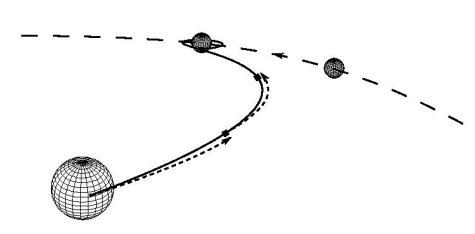
Oh boy. I can tell you one thing; sending out satellites is a bitch and a half to do. I would say that it’s almost like rocket science, or that it’s pretty much Greek to me, but I use Greek letters for my maths and it *is* pretty much rocket science, so it pretty much is that.

Trying to get a read from where the satellites are, and filling them up with fuel is a whole other issue entirely.

The thing is, up until recently, I had only simulated the rockets I shot up, but I needed to do things more accurately. I wanted to get into stable (hehe, horsies) orbit, and just having an inaccurate simulation was not enough. So I started sending up lots of satellites, and found that I was a bit off with my simulations. Who’d ‘ave thunk it. Luckily, I had this amazing thing called a rocket! Amazing what science can do nowadays; release a continuous explosion of particles to push myself in one direction. Anyways, I needed to find a way to get my satellites where I wanted them to be.

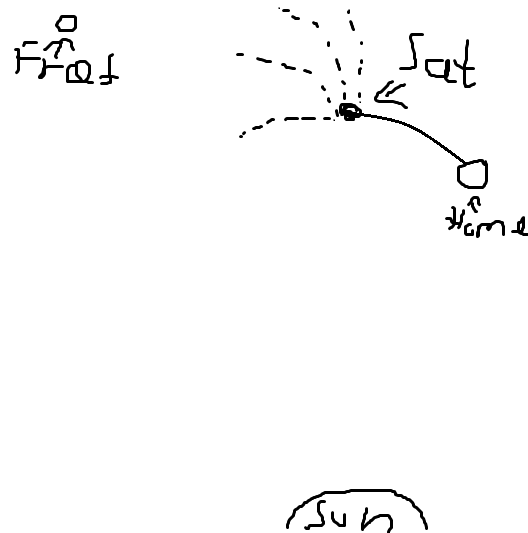
First I needed to find out where my satellite was at any point in time during the launch, which I had conveniently already gotten earlier (remember what I wrote last time? Kinda important, that bit)

Secondly I needed a way to simulate a new trajectory from a point during the flight, find out which trajectory would get me closest to the planet Fræfdal, and use my huge-ass rockets to get me into that trajectory.   
I already had a trajectory simulation algorithm that I used earlier to find the original trajectory, but that revolved around (hehe, get it?) finding out at which point in time a launch at a specific speed and a specific launch angle would get me the closest to Fræfdal, and I made it that way because it was just easy to do. I couldn’t really do that anymore since I needed to boost at specific points in time to get to the planet



I couldn’t be arsed to draw a picture myself today (I was busy being on the shitter because of this weird pickle I ate for breakfast), so you’re getting the store-bought one  
The thick line shows the actual trajectory, while the dotted lines coming out of that thick line show simulated trajectory started at various points along the actual trajectory. The bigger dotted line is just the target planets orbit

So in order to find the best direction to boost, I made some new software that took a boost of a specific speed in a specific direction and checked how close that would get to Fræfdal. I then rotated that by a specific angle, and checked how close it got again. I continued this for a while and found the angle that would get me the closest. Seems pretty simple to make. Turns out, this was a total fucking bitch and three halves (see, an increase of a whole two halves here) for me to do. There were so many things I fucked up, and oh boi, let me fucking tell you.



A very poor description of the method of simulation. It's bad, I know, I blame the pickle.   
The stippled lines are the possible simulations (just imagine they extend to Fræfdal (marked as Fræf), and that they follow gravity to some degree), and the full line is the actual trajectory up until that point.

I started out with using my trusty simulation software that I made before writing my 3rd post to simulate satellites going through space, to make the simulations (as it’s made to do), and found out I hadn’t made it to be able to change the direction that I sent the satellite in, so I had to rework that function, so I did. I then found out that the function I used to rotate the launching direction was faulty, because it didn’t work the way I wanted to, so I had to exchange that for another one.  
Next I found out that it just plain didn’t work, it just gave me an initial direction, and the minimum distance between the satellite and the planet didn’t decrease with subsequent directions, and I figured that I was just hella lucky, and hit the best I could at once (Yes, that’s stupidly unlikely, but I was tired and it was late)  
My thought process went something along the lines of “if I had had the wrong angle, rotating would have gotten me closer to the best angle, and since it didn’t, I had the optimal angle immediately”  
It turned out I was rotating in the wrong direction in that specific case.  
 I had to work in something that would reverse the rotation direction if rotating in one direction did not bring me closer to the planet. I did, and it took a long time to figure out that it was what I had to do and to do it at all.  
After some unnecessary hours of frustration later, I found a bug where I measured the distance to the wrong planet (planet 0,where I am now, instead of planet 3, Fræfdal), so I had to fix that.  
And for every time I tried to fix it, I had to launch another satellite and manually read out the data and check it against the data I was expecting with my simulations, which took a lot of time, so every single mistake I made took an eternity to fix and I am positively miffed >:(!

