# Monday

## 7:15-7:30 Fill out today's plan

## 7:30-8:00 Gloss over the chapter about the ray tracer

My assumption was correct. The last part of the chapter discusses the math behind the ray tracer. Formulas for shading and intersections. I will read those through when I'll start creating the ray tracer.

## 9:30-10:00 Discuss learning log planning

## 10:15-10:45 Help pears with SFML

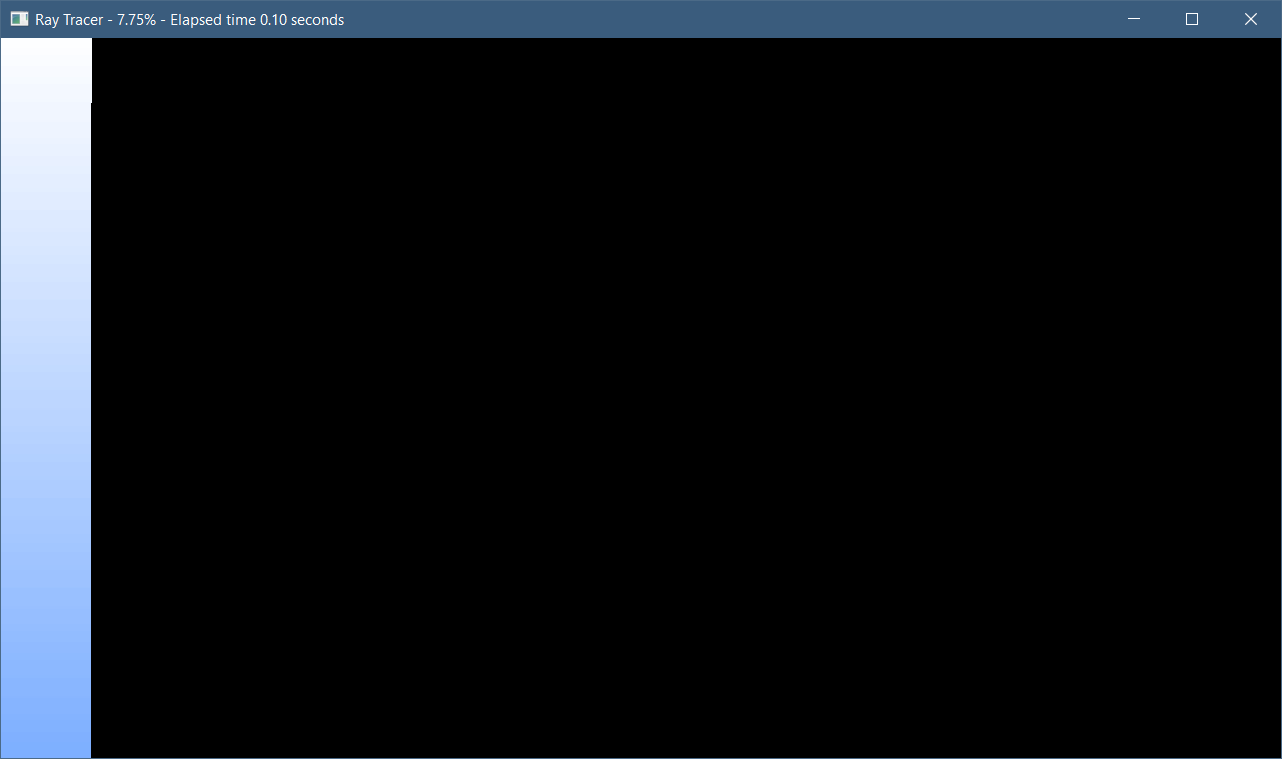
## 11:15-12:45 Fill out last Friday's and today’s planning

