Worklog summary week 5

Angelo Anthony Rettob  
181494, IGAD Programmer

Table of Contents

[**Monday**](#_3y5dxel9s105) **3**

[Schedule](#_9ary9asaem4l) 3

[Events](#_ue509jhvbx48) 3

[Fixing the physics engine](#_99hbvotsvir) 3

[**Tuesday**](#_scsc92l491er) **5**

[Schedule](#_o87c7rgc904) 5

[Should I restart or not?](#_mh8rsqopsyqv) 5

[Rewriting some classes](#_sf0k4guh9qcs) 5

[Events](#_bx2dlr8r1u8q) 6

[**Wednesday**](#_wyif0cqvidyu) **7**

[Schedule](#_454kx26109y8) 7

[Events](#_vny33qep6gld) 7

[**Thursday**](#_jaaiz49otk9q) **8**

[Schedule](#_kbbv2ohhfidw) 8

[Mapping texture based on surrounding tiles](#_317hlgdmky5o) 8

[Shadow map](#_6kftmhszz4h1) 8

[Monster AI, states and pathfinding](#_opfjkdt9boci) 9

[Events](#_uhim8boblxoe) 9

[**Friday**](#_6leoq1ock4dq) **11**

[Schedule](#_1mq19um3beqi) 11

[Events](#_dztfp1lq5ac9) 11

[**Saturday**](#_7q3p7mu4cl19) **12**

[Schedule](#_3iyqzwajb588) 12

[Events](#_7fhmtvu2e0bd) 12

[**Sunday**](#_m5vwfcpjupci) **13**

[Schedule](#_enz5x66vd1v9) 13

[Events](#_w2pu2j2cwyy3) 13

# Monday

## Schedule

Today I’ll be working on animations. The animations are ordered by rotation. Because of that, we have to multiply the animations by 8.

I’ll also be implementing the consumable class. These are objects that can be picked up by the player. With this, every object should be displayable.

Hopefully, I can also implement some projectiles.

## Events

Programming is going well. A lot of code has been exchanged. Nothing special is going on but I am not really progressing my own project. I also had to create all the documents for this sprint and finish up the work log from yesterday.

After quite a bit of struggling with UV coordinates, I have found the solution for my sprite sheet.

Abhishek actually interviewed my decisions without telling me. I kind got caught off guard but could explain my decisions. Adriaan also helped me so I stayed a bit silent. That is quite the bummer because I should easily be able to explain my decisions. I’ve always been a bit unsure. I should work on my confidence in displaying my code and explaining my decisions instead of thinking that the teachers are always right.

Also, Abhishek is ruthless.

I’m getting the hang of UV coordinates. We just have to normalize the values. I am getting confident in working with different coordinates system.

### Fixing the physics engine

The teachers keep telling us to first fix one thing and only then move on to the next part. I have some weird behavior inside of the physics manager. Whenever an object collides with a tile edge, it bounces back. But it should not move at all actually.

Observing the frames one by one showed me even weirder behavior. The object just gets stuck inside the edge for one frame. It really isn’t supposed to do that.

The way that I calculate the position is explained [here](https://docs.google.com/document/d/1Dvq9wEecv_puZIUn0H3HDMKf63EUWNmJllBPZSACXlE/edit#heading=h.bpeau7re1m5z). To summarize, I calculate the distance between the x and y-axis and move the object back with the shortest one. I calculate both independent so that we can walk parallel to tiles.

One thing that I thought is that when we move inside an edge, the distances should be the same. So we just have to move it back to its beginning position. This didn’t work though.

I decided to delay this because it is halting my progress quite a bit.

# 

# Tuesday

## Schedule

### Should I restart or not?

The project is progressing quite nicely, but there is a bit of a struggle that I’m having with deciding if I should continue my project this way. Why am I thinking about this so far in the project? I’ll list a few things about the project and why I should restructure it.

* I don’t feel like my object classes are making a lot of sense, I have a consumable class, enemy class, and tile class. I order them quite odd. For example, where should the exit tile go? Not in the tile class because tiles only handle collision and not in the consumable class because consumables are destroyed upon collision with players. I feel like I have to reconstruct the way I structure my objects.
* Right now I’m managing tiles independently from each other. I should probably write some kind of function that calculates the appropriate tile depending on the surrounding tiles. The reason why this is quite an important thing is that if you [look at this](https://youtu.be/YqVvvARVq3w?t=22m10s), you can see that the wall adjust itself after that wall has been removed. I assumed that walls are static and never change. Walls can also disappear, so I should really start fixing these things. Which brings me to the following thing
* The way Tiled and my game communicate. I changed the resources so that Tiled can use them and I can implement them easily, but because of this, the flexibility of Tiled has been lost a bit. You can’t use your own layers because I use them to separate the way objects are loaded into the map.

So alright, why don’t you just fix these? It’s because I would have to change the way the Map class manages objects. I would have to change every object to support the new layout Tiled will supply. I would first have to make it all up.

It’s a bit my fault because I noticed different ways to do things while developing. Because of this, my initial plan got a bit mixed up. Experimenting isn’t a bad thing, but next time, I have to be sure that I keep everything up to date/ in sync with the rest.

So my solution is not to change all of the old code, but to adapt to it. I’ll refactor some pieces a bit, but I won’t change the whole backbone of the project just to see if it will be better or not. I’m glad that I see at least am starting to see things that I could have done better.

### Rewriting some classes

I will rewrite some classes in preparation for the other objects.

* The tiled class will import just one layer. All work will be done on one layer.
* I will change the structure of object classes.
* I will rewrite the way map class works a bit to support tiles that can be destroyed.

## Events

I’ve made a new resource file.

