DOTNEXT

Building an educational game with .NET Core and Unity

Johnny Hooyberghs

@djohnnieke

johnny.hooyberghs@involved-it.be

Who am I

- Johnny Hooyberghs
- @djohnnieke
- github.com/Djohnnie

involved

www.involved-it.be

What and Why

Why did I make this game?

- ***Gamification*** to teach ***basic C# knowledge*** in school
- ***Learn*** the ***Unity Game Engine***
- ***Play*** with the ***Microsoft Compiler Platform*** (Roslyn)
- ***Play*** with ***.NET Core 3***
- ***Play*** with ***Azure DevOps Pipelines, Docker and Kubernetes***
- Use ***Test Driven Development***

What is CSharpWars?

- ***Robots*** on ***arena*** (15x15)
- Robots take ***turns*** (every two seconds, simultaniously)
- Robots can ***move*** around the grid
- Robots can ***attack*** other robots
- Robots can ***see*** part of the arena
- A turn is ***scripted*** using C# and ***all scripts*** will ***run every two seconds***
- A robot has a ***limited*** amount of ***health*** and ***stamina***

Let’s play!

- Register as a player
- Select a pre-defined script
- Create your robot
- Watch him struggle!

https://api.djohnnie.be:8802/

Architecture

What about the architecture?

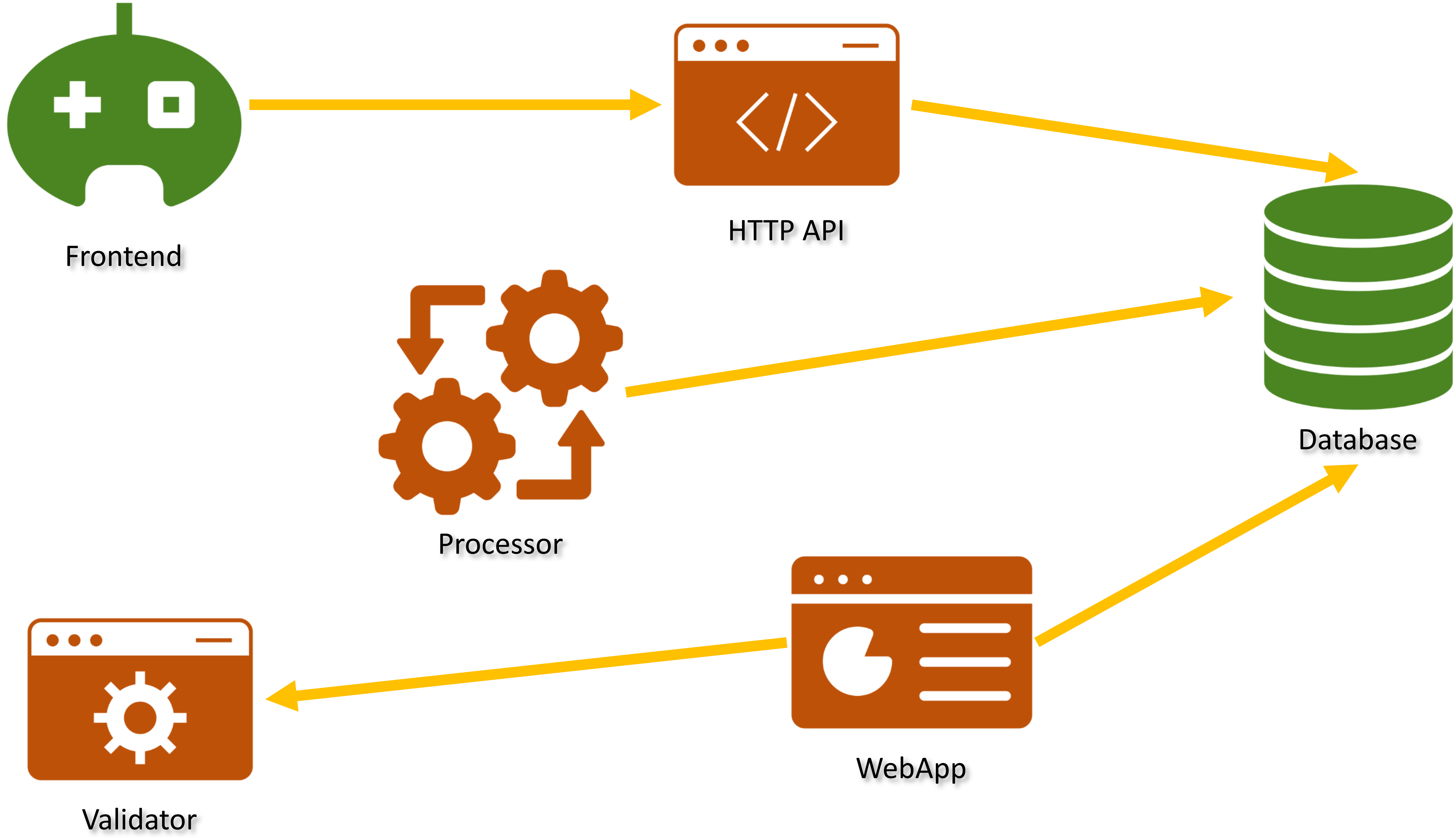
- The Unity game engine for the arena frontend
  - I wanted a full 3D experience
  - I don’t need a full game experience, just a visualization of the battle
  - I can fetch the game state every two seconds from an HTTP backend
  - I can learn Unity!

