

## Personal reflection Personal Portfolio Nils Meijer ECM2V.Ec 466301

### Hour registration

#### Learning goal 1

Learning goal 1	Date	From-to	Hours	Activity	Learn
(Web game)	15-Dec	19.00 - 21.00	2	Starting project, setting up Three.js	(Unre
	16-Dec	17.00 - 18.00	1	Creating basic player controller	
	17-Dec	14.00 - 15.00	1	Working on movement	
	20-Dec	9.45 - 11.30	1.75	Working on movement	
	20-Dec	13.30 - 16.00	2.5	Fixed movement (left/right)	
	21-Dec	19.30 - 20.30	1	Trying to add Physi.js (physics lib)	
	22-Dec	9.30 - 12.00	2.5	Added Cannon.js (other physics lib)	
	22-Dec	13.15 - 15.00	1.75	Implementing physics	
	11-Jan	13.30 - 16.00	2.5	After a (too) long break from this LO, trying to implement physics again	
	12-Jan	13.00 - 18.30	5.5	Implemented physics, jumping, cleaned up code	
	13-Jan	13.00 - 16.00	3.5	Worked on level	
	17-Jan	10.30 - 12.00	3.5	Vertical UI movement	
	18-Jan	10.30 - 13.15	2.75	Working on physics/contact materials	
	26-Jan	12.00 - 20.00	8	level, UI movement	
	27-Jan	12.00 - 19.30	7.5	Added tweening lib. Implemented door functionality, replaced HTML UI with THREE text	
	28-Jan	14.00 - 17.00	3	Cleaning code, working on scene loading	
	31-Jan	10.00 - 16.00	6	Implemented scene loading	
	02-Feb	12.00 - 15.00	3	Writing reflection	
		total hours spent=	58.75		

#### Learning goal 2

Learning goal 2	Date	From-to	Hours	Activity
(Unreal Engine)	15-Dec	12.30 - 14.45	2.25	Exploring UE5, creating scripts
	21-Dec	8.45 - 12.00	3.25	Worked on character controller
	22-Dec	15.45 - 17.00	1.25	Implemented box trigger
	04-Jan	10.30 - 12.00	2	Restarted in template project (remade trigger box & widget)
	04-Jan	16.00 - 17.00	1	Started level design
	05-Jan	9.30 - 13.00	3.5	Worked on level
	05-Jan	15.30 - 17.00	1.5	Worked on level
	06-Jan	10.30 - 12.30	2	Worked on level
	06-Jan	18.00 - 21.00	3	Recreated project after crash
	07-Jan	10.30 - 13.00	2.5	Continuing level
	07-Jan	22.00 - 23.00	1	Worked on level (puzzle)
	08-Jan	21.00 - 23.00	2	Implemented first puzzle
	09-Jan	21.45 - 22.45	1	Implemented game over trigger
	10-Jan	10.30 - 13.00	2.5	Added lava
IE text	12-Jan	9.00 - 12.30	3.5	Worked on level
	13-Jan	8.30 - 12.00	3.5	Worked on level
	28-Jan	11.00 - 13.30	2.5	Menu - level transition
	01-Feb	12.00 - 14.00	2	Full scene transitions winning/losing
	02-Feb	9.00 - 12.00	3	Writing reflection
		total hours spent=	43.25	

# Process reflection

## Learning goal 1

It took me a while to get the “three.js” library working. I was trying multiple methods of importing libraries in a project, but none of them worked (I have no idea why, it’s really not that hard by using a CDN). I ended up using a CDN (Content Delivery Network). I hadn’t thought out exactly what the game was supposed to look like or any other details like that (which I should have done, it would have been useful to have a detailed “level” design already). So I started off trying to get to know the three.js library as that was the first experience I had with it, by following tutorials and the (relatively extensive) documentation. I quickly found a physics library were pretty essential if I wanted to have collision detection, gravity etc. I first found “physi.js”, but ended up with “cannon.js”.

During the Personal Portfolio course last term, I made a (very janky, buggy and bare-bones) web platformer as well. Although that was more in a sense of being able to move left & right and seeing UI move along with the player. It had no physics, animations or anything of the sort. But to get to the point: when the player would move, the UI would move as well, as mentioned before. However, since the UI was part of an HTML document and not the three.js world, the displacement wasn’t consistent and would move more than the player would move, resulting in a messed up UI. So I found that three.js allows TextGeometry objects to be created. Physically (although without a collider, wasn’t necessary) present in the world, and therefore proper displacement when the character would be moving around the world.