## 12:45-13:00 Write pseudocode for the ray tracer

## 12:45-14:15 Write UML for the ray tracer

## 

## 14:15-16:00, 18:15-19:00 Translate design into code

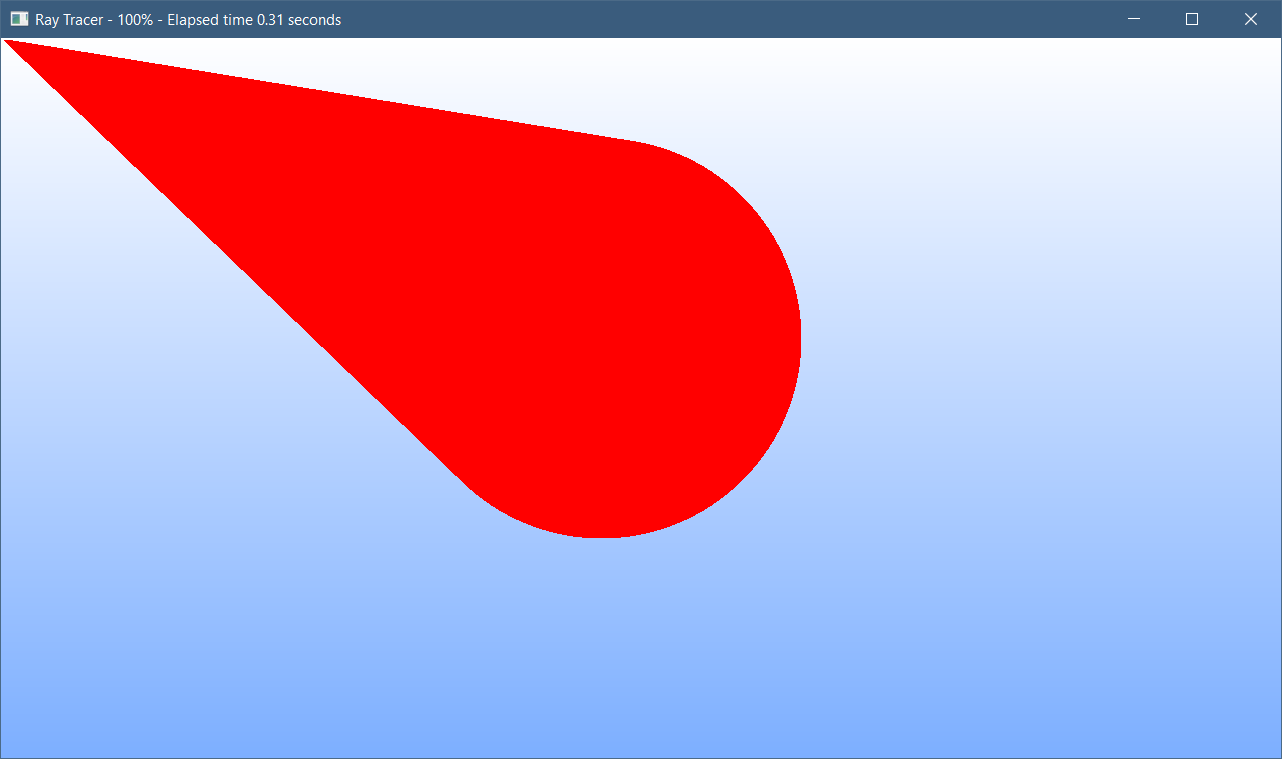


I got the basic renderer working. The forward rays are being cast. I can’t seem to implement objects properly. I’m still researching why it doesn’t work.