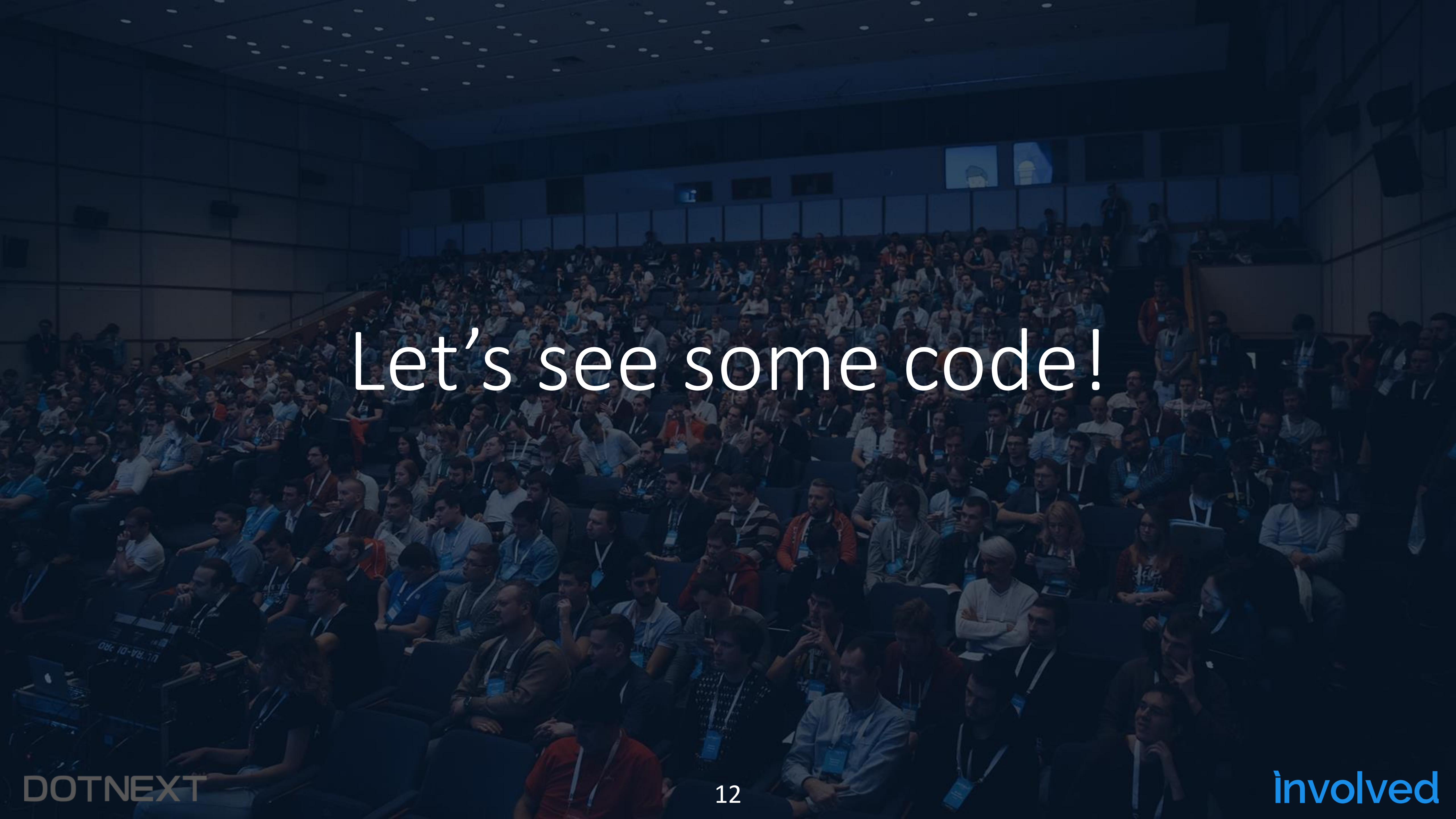
What about the architecture?