There really isn’t much to say about today. I only rewrote a bunch of code. To explain why I did this, it is because I thought that it was a smart idea to map the enumerator value directly to the UV coordinates. This should make it easy because we don’t have to fill in any texture coordinates. The thing is that my enumerators became very big. For me, it became very unmanageable. So I decided to split those up into their own files.

I build the whole texture layout around that principle, so I had to change the layout of the images too. This really isn’t a bad thing, because it allows me to easily add new textures without having to worry about the enumeration layout. 

This was a very long day, I knew that from the start. Either way, the animation class is very clear now in my opinion. This should speed up development in the long run. I couldn't finish everything that I had planned, but I did talk with a few students about the rounding issues of the textures.

Adriaan found an article on stack overflow which said that us should index the tile at 0,0,31,31 instead of 0,0,32,32. This is because you can see the range as pixels, and pixel 0 also gets counted with it. It can be quite confusing, because using ranges with iterators includes up to the last value, but not including it. For some reason OpenGL does.



If we look at this picture, the result of maxV should be 0.266, but instead it is 0.636. This is not good. This means that there is something wrong with our calculation.

# 

# Wednesday

## Schedule

Today I’ll complete the implementation of objects.

I’ll also be writing a function that calculates the correct texture depending on the surrounding tiles

## Events

I haven’t worked on the function that calculates the correct texture depending on the surrounding tiles.

The debugging workshop was really useful. I now know how to find memory leaks using Visual Studio. I also know quite a few tools that can be used to do static validation of code. This will certainly serve helpful one day.

# Thursday

## Schedule

Today I’ll be implementing new things again. First of all. We need to update tiled depending on their surrounding tiles. The reason why this is important is that walls in Gauntlet can be broken. If this happens, tiles adjust to their surroundings.

To support this, the tiles should map their texture themselves. It is not only tiles that do this, but doors also do this, with the added behavior of course.

We will also be making a shadow map, something else I haven’t ever done before. I’m thinking of two ways to do this. Either manage them independently as standalone objects or make them part of the tile texture and update that one according to the wall that’s next to it.

We will also be implementing pathfinding. I’ve never done AI that needs to keep track of its surroundings, but the physics manager should be responsible for this. Which means that implementing the AI is as simple as giving it a movement function.

I’m also gonna start implementing the entity stats, which doesn’t really affect behavior. It just modifies it a bit, so this should be easy. I only need to know what stat values they use.

### Mapping texture based on surrounding tiles

We already have a function that returns the surrounding tiles of a tile. So we can use that to calculate the position. I’m actually thinking of mapping it to a 2d array boolean array and make if statements that calculate which position is to be used. This sounds like the most straightforward way and because it isn’t a function that gets called every frame, which means that it’s not expensive.

Another problem that I’m facing is that doors also need this functionality. I don’t want to make the function twice, so the most logical way seems to make both inherit from a Surround Tile class of some sorts. All that this does is add a function that updates these tiles depending on their surrounding tiles.

Or instead of making a class for it, we can make a function that takes a map and the sprite id and works its magic from there. This spares us the additional inheritance. Though inheritance (most of the time) means that the object is a type of, and in our case doors and tiles are types of surrounding tiles.

### Shadow map

Now that I think about it, tiles also need to interact with surrounding tiles or at least the tiles that are on top of them. How will we implement this behavior? One thing is for sure, only walls affect floor tile shadows. Also, shadows from floor tiles and walls overlap, increasing the intensity of the shadow.

The tile only needs to know on what kind of wall it sits. The tile is dependent upon the wall. So instead of making the tile responsible for what it should be, we do it in the wall logic. Wait, that sounds totally against OOP. But should I really complicate my solution just to make it OOP?

I actually think I should complicate my solution. Because doing OOP now will only increase my skills at applying it next time.

So how will the shadow map be implemented? I’m thinking of updating the surrounding tiles received from the walls. Only tiles around walls have to be updated. If this really is against OOP too much (because why should a wall be responsible for a tile), I can just update every tile in the map everytime a wall gets changed.

It follows the same principle as the wall. It takes a 2d array of booleans. Where there are walls, the elements will be true. Using this, it will calculate the appropriate UV coordinates.

### Monster AI, states and pathfinding

I have never implemented AI pathfinding where the AI needs to be aware of its surroundings. If I’m being honest, I’ve never done one of the things I’m gonna do today before.

Fortunately, the AI of Gauntlet is beyond simple, conceptually wise in my opinion. From my observations, the pathfinding of Gauntlet doesn’t keep track of its surroundings. It just calculates the shortest distance to the player. If there is a wall in between, it will just pause there.

There is one thing that I’m unsure about. It seems that objects move according to a grid. Brian told us it’s so that they don’t overlap, but my physics engine makes sure that doesn’t happen. Should I implement pathfinding according to the grid then? I don't mind doing so, but I will do that after I’ve finished the initial implementation

Ooh and a short bit about **states**. Object have but two states, Moving and attacking(shooting/ throwing some kind of projectile). Some objects can’t even attack something so they will only move (like the ghost). We will, of course, be working on the movement state today.

## Events

I'm deciding if I should use a function pointer or inheritance to implement this function. The wall and doors need the same logic, but the tile needs different logic. Both need to be updated when the function is called. How should I implement this?

I’m probably gonna add the 3 new class types.

Converting the classes took quite a while, but I can now define the function without copy pasting it.

After researching a bit on the internet I found out about bitmasking. I’m trying to apply it myself now.

It worked! And it felt quite amazing.

## 

# Friday

## Schedule

## Events

# Saturday

## Schedule

## Events

# Sunday

## Schedule

## Events