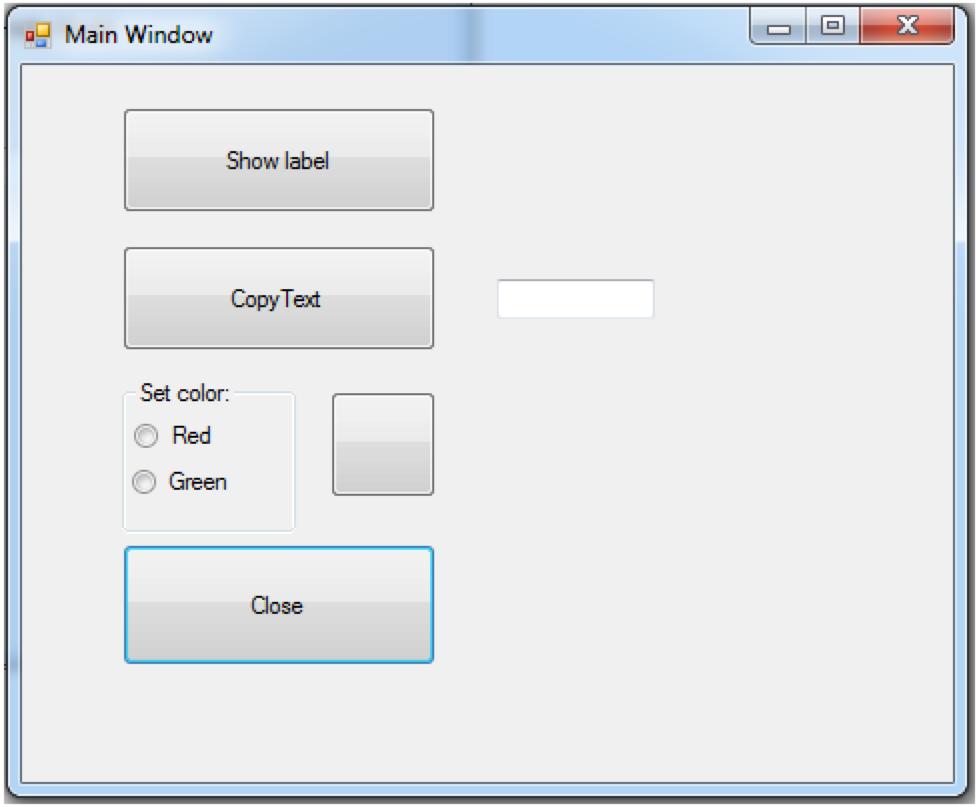
Solution name: Csharp2017_Lab1_WindowsFormsApp