- ASP.NET Core for the backend
  - An ASP.NET Core WebApi HTTP API for Unity requests
  - An ASP.NET Core MVC for the demo website
  - An ASP.NET Core gRPC service for the validation service
  - I can learn gRPC!

What about the architecture?

- .NET Core Worker Service for the processing middleware
  - Worker service template for Windows Services, Linux Daemons or docker containers
  - The Microsoft Compiler platform for C# compilation and execution at run-time
  - I can learn Roslyn!


Frontend
HTTP API
Processor
Database
WebApp
Validator



Let’s see some code!

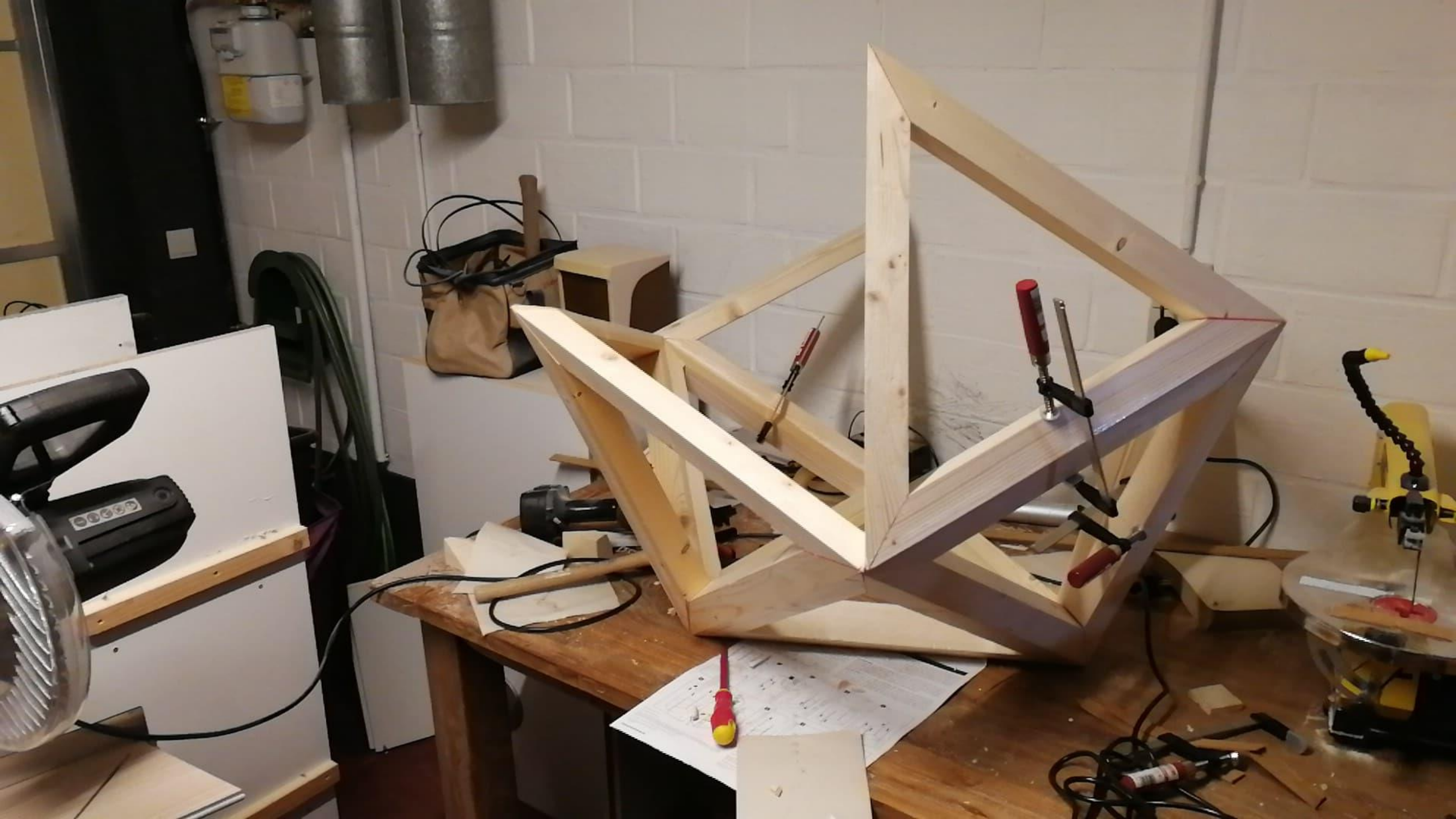
What did I learn?