I wanted to add animations to spice things up, so I found the tweening library “tween.js”. It allowed me to make a “door” functionality, which would open when the character approaches (I thought it would be nice if certain areas of the game were separated from each other, resulting in a clear distinction of areas).

While working on the portfolio area (with all the different portfolio areas presented in the world), I quickly realized it would either become *very* messy trying to add all the information, images/videos and everything else there was to tell about a specific portfolio item (and that \* x amount of items), or not being able to add any information at all and just display a screenshot and the project’s name (in which case there’s no way that’s enough for a *portfolio* website). So, I had to find a way to get enough space for every single portfolio item’s data. The most logical and easy to implement solution for that was to switch to a new scene entirely, and just switch back when you want to go back to the “main area”. That means every portfolio item’s scene is a new script deriving from parent class “THREE.Scene”, allowing for extensive customization (of course, there can be some generalized scene settings so a template can be created from that, to use over and over). To allow for a smooth transition to the new scene (and back, when that happens), I added an image covering the whole window and set the opacity to 0. Then, when the scene-switching actually happens, the image fades out and back in when the new scene has loaded (which is instantly because the “loading” is really just setting a different “activeScene” for the renderer in the main Javascript file.

I was (and still am) struggling with all the visuals in the game and trying to make it looking smooth and beautiful (unfortunately I don’t quite have the capabilities of an artist), so during the project & holiday next term (already passed it last year), I’ll be filling the game with portfolio items and other content,

implementing more functionalities and polishing the visuals. I am planning to buy a start-to-end three.js course (granted, I have already surpassed the “start” point, but I think there’s still a lot of useful things to get from this course; mainly, how to make it look great. And, not unimportant, improve performance, especially because I have plans to allow for mobile usage of my website as well.)

All with all, I think the result is great, and is unlike many/most other portfolio websites, even though it needs a lot more work and polishing.

## **Learning goal 2**

This was my first experience with Unreal Engine (I don’t count the one I had 2 years ago where I was intimidated by the UI, and as a much less experienced developer back then, it was better to stick with Unity for a while longer).

Initially, I wanted fully work in C++ (or rather, the Unreal version of it) but as I kept working, I found learning a new engine was already (I wouldn’t say overwhelming, but it was definitely intense) a challenge. Besides that, it had been quite a while since I wrote any C++ code (during the C++ course last year), so getting back into it while also learning a new engine was too much. I decided to try out the very extensive visual scripting/blueprint system, which I have to say is very intuitive and rather easy to get into and use.

I haven’t made as many puzzles or challenges as I would have liked, since my other learning goal and courses took more time than I expected (although I still exceeded the minimum time goal I had set, for both learning goals). Then again, the point was to learn Unreal Engine, which I definitely did, even though I only scratched the surface (especially with all the new features of UE5, which I am eager to get into with future projects, possibly with future PP courses. Perhaps similar to the 3D Art course of the artists in Unity?). I now understand how the basics work, like collision, scene loading and movement.

I started with an orange-ish plane for the lava, but I didn’t like the way it looked (it wasn’t even clear it was lava), so I decided to find a free asset for that. Although I forgot to ask if it was okay to use 3<sup>rd</sup> party assets, I expected it would be treated the same as other courses (as long as it’s free and a source link has been given; see the list at the end of this document) and didn’t think it would be an issue.

## **Masterclasses reflection**

### **Version Control/GIT (Yvens Serpa)**

I already followed this masterclass last year, and I probably could have transferred it but I didn't write a reflection and I didn't mind following it again so I did. For the most part, he went through the basics of GIT (creating a repo, pushing, pulling, the difference between local and origin commits). While Yvens is a fan of the terminal/command line, I still prefer just using the GUI. I feel like it's much more work to make changes with the command line, than it is with the GUI of the software. That being said, it was a nice refresher about how GIT works. Still, my version control during this course could be improved (I divided up my learning goals into 2 different branches after 2 weeks or so instead of at the beginning).