# 

The discriminant is never printed. This is incorrect because this means that the sphere is always intersected. I will gloss over my implementation again

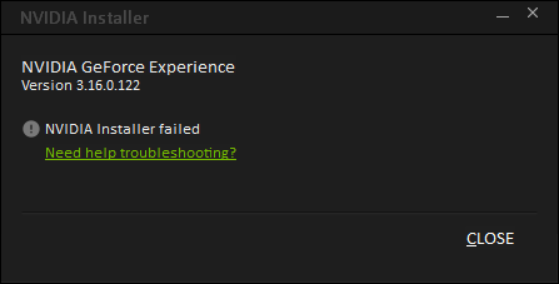
## 20:30-20:45 Try to fix the ray sphere intersection

I have noticed that the t value (the value that is supposed to show the point of intersection) is always negative, which means that the circle appears behind the ray. Also, for some reason giving the sphere a negative position moves it in the supposed direction. In this case, the sphere has a position of -400 and -600. I probably made a mistake while translating the formula from paper to code. I will try to fix this tomorrow

# Tuesday

## 9:15-9:45 Create daily planning and finish up yesterday’s work log

## 9:45-12:00 Update Visual studio, video drives and shelve files to computer and make food



While I don’t need the most up to date video drivers. It may help in the project (I doubt it will). Normally we don’t fix things that are not broken. But Abhishek told us that we should keep things up to date and I haven’t updated this in a while.

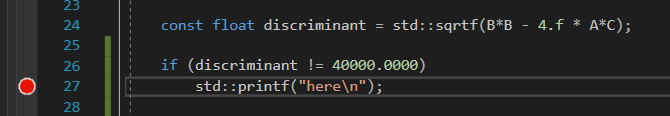
Following the troubleshooting guide hasn’t helped either. I should’ve never updated the drivers.

Going into properties, compatibility, apply to all users and choosing run as admin helped to resolve the problem

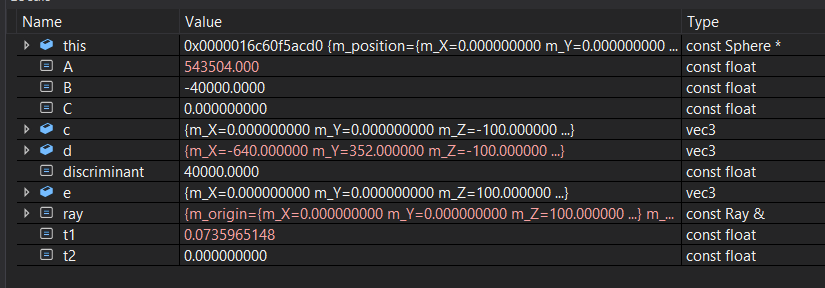
## 12:30-13:15 Fixed progression percentage to represent an actual value.

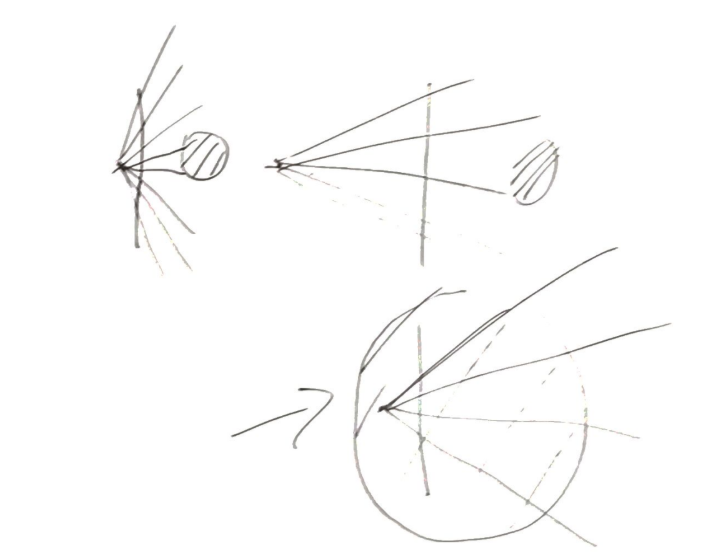
I calculated the pixels wrong. I thought that using the current pixel position and dividing that by the size of the image would result in a correct value. While this logic works, my calculation was incorrect. I added the sizes, resulting in the perimeter value, instead of multiplying the values, resulting in the area value. This little oversight has been fixed.

