# Monday

## 8:55-10:55 Build multiple vessels

2

## 10:55-11:55 Fill out learning log and work log

1

3

## 13:15-14:00 Fix transparency issues

0:45

3:45

## 14:00-15:45 Add SOI

1:45

5:30

## 17:00-18:30 Set target planet for vessel

1:30

7:00

# Tuesday

## 11:15-12:30, 17:00-18:00,19:15-20:15 Make vessels move

1:15

1:00

1:00

3:15

## 20:15-22:15 PDP

# Wednesday

## 13:20-17:20,18:10-21:25 Make vessels move

4:00

3:15

# Thursday

## 7:50-8:30, 9:10-12:00, 13:15-17:10 Research how to create a map

I could just take the center position of every planet and make a texture from that list of coordinates. Create it in memory and apply it to some texture. This makes the whole map 1 draw call, and since I can draw overlays on it if needed. That seems like the best way.

I can just create a map of bytes consisting of only alpha values, then in a fragment shader, I will calculate the appropriate color values. All the scaling will be handled for me by the fragment shader.

One of the problems is deciding how large the texture will be. Assuming 8192 planets, I can divide this by two (which doesn’t really become a nice number) but 90 \* 90 becomes 8192. Of course, planets are not 1 pixel and they have space in between. Almost every planet is located at least 10 units from each other. If I were to represent planets as pixels (this seems easiest but I may have to change that. The texture would become at least 90 \* 10 \* scale, taking the scale \* 3 because a planet the size of 1 pixel seems too small, the dimensions would be around 3000 pixels in width and height.

[The Pi supports textures up to to 2048 something](https://www.raspberrypi.org/forums/viewtopic.php?t=1693). Assuming that means in pixels, my texture would be too large for this.

Another way would be to create a bitmap containing all the information and updating the vbo accordingly. Using an algorithm like [this](https://en.wikipedia.org/wiki/Marching_squares), I would be able to create a map based on the planets positions and state. I could update this map every so often and update the VBOs. Updating the VBOs seems cheaper than a whole texture.

Having read this [part](https://stackoverflow.com/questions/7537013/opengl-is-it-better-to-batch-draw-or-to-have-static-vbos), it appears that I should make the map one huge vbo.

# 

# Friday

## 10:50-14:00,15:10-16:00 Add map to the project

3:10

0:50

2 4:00

When binding a new vbo, you have to specify the attribute pointers again

## 17:25-18:00 Research pathfinding algo

<https://en.wikipedia.org/wiki/Pathfinding>

I will use a\*

## 18:00-21:10 Add line renderer

3:00