

# Riko Ophorst: Curriculum Vitae

---

Phone Number (+31) 6 48 21 81 82  
Website [rikoophorst.com](http://rikoophorst.com)  
Email [info@rikoophorst.com](mailto:info@rikoophorst.com)

## Goal

To get an internship at a game studio where I can focus on graphics programming or game engine programming for 18-20 fulltime weeks starting around September 2018.

## Education

2012-2015 Media & Game Development, SiNTLUCAS Eindhoven

2015-2019 [BSc, Computer Science, International Game Architecture & Design, NHTV Breda University of Applied Sciences](#)

## Employment History

Jan 2014 – Jul 2014 Spil Games  
*Intern HTML5 Game Developer*

Intern HTML5 game developer on a team of around 15 people. Contributed to various projects for Spil's mobile gaming platforms using HTML5 and JavaScript. Responsibilities included: developing scalable UI for dynamic resolutions, developing back-end tools using NodeJS, building tutorials in the games we made. Heavily involved in the process of developing the core game mechanics for the projects I worked on. Built tools to facilitate a faster production pipeline, for example a way to easily convert source assets to usable assets in game. Built unit-tests according to TDD standards to ensure stable builds.

**Technologies Used:** JavaScript, NodeJS, GruntJS, HTML5, Scrum, JIRA.

Sep 2014 – Feb 2015 Cool Games (previously BoosterMedia)  
*Intern HTML5 Game Developer*

Intern HTML5 game developer on a team of around 15 people. Worked on various projects for the mobile gaming platform using HTML5 and JavaScript. Responsible for developing UI for dynamic resolutions, back-end tools using NodeJS, fixing bugs. I also was heavily involved in the process of developing the core game mechanics for the projects I worked on. Developed clopp, a custom tool to precompile JavaScript. Built tools to easily design scalable UI using the Cocos2D UI editor.

**Technologies Used:** JavaScript, NodeJS, HTML5, Scrum, Cocos2D, Regular Expressions, JIRA.

## Skills

- Intermediate to advanced knowledge of **Graphics Programming** using **DirectX 12**, **DirectX 11** and **OpenGL**.
- Advanced knowledge of **Renderer Architecture**.
- Advanced knowledge of **Object Oriented Programming** and **Data Oriented Programming**.
- Advanced knowledge of **Game Engine Architecture**.
- Advanced knowledge of **Unity3D**.
- Advanced knowledge of **Scrum** and tools like **JIRA**, **Trello** and **Pivotal**.
- Advanced knowledge of **Programming Languages** like **C++** and **C#**.
- Intermediate knowledge of **Shading Languages** like **HLSL** and **GLSL**.
- Intermediate knowledge of **Embedding Scripting Languages** like **Lua**.
- Intermediate to advanced knowledge of **Version Control Systems** like **Perforce** and **Git**.
- Intermediate to advanced knowledge of **HTML5 Game Development** using **JavaScript**.

## Selected Personal Projects

### Project Sulphur

I was the **project lead of 15 people** and **graphics programmer** on Project Sulphur, which is a game engine with support for **PlayStation 4** and **Windows** using **DirectX 12**, **DirectX 11** and **GNM**. My main contributions to this project are: **leading the team**, building the **Skeletal Animation System**, organizing the **planning**, designing the **Renderer Architecture**.

**Keywords:** C++, C#, HLSL, PSSL, JIRA, PBR, Deferred Rendering, DirectX 12, GNM, PlayStation 4, Skeletal Animations.

### Blowbox 2017

Blowbox is an ongoing personal game engine project. It's targeted for **Windows** only using **DirectX 12**. Most important contributions to this project are: the **advanced GPU resource & descriptor management**, the **advanced command list management** and the debugging tools like a **live performance & VRAM profiler**.

**Keywords:** C++, CMake, ImGui, DirectX 12, DirectX 11, GNM, PlayStation 4, Windows, HLSL.

### Tremble

Tremble was a 16-week game engine project in which I was the **project lead of 7 people** and the sole **graphics programmer**. I implemented an **advanced rendering architecture using DirectX 12** with support for **skeletal animations**. I spend some time on optimizations as well: I implemented **hierarchical view frustum culling** using **octrees**, as well as a **depth pre-pass** to reduce pixel shader invocations.

**Keywords:** C++, DirectX 12, Skeletal Animations, HLSL, Hierarchical View Frustum Culling, Depth Prepass, Advanced Rendering Architecture.