## 13:15-14:30 Fixing sphere ray intersection formula



For some reason, the discriminant is always equal to 40000. This is odd considering that the only sphere in existence should be in the middle. Worst of all, this sphere is never drawn.



All values stay the same except for A and d. The discriminant should change. It should be 0 because there should be no intersection with the given ray. B should also change because d is changing. 

I found the solution. In my program, the sphere has such a big radius that our viewpoint existed inside of it. Because of this, t2 always returned 0. I thought that this was an error which is why I checked for values lower than 0 but not equal to 0. It was a logical mistake on my part. I solved it by moving the sphere with an absurd amount. I finally got the infinite error, which means no intersection was found. There was never a mistake in translating the formula to code, but the mistake was simply me misinterpreting the information. I could have found this out earlier if I were to draw the solution on paper. I will try to do that next time when I have a problem like this.

The discriminant still never is 0 though. Which is odd and I will research that further.

## 14:30-15:00 Messing around with positions of spheres out of pure curiosity

I really want to make a camera now to move around. I actually think that I’m going to do that. I also want to make my ray tracer data driven so that I don’t have to compile the project again if I want to change the positions of objects. My priorities now lay in creating shading because I want to make the project pop out in 3D.

## 15:15-15:45 Research about discriminant

I always take the square root of the D (discriminant). But that is not what the D is supposed to represent. Because D could be negative, it would, of course, end up in an invalid value. That also explains why the program would return t1 or t2 as an infinite/invalid number. We can’t add invalid numbers.

## 15:45-16:00, 18:00-19:00 Research about shading

* There are many shading models. I will research two very basic ones. Lambertian and Blinn-Phong.
* The important variables for light reflections are a unit vector towards the light, a unit vector towards the viewpoint and a normal of the reflected surface. Material properties can also be important. Things like color, shininess, transparency etc.
* Lambertian shading is an observation of real life. It is very simple and only requires 4 values. The light intensity, the object color, a unit vector from the point of collision to the light source (**l**)and a normal unit vector from the point of collision (**n**).  
  It rests on the fact that when a normal is facing the light source directly, it is more bright, and the further is faces away from the light source (until a 90 degrees angle) it gets darker. We use the dot product to calculate the angle that it faces. We can do that because the shading is proportional.   
  Color = object color \* light intensity max(0, **l**.dot(**n**))

## 20:15-20:30 Research how to get the normal of the point of intersection

It appears to be discussed in section 2.5.4

# Wednesday

## 6:30-7:00 Read section 2.4 about vectors

## 7:00-7:30 Read section 2.5 about calculating the normal

* Learned how you are supposed to interpreter the differentiation formula

## 9:00-9:45 Workshop Google sheets

* Data validation and conditional formatting help you with cleaning your columns or rows up and the data can be used by graphs.
* I learned about formatting.

## 10:00-10:45 Workshop Ray Tracer

* Don’t try to link program lightning concepts to real life, they don’t relate.