After the program was working, I started the tedious process of sending out satellites and reading out where they arrived, which grew more and more tedious as I got closer to Fræfdal, as I needed to be more accurate the closer I got, so I needed to do more checks, as well as compensate for the previously mentioned times when I naively thought that because the accuracy didn’t increase when I rotated the boost direction (when I turned it the wrong way, because fuck me amirite?), I had gotten it perfecto on the first try.

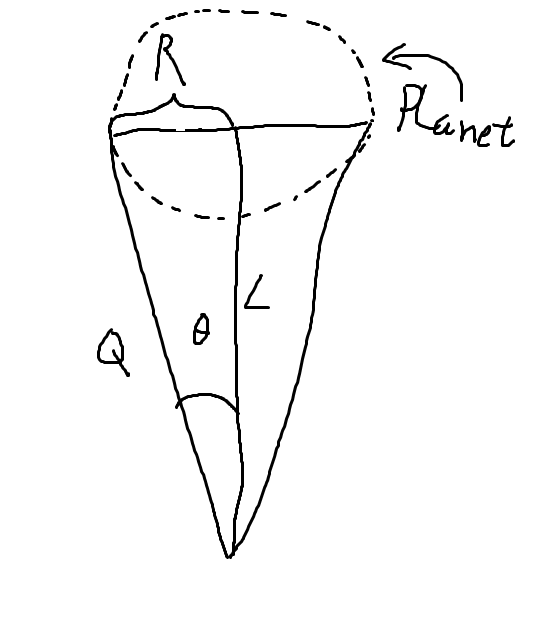
So basically, trying to get my satellites to Fræfdal was one big fucking fuckfest. At least it’s done now, as I managed to get satellites into orbit (at fucking last).



A picture from orbit, sorry that I didn't get any video, that software kind of stopped working.  
Also don’t mind the simulation-esque way the picture looks, it’s real, you can trust me

As a fun side-note, something I was working on as that pickle was working its way through my system and seemingly punching every internal organ I own and some I don’t own (long story, don’t ask), if you know the resolution of your camera, its field of view and the size of a planet, you can figure out how close to the planet you have to be in order for it to take up more than a single pixel. All you need is some simple trigonometry, and we all love triangles, right? Better than pickles, anyway... Why did I even eat that, I don’t even like pickles!

To start off, I found out many how angles a pixel took up. I simply divided the cameras field of view by the amount of pixels it has, and because I did it in generalized form for square video feeds, a camera with a field of view of F by F degrees, and P by P pixels gives an angular pixel size F/P.



Thank you thank you, I know I make great triangle drawings

This is just a general triangle, from which we can grab some trigonometric equations. Specifically, R is the radius of the planet, L is the distance to the centre from the satellite and Q is the distance to one of the sides… from the satellite as well. I love repeating the same three words before even finishing a line.

cos(Θ) = L/Q  
 sin(Θ) = R/Q

which gives

Lsin(Θ) = Rcos(Θ)

Since we can make the approximation sin(Θ) = tan(Θ) = Θ for small Θ’s:

L = R/tan(Θ) = R/Θ

For a planet to take up more than a pixel, the angle (as in the area described using angles) it takes up on the camera, Θ, has to be bigger than the angular size of a pixel, F/P (I write bigger than or equal to, because tbh if the size is accurately shown by a single pixel, that’s good enough. And to the people who say Θ has to be bigger than 2F/P to be shown on two pixels, which is more than one pixel, I say shut it naysayer, here we always round up! Shoot for the stars!). Therefore we write:

Θ ≥ F/P

Which means that

1/Θ ≤ P/F (when inversing inequations, flip inequality)  
 R/Θ ≤ RP/F

L ≤ RP/F

So if you ever want to know, for whatever reason, how close to a camera you have to be to be more than a pixel, now you can :)!

So, now I’m going to treat you to a detailed description of how I simulated and boosted my satellites away!

