**Development diary**

Rapid prototype

24th January

Met with group to discuss what should be included in the rapid prototype. We settled on including a small part of everything that would appear in the finale game i.e. ropes ladders enemies, a player, jump pads, convey belts, low gravity. We also agreed we would