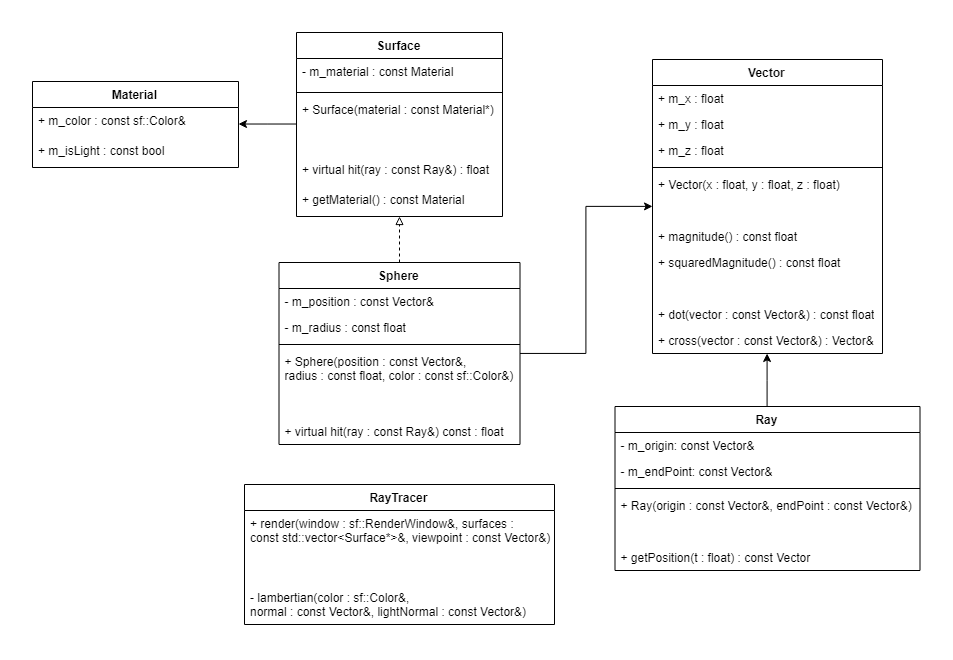
## 10:45-11:15 Fill out yesterday's and today’s work log

## 11:15-0:00 Workshop C++ modularity

* It’s a workshop about how to make a more modular program.
* A more modular program means to hide implementation and separate processes from each other. It is important for code reuse and helps with getting fewer errors. By separating logical parts of the program, you can easily focus on the task that is happening right now.

## 11:15-11:45 Fill out yesterday's and today’s work log

## 13:30-15:00 Update UML of the project

I spend a lot of time 

I decided to make a struct called material. This contains the texture that the surface has. Right now it functions only as a way to check if something is a light or not. I decided to contain this inside a struct because it is information related to each other.

I will try more shading algorithms out. It is a function right now which ray tracer will call. I wanted to make it a function pointer but I’m not sure if all shading algorithms use the same parameters. Thus I have made it it’s own function temporarily.

## 15:00-15:30 Implement design

## 15:30-17:00 Cook with Bojan workshop

It was more spectating but the food was good. I don’t think I will attend this again because I’d rather spend the time cooking on my own or programming.

## 18:00-18:30 Continue implementing design

I have implemented material and changed split the renderer function for less bloating.

## 18:50-19:15 Finish writing today’s work log

# Thursday

## 7:35-7:55 Fill out today’s work log

## 9:00-10:15 Helping Bas with gradient

## 10:15-13:00 Research how to calculate the normal of a point on a surface

* Found out about partial derivative. Have no clue what that is. So I’ll read [this](https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/partial-derivative-and-gradient-articles/a/introduction-to-partial-derivatives).
* Follow Khan academy lessons basics of the derivative.
* Learned how to interpret multivariable functions. Function f(x) = x\*x is interpreted like (x) -> (x,f(x)). Function f(x,y) = x\*x + y\*y is interpreted like (x,y,f(x,y))
* I need to refresh upon derivative notation becaus I don’t understand Leibniz notation.
* The gradient is the direction of the maximum increase. It points to the maximum increase. I don’t know how it is calculated, but when you calculate this value, you move only a tiny bit into that direction and calculate it again.
* I actually don’t know what the gradient is supposed to represent. Maybe [this](https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/gradient-and-directional-derivatives/v/gradient) will help.