### **Lighting a scene in Unity (Mark Schipper)**

While I am not an artist, I felt like it would still be useful to improve my knowledge on the lighting system of Unity. During the project of term 3 last year/ year 2, I was tasked with doing the lighting in Unity because the other engineer and me had the most experience in Unity (getting back to this) and I was able to free time for it. However, we were doing the project in HDRP and I underestimated the pressure HDRP puts on the CPU and GPU. What I did was throwing a point light in every area of the game (and as I found out during my THREEjs learning goal, that means the engine has to calculate lightmaps for 6 different directions, and then that's " \* the amount of pointlights".....), which caused performance problems in the last few days of the project. So that meant I stayed up during the late hours of the night, trying to improve performance.

Basically, the masterclass showed a lot of details on how to improve the lighting and ways to actually prevent performance problems. Which means not carelessly creating lightsources and thinking it'll be alright.

### **Game Development in Unreal Engine (Javid Ladhani)**

Even though this one apparently counts towards the masterclasses of term 3, I might as well do the work now (just to be sure add it to this document as well), and add it to the reflection next term.

Since this masterclass fitted well with my 2<sup>nd</sup> learning goal, it was nice to get some insights in how the engine works, the history & iterations of the engine (including what has improved after the pre-release of UE5). While Javid used UE4 for his demonstrations, it still helped while trying to find my way in UE5.

The last part of his masterclass, he spoke about programming C++/Unreal Script, which was interesting to watch, especially as an engineer. Even though I ended up not using it in my project after switching to the blueprint system, it was still quite nice to get to know.

## Did you complete your learning goals?

Since I already elaborated rather extensively on this in the “Progress” paragraphs, I won’t go too much into detail here. I definitely think I completed both my learning goals. I now have a foundation (which still needs a lot of work but I didn’t expect otherwise after a set goal of ~50 hours) for my portfolio website. As for the other learning goal, I’ve succeeded in making a small game with 1 main goal (reaching the end of the level before the lava catches up with you) and 1 main mechanic (solving puzzles to unlock passageways), while familiarizing myself with the engine and the programming aspect (in this case, blueprints).

## What do you want to work on next term?

I haven’t given too much thought about this yet. When I had my SLB meeting, my coach suggested I would fix/remake my current portfolio website in term 3 (<http://www.nilsmeijer.com/Website/html/>), but I’m not planning to do that. It would mean 1.5 terms of Personal Portfolio would go to waste (Learning goal 1 & 2 of Q1 – UI/UX design, and making a platformer game but without any 3<sup>rd</sup> party libs such as THREE -, and Learning goal 1 from Q2), because those were focused on making this “game”/interactive website.

Besides that, I don’t want a rather “standard” website when I start applying for internships. Instead, I’ll need something that stands out; which is exactly what this website foundation does.

Instead, I’m considering making a VR game (in Unity), as I’ve never done that before and when I start applying for internships, it’d be nice to have a more diversified portfolio which includes VR. Especially if the company I’m applying to specializes in VR.

I don’t know yet if that VR game would be divided up into 2 – or even more – learning goals or if I’m going to come up with something else for other learning goals.

Perhaps an AR project could be useful as well, for more diversification of my portfolio.

## What do you need for year 3 & 4 and beyond?

As said before, I’ll need a new & up-to-date- portfolio website. During the next project & holiday, I’m going to continue working on the current website foundation. To get the visuals and performance I want, I am planning to buy a THREEjs course (<https://threejs-journey.com/> if you’re curious). The maker of this course (<https://bruno-simon.com/>) is also part of what inspired me to make this website in the first place). Even though the course assumes the “student” has no JavaScript or three.js experience whatsoever, I’m quite sure there will be plenty of useful things to learn from it.

## **Indication of your learning outcomes for your next Personal Portfolio?**

Learning goal 1: Learning how to make a VR game/application

Learning goal 2: if I'm not splitting the 1<sup>st</sup> goal up (perhaps something like level design), I'm considering to work on an AR project here.

## **Resources/assets used:**

### **Learning goal 1:**

Font (link): <https://fonts.google.com/specimen/El+Messiri>

Threejs: <https://cdnjs.cloudflare.com/ajax/libs/three.js/r128/three.min.js>

Cannonjs: <https://cdnjs.cloudflare.com/ajax/libs/cannon.js/0.6.2/cannon.min.js>

Tweenjs: <https://cdnjs.cloudflare.com/ajax/libs/tween.js/18.6.4/tween.umd.js>

### **Learning goal 2:**

Dungeon pack: <https://www.unrealengine.com/marketplace/en-US/product/infinity-blade-fire-lands>