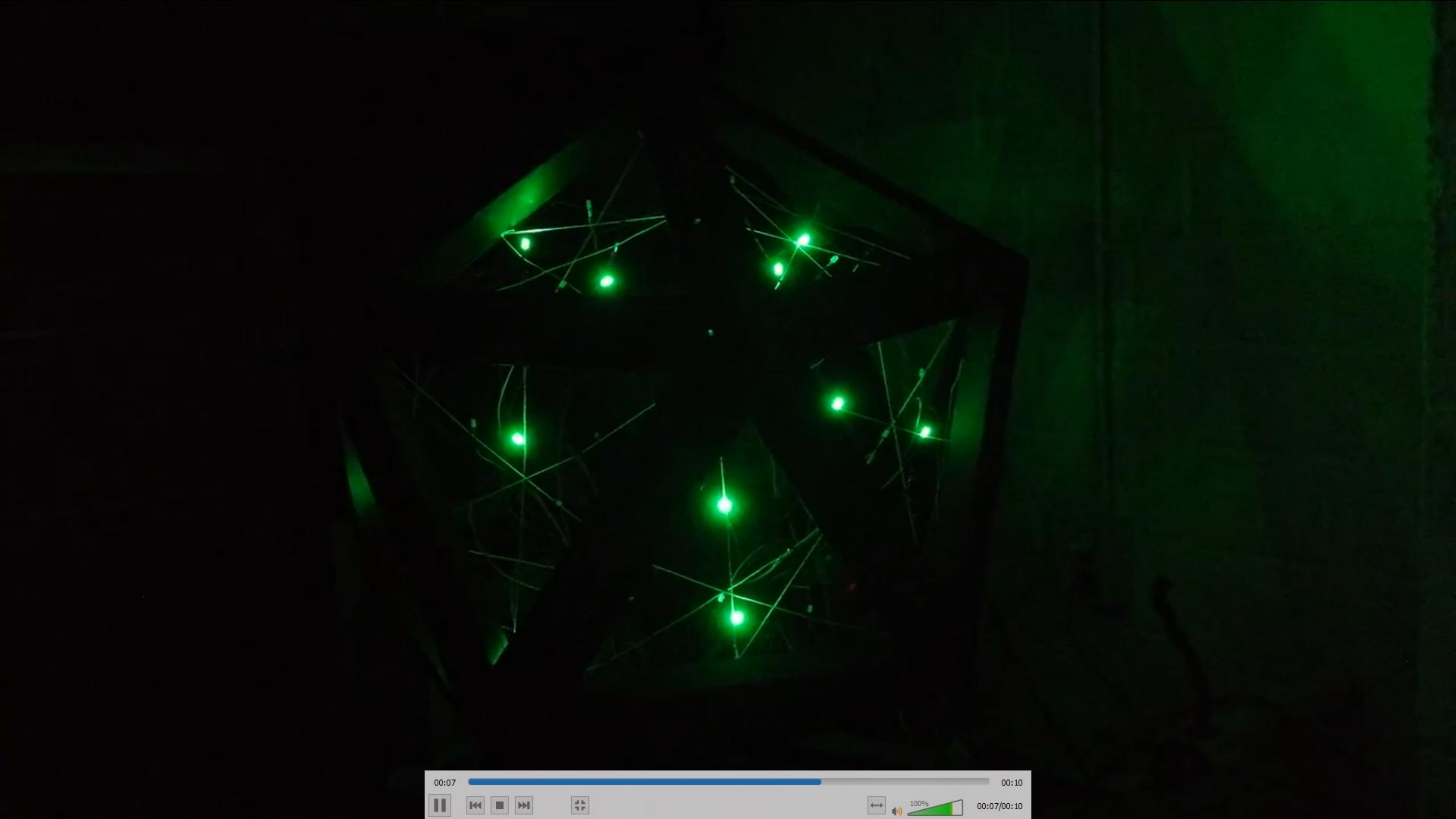
Unity!

HACK THE FUTURE 2018

NEW EDITION - NEW THEME!

Working on the this years theme for Hack The Future 2018. Curious yet? To be announced next week. We'll keep you posted.





00:07
00:10
100%
00:07/00:10

61

What did I learn in Unity?

- **Unity = C#**
  - I can use my **current knowledge**
  - I can use **external assemblies** and **libraries** (NuGet, …) but no direct support for NuGet
  - Unity uses the **Mono runtime**, so no .NET Core
  - Unity also provides the option to compile to native using **IL2CPP**

What did I learn in Unity?

- **Modular workflow** based on **GameObjects**
  - I had to get used to the workflow, but I am getting used to it
  - Lot’s of **flexibility** makes it **easy and hard** at the same time
  - Unity has a **wide range** of **support** for **3d models** (meshes), even Sketchup
  - There is community support for **dependency injection** (ADIC, Extenject, …)

What did I learn in Unity?

- **Platform independent**
  - Support for a number of **platforms** thanks to the **Mono runtime**
  - Support for even **more platforms** thanks to **IL2CPP**

What did I learn in .NET Core?

- Using **.NET Core with Docker containers** is extremely easy
- **.NET Core** is **more performant** than **.NET Framework**

What did I learn in gRPC?

- For **service to service communication**, it is perfect and easy to use with support for async streams

- **Better performance** thanks to **smaller payload** size (binary)

- Only **experimental support for Unity**

What did I learn in Roslyn?

- Using the Microsoft Compiler Platform is **easier than expected**
- Keeping **memory usage** low is a bit of a search
- **Infinite loops** are not easy to detect
- Running **scripts cannot be cancelled**
  - Use the validator service to check the scripts without damaging the processing middleware
  - Hosting the validator and middleware processor inside a docker container makes it easy to quickly restart

Thank You

johnny.hooyberghs@involved-it.be
@djohnnieke

https://github.com/Djohnnie/CSharpWars