## 13:00 - 13:50 Meeting with David

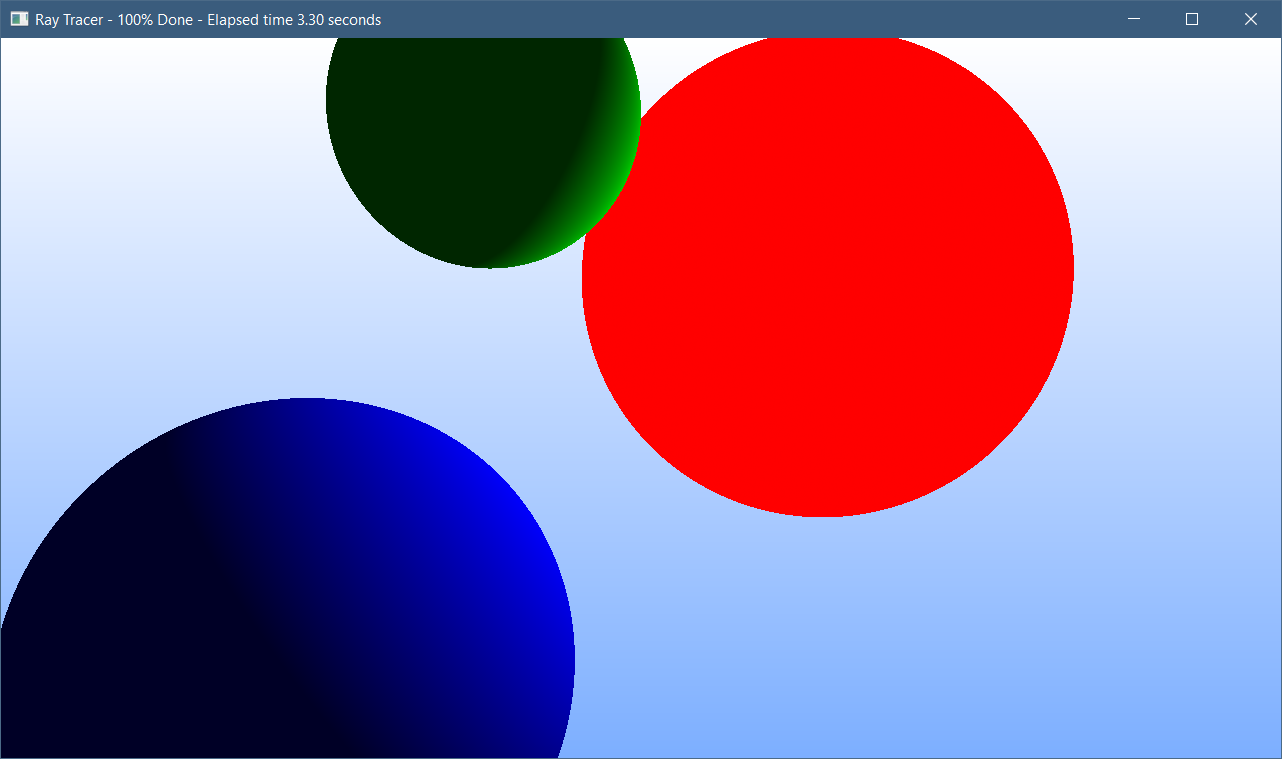
## 14:15 - 15:00 Write down feedback of David in learning log

## 15:15-15:30 Update work log to make mark huge misguesses concerning time

## 15:30-16:00 Helped Bryan but miserably failed

## 15:30-17:15 Implement Lambertian shading

It’s a bit hard to implement multiple light sources. In Lambertian shading, the color value is just the surface color times the light intensity. I will make the light intensity a normalized value to support multiple types of light. But I’m not sure how to calculate the final color.

Light values add up. The more light the brighter the object. Light is absorbed and only the color that the surface has is reflected so we need some kind of surface.

# Friday

## 12:30-12:45 Research more about Lambertian shading

The formula for Lambertian shading is:

**Pixel color** = **diffuse color** \* **light intensity** \* **max(**0,**dot(normal, unit to light))**

I’m not sure what diffuse color and light intensity represent. I think that the diffuse color is the color of the surface and light intensity is a value between 0 and 1. But that would not support lights of a different color, It would assume that every light is white.

I will first implement that method, after that, I will create the work log for today.