☑ Create directory for solution

☐ Create new Git repository

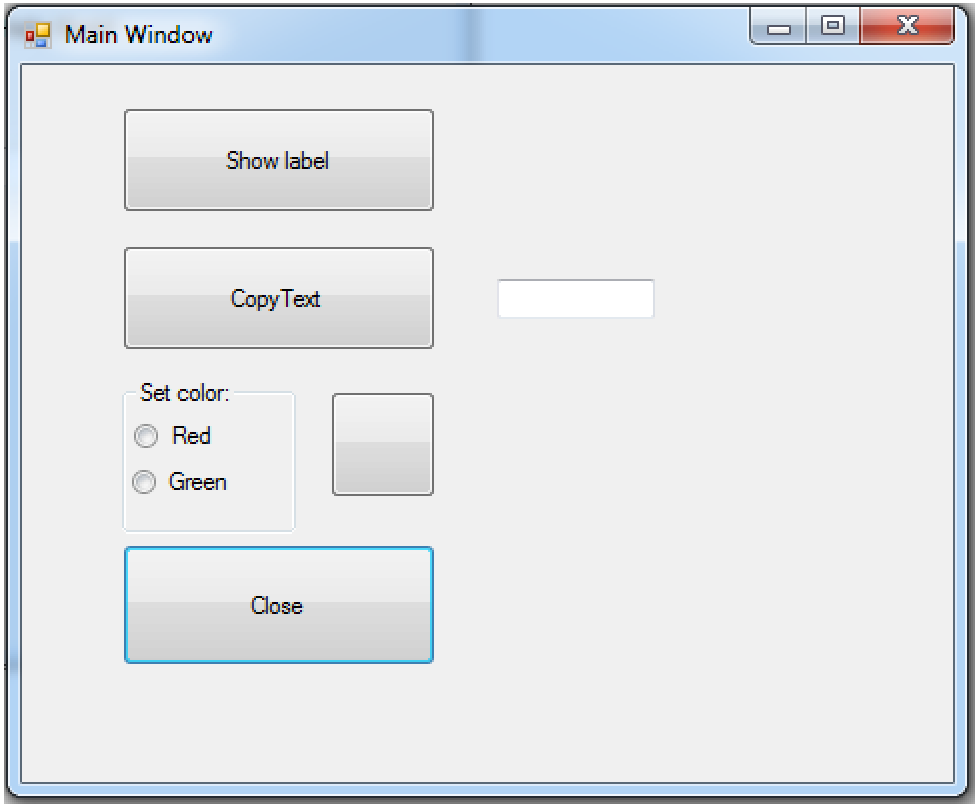
# Planned application

Initial window:

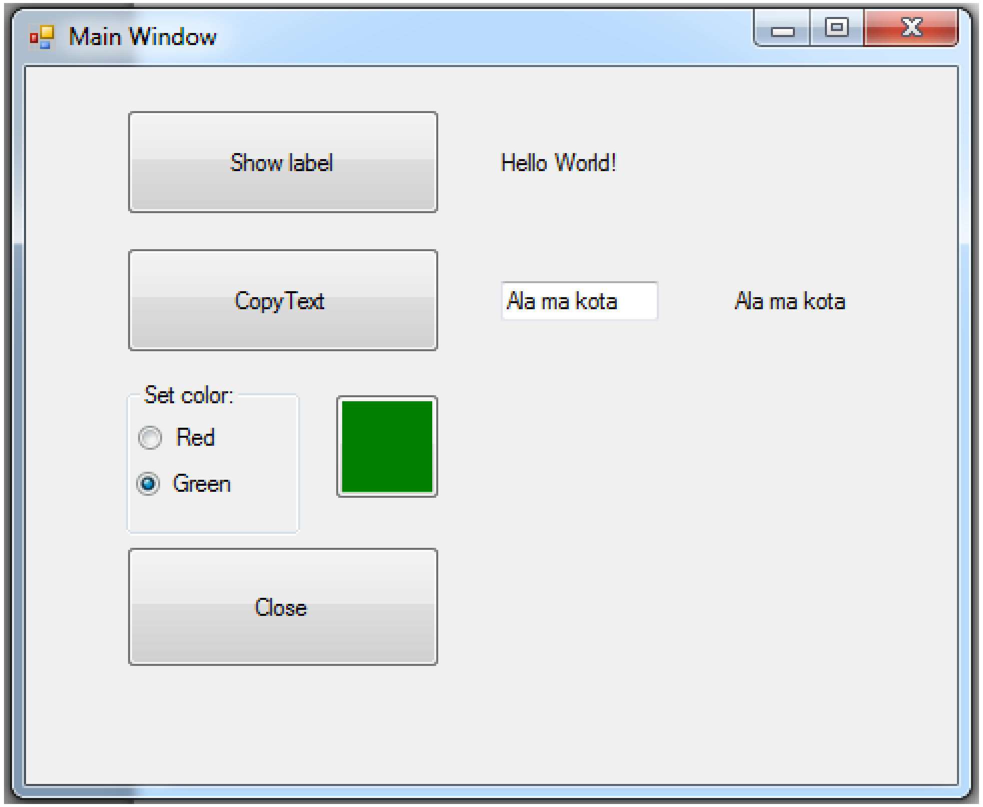


# Planned application

Initial window:



Used functionalities:



Hello World!

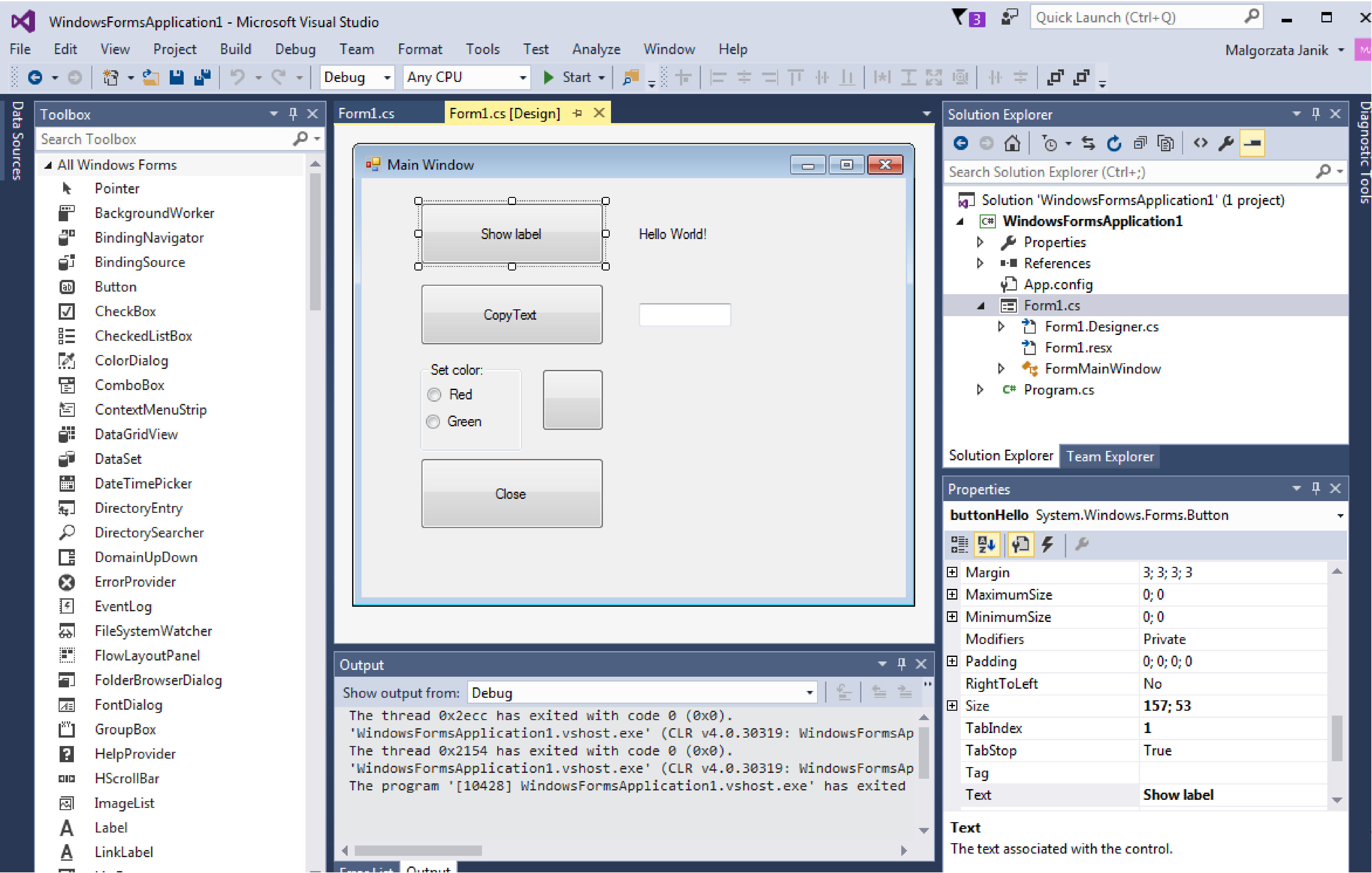
Ala ma kota

Ala ma kota

○ Red
● Green

🟩

# Create new project and synchronize it with repository



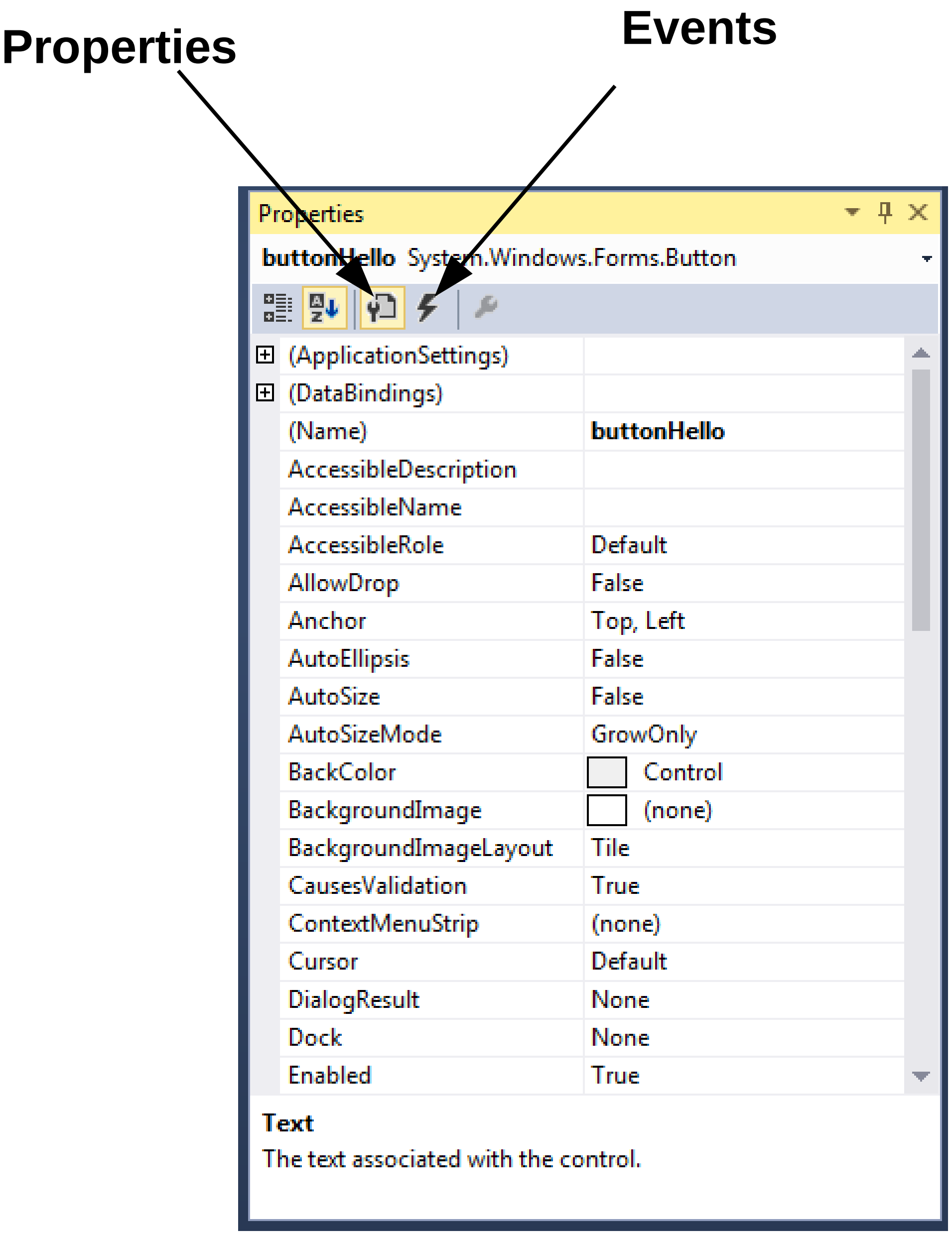
```
The thread 0x2ecc has exited with code 0 (0x0).
'WindowsFormsApplication1.vshost.exe' (CLR v4.0.30319: WindowsFormsAp
The thread 0x2154 has exited with code 0 (0x0).
'WindowsFormsApplication1.vshost.exe' (CLR v4.0.30319: WindowsFormsAp
The program '[10428] WindowsFormsApplication1.vshost.exe' has exited
```

**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

buttonHello System.Windows.Forms.Button

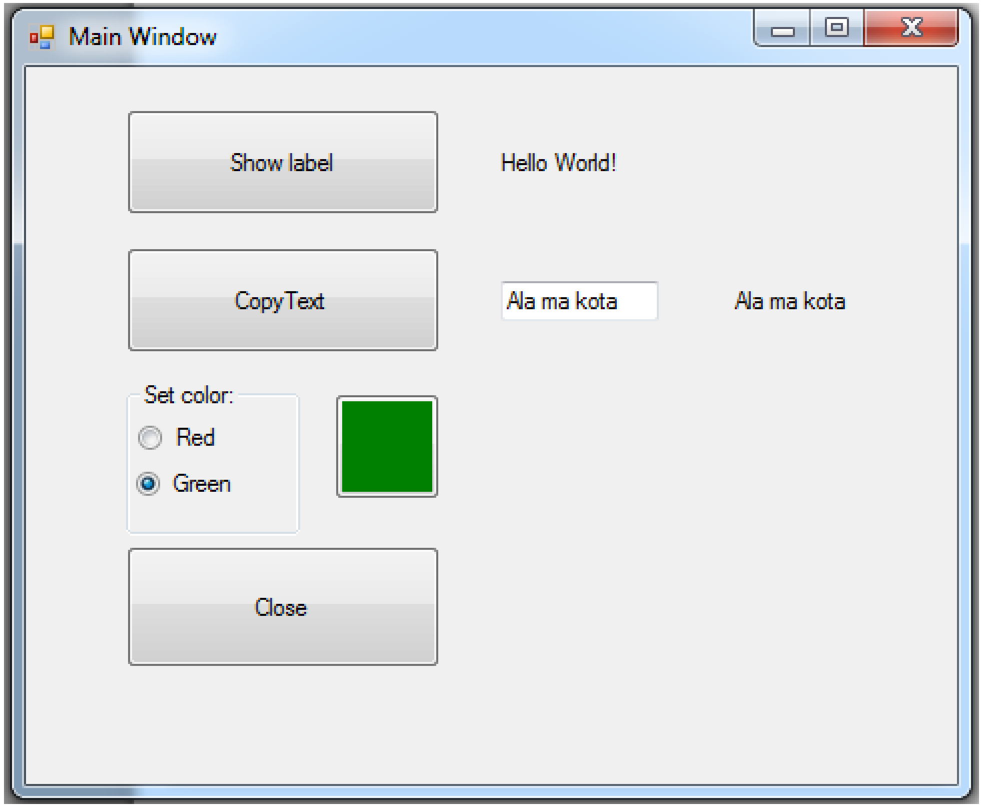
|  |  |
|---|---|
| ⊞ (ApplicationSettings) |  |
| ⊞ (DataBindings) |  |
| (Name) | **buttonHello** |
| AccessibleDescription |  |
| AccessibleName |  |
| AccessibleRole | Default |
| AllowDrop | False |
| Anchor | Top, Left |
| AutoEllipsis | False |
| AutoSize | False |
| AutoSizeMode | GrowOnly |
| BackColor | □ Control |
| BackgroundImage | □ (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