Introductie C#

(voor Unity beginners )

Unity is een populaire game engine die gebruikt wordt voor het ontwikkelen van diverse computerspellen. Om logica toe te voegen aan jouw spel maakt Unity gebruik van de programmeertaal C-Sharp, afgekort C#.

In de volgende lesbrieven zul je leren over de programmeertaal C#. De lesbrieven zijn beperkt tot een aantal concepten die relevant kunnen zijn als je in Unity gaat scripten.

**Om te beginnen**

* Installeer Visual Studio IDE - Community Edition: <https://visualstudio.microsoft.com/vs/community/>
* Tijdens de installatie kun je diverse “Workloads” installeren. Kies voor *.Net Desktop Development*.

## 

## Lesbrief 1: Een Hello World programma

| \ ^\_\_^  \ (oo)\\_\_\_\_\_\_\_  (\_\_)\ )\/\  ||----w |  || || | * Probeer de voorbeelden uit. * Voer de opdrachten uit. |
| --- | --- |

In deze lesbrief maak je kennis met de Visual Studio IDE. Je gaat leren

* Hoe een nieuw project te beginnen.
* Hoe een eenvoudige Hello World programma te schrijven.
* Hoe een programma te runnen.

We beginnen met een “Hello World” programma, een programma die de tekst “Hello World” print op een console scherm. Omdat wij iets op een console scherm laten zien wordt dit een *Console Application* genoemd.

### Een nieuw solution en project aanmaken

In Visual Studio begin je eerst met een nieuwe “solution”. In je “solution” maak je het project aan.

* Start Visual Studio.
* Selecteer *Create a New Project* van het startscherm.

Het kan zijn dat je nog eerst een “Blank Solution” moet maken:

* In de zoekbalk type “Blank Solution”.
* Selecteer vervolgens “Blank Solution”.

Bovengenoemde stappen en schermvoorbeelden zijn te vinden op de Microsoft Visual Studio website:

<https://docs.microsoft.com/en-us/visualstudio/get-started/tutorial-projects-solutions?view=vs-2019>

<https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio?tabs=csharp>

* Selecteer vervolgens het type project: *Console Application*.
* Geef het project een goede toepasselijke naam bijvoorbeeld *HelloWorld.*
* Selecteer de locatie op je harde schijf waar het project wordt opgeslagen.
* Geef eventueel de solution een naam bijvoorbeeld *HelloWorld*.
* En klik op *Create*.

Visual Studio heeft een project aangemaakt en deze op de opgegeven locatie opgeslagen.

### De Code

Bij het aanmaken van een project creëert Visual Studio een bestand genaamd *Program.cs* - *cs* staat voor *CSharp*. Dit bestand bevat reeds de **basisstructuur** voor je programma:

| using System;  namespace HelloWorld {  class Program  {  static void Main(string[] args)  {  }  } } |
| --- |

**Wat houdt deze structuur in?**

De methode *static void Main(string[] args)* is de hoofdmethode van je programma. Als je jouw programma opstart dan wordt deze *Main* methode als eerste aangeroepen en de code in deze methode wordt uitgevoerd.

Deze methode staat in een *class Program*. In een latere lesbrief wordt nader ingegaan op classes.

Code voeg je dus toe aan de methode *Main*:

| static void Main(string[] args)  {  *//voeg hier code toe*  } |
| --- |

Om een console scherm te openen met de tekst “Hello World”, voegen we de volgende code toe aan de methode *Main*:

| using System;  namespace HelloWorld {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Hello World!");  }  } } |
| --- |

De code *Console.WriteLine("Hello World!")* print de tekst “Hello World” op een console scherm.

We gaan het programma uitvoeren

* Klik op Start of run bovenaan het scherm (meestal is dit een groene “play” knop).

Je zult merken dat er een scherm wordt geopend die meteen weer sluit. Om ervoor te zorgen dat het scherm langer open blijft voegen we een regel code toe.

| using System;  namespace HelloWorld {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Hello World");  Console.ReadLine();  }  } } |
| --- |

De code *Console.ReadLine();* leest input van je Console scherm. Dit zorgt ervoor dat het scherm niet meteen afsluit. Het scherm sluit pas af als je een letter of cijfer gevolgd door *Enter* intypt.

Je hebt nu een “Hello World” programma gemaakt.

## Figuren maken

Nu we weten hoe “Hello World” op een console scherm te printen kunnen we ook figuren maken.

Hieronder zie je in code een voorbeeld van een driehoek.

| using System;  namespace HelloWorld {  class Program  {  static void Main(string[] args)  {  Console.WriteLine(" /|");  Console.WriteLine(" / |");  Console.WriteLine(" / |");  Console.WriteLine("/\_\_\_|");   Console.ReadLine();  }  } } |
| --- |

| **Opdracht**  Schrijf een console applicatie die het volgende figuur toont (cowsay):  \ ^\_\_^  \ (oo)\\_\_\_\_\_\_\_  (\_\_)\ )\/\  ||----w |  || ||  Let op: Op **windows** gebruik je een dubbele “\”:  \\ ^\_\_^  \\ (oo)\\_\_\_\_\_\_\_  (\_\_)\ )\\/\\  ||----w |  || ||   * Bedenk zelf een figuur en programmeer deze. |
| --- |

## Input - output

**WriteLine() - Write()**

Om karakters op het scherm te tonen hebben we gebruik gemaakt van de methode *WriteLine()*. Deze methode schrijft de meegegeven tekst naar het scherm, inclusief een <enter>.

Er is ook een methode *Write()*, deze schrijft de meegegeven tekst naar het scherm zonder een <enter>.

**ReadLine() - Read()**

Om input van het scherm te lezen hebben we gebruik gemaakt van de methode *ReadLine()*. Deze methode leest een regel van het scherm.

Er is ook een methode Read(), deze leest 1 karakter van het scherm.

Beide methoden zijn onderdeel van de klasse **Console**. Met andere woorden WriteLine, Write, ReadLine en Read zijn methoden van de klasse Console.

**Voorbeeld**

Onderstaand programma vraagt een gebruiker om zijn of haar naam en school en drukt deze vervolgens af. De ingevoerde naam en school worden in de variabelen *naam* en *school* onthouden.

| using System;  namespace InputOutput {  class Program  {  static void Main(string[] args)  {   Console.Write("Wat is jouw naam? ");  string naam = Console.ReadLine();  Console.Write("Waar zit je op school? ");  string school = Console.ReadLine();  Console.WriteLine("Hallo " + naam + " jouw school heet " + school);   Console.ReadLine();  }  } } |
| --- |

# Samenvatting

| Een nieuw project starten | File -> New -> Project of Create a New Project vanaf het startscherm. |
| --- | --- |
| Console Application | Applicatie type. |
| static void Main(string[] args) | De hoofdmethode van je programma. |
| Console | Is een klasse (Class). |
| Console.WriteLine() | WriteLine is een methode van de klasse Console.  Schrijft tekst naar het scherm, inclusief een enter |
| Console.ReadLine() | ReadLine is een methode van de klasse Console.  Leest een regel van het scherm. |
| Console.Write() | Write is een methode van de klasse Console.  Schrijft tekst naar het scherm zonder een enter |
| Console.Read() | Read is een methode van de klasse Console.  Leest een karakter van het scherm. |