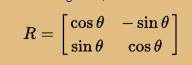
The most basic principle is using the simulation software I made and described in my 3rd post. After editing it so that I could change the starting position and velocity at will, as well as provide a boost in any direction I want, which I needed, it was ready to use!  
I already had a simulation that, according to itself, brought me incredibly close to the planet, but it is only a simulation, and is probably erroneous, so I had to send up actual satellites, as I mentioned previously. What I did was let the satellite fly for a bit, until it had gotten far enough from what I had simulated. Far enough means whenever I said “ahh, that’s probably far enough. I mean, if I let it go any further, I would be as off as Johnson was with his throwing. Fuck, Johnson sucks at throwing”. I said that phrase specifically, every time. Totally.   
Anyways, when it’d gotten far enough, I recorded where it was and how fast it was moving, and started a new simulation from that point, as described beautifully by my drawing earlier. I first gave it a boost towards where the position that Fræfdal had at that time, ran the simulation and checked how close it got, then I started a new simulation with a new boost turned a tiny bit towards where Fræfdal would be at the time I hit it in the original simulation. I repeated this while the simulations got me closer and closer, and cut it off when it started going farther away from the planet. I then sent up a new satellite with instructions to give a boost at the time I did simulations for the previous satellite, let it go for a while, and then started making new simulations and predictions.

I continued this process for a while until somehow the first direction I tried to boost in was the most accurate, and turning it only made it less so. “lucky me, I thought”, with suspicion growing in my mind. Of course, I didn’t do anything about it for like the next 3 or something times I gave new boosts, before it realised why things were going *too well*; “oh wait, maybe I have to turn it the other way”. I did. I’m dumb. I changed to simulation code so that if the satellite only got progressively further away after the first simulation, I would have it go back to the first direction it simulated at and start turning in the opposite direction. This worked. I’m dumb.

I continued this until I was close enough to go into orbit, where I first performed a boost to give the satellite the same velocity as Fræfdal around the star, then I gave it another boost equal to the velocity it would have had if it had orbited the planet, which would make it orbit the planet. Orbital speed can be found where the centripetal force is equal to the gravitational force, which can be simplified to:

Where G is the gravitational constant, R is the radius from the planet and M is the mass of the planet. I just gave the satellite I had sent out (I don’t even know how many I had sent out earlier, just that it was a lot) that speed as a boost around the planet, and then it got into orbit. An elliptical, really ugly orbit, but an orbit nonetheless!

In order to find the direction of the first boost, I took the x,y coordinates of Fræfdal and subtracted the x,y coordinates of my satellite, which gave me a vector pointing from the satellite towards the planet. To rotate that vector, I put it through a rotation matrix which, as you might guess, rotates the vector.



Technically, I got 0.003 Astronomical Units (look it up) away from the planet, but because I now have elliptical orbits, and I can’t be arsed to fix that, that distance varies. 0.003 sounds like a small number, but it isn’t when we deal with AU’s, trust me.

Technically, these travels take like half a year to complete, but I have a time-warping device, so I don’t have to wait that long (hah, suckit physics, I don’t have to abide by you, *I’m fictional, hah!*)

Notes for myself while writing, don’t add this in, dumdum:

Something something vector\_planet – vector\_satellite = direction

Something something rotation\_matrix \* direction = New\_Direction

Things to detail:

* Description of angular simulation (to find planet from spaceflight)
* Rotation\_angle\_2d
* Final orbital boost

Criteria:

* How well is the task solved (previous 90% )
* Explain the actual problem (previous 100% )
* Explain thought process (previous 80% )
* How the blog is built, and results (previous 70% )
* FIGURES/ILLUSTRATIONS (previous 0% )

REMEMBER TO MAKE SURE THETA STAYS AND DOESN’T GO BYEBYE WHEN TRANSFERRING TO BROWSER

1) i hvor stor grad har oppgaven blitt løst: Alle delene ser ut til å være nevnt, men ingen test av hastighet og retning. 90%  
  
2) gjengi fysisk problemstilling (og dermed vise fysisk forståelse): Bra, 100%  
  
3) fortelle hvordan du tenkte for å komme frem til løsning: Virker veldig naturlig forklart, også hvilke feiltrinn du gjorde etc. men du kunne vist mer forståelse her med noen figurer, det ville gjort det mye klarere hvordan du tenker. 80%  
  
4) hvordan er bloggposten skrevet og hvordan er resultatene forklart og tolket: Bra oppdeling med en liten introduksjon som forklarer overfladisk de tre delproblemene, deretter mer i dybden. Mangler litt resultater og tester fra retning- og hastighetsdelen. I tillegg så hadde jo noen figurer hjulpet veldig, dette er jo egentlig en veldig visuell del, du kunne f.eks. vist bildet som du skal plassere på himmelkulen samt projeksjonene av himmelkulen og hvordan de passer sammen samt mange andre ting som hadde vœrt naturlig å vise frem her. 70%  
  
5) figurer/illustrasjoner: Fraværende! (Jeg syns ikke bilder av formler teller) 0%