# Reflection report

In this document, I will be reporting on my graduation work.

## Time management

For time management and project planning, I have chosen to use HackNPlan. I learned to work with it during Game Project and Group Project. It is a well-known management tool with many easy-to-use and useful features that make planning for any kind of project more efficient. I have also started to use it for other portfolio projects in my free time, so I have an extended knowledge of how to organize and arrange my tasks properly.

HackNPlan is very useful for keeping track of what tasks have been done, which tasks are in progress, and what tasks still have to be started. The categories are also a good and useful tool, since they grant a clear visual indicator of what type of work a task will be.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Risk analysis

### No supervisor/coach:

**Risk**: Due to this being a retake, the power of the teachers is, of course, lessened, and the regular pipeline of gradwork is not applicable. This means that a lot of stuff has to be figured out by myself without the help of a supervisor or coach.

**Solution**: I reached out to students that already completed gradwork for feedback and guidance with my project.

### Complexity of subject:

**Risk**: I had to switch topics during the semester due to the complexity of my first choice. There was an unexpectedly steep learning curve that I was sadly unable to overcome, due to a lack of good resources to learn from. Perhaps with more time, I would have been able to continue with this work, but it was not wise to spend so much time on a project I was not fully sure I was going to be able to complete.

**Solution**: The change of my topic to something I am already more familiar solves the issue this time, but for future projects I will keep in mind I have to try to find as much research and resources on the topic before fully committing to it, or be more aware that without that much information to start with, I will have to take into consideration the time that will be spend figuring some aspects out and then plan that into my schedule for the project.

### Overscoping:

**Risk**: For my paper, I will be implementing and comparing 3 different types of ambient occlusion generation at runtime. It is, of course, a possibility that not all types of ambient occlusion will be covered because of a lack of time.

**Solution**: I will have to realize when to cut my losses and drop one of the methods of ambient occlusion. This will make the experiment less interesting, of course, but it will still be a very useful study for future projects.

## SWOT

### Strengths:

-The Vulkan API is very well documented, giving me a lot more resources to work with.

### Weaknesses

-Due to ambient occlusion being an approximation, there is no real standard to compare it to.

### Opportunities

-This graduation work focuses on graphic programming, which is the subsection of programming I am most interested in. This means this project has already taught me a lot about the sector I want to work in, helping me become a better graphics programmer later in life.

-This project is a great portfolio piece that will help me get my name out there as a graphics programmer.

### Threats

-Because my subject is mostly analysis of framerates, CPU, and GPU usage, there will not be any peer-reviewed metrics.

## Future work

Programming ambient occlusion can be done in many different ways, even if you try to accomplish the same thing, different people will have different ways to get there. Because of this, there is no one true way of programming something and the tiniest difference can have an impact on performance. This means that my code is most likely not as optimized as it can be and this will have an impact on the results of the experiment.

A lot can still be done in terms of ambient occlusion, there are still many types out there I haven’t touched on yet, like:

-HBAO+: An advanced version of HBAO proposed by NVIDIA

-RTAO: Ambient occlusion that uses ray tracing

-SSDO: An advanced version of SSAO

-SEAO: A unique take on ambient occlusion from the book GPU Gems 2

## Reflection

This project went a lot smoother than my previous attempts at gradwork, I worked more consistently, I got faster results, etc.

There were moments where I lost some hope because I tried adding tiny features to my renderer, and this ended up as a whole overhaul because the underlying code was not the best, but the overhaul was worth it because now the code is a lot cleaner and easier to work with.

I learned a lot about graphics programming, programming in general, ambient occlusion, debugging, and a lot more. This project definitely had a good impact on my career as a programmer.

Next time, I would do most of the things I did the same, but I would focus more on project planning and my learning log. My project planning was good in terms of listing the tasks I had to do, but the actual planning of when I would do what tasks was very lacking. My learning log is also very lacking and is more of a work log where I note down what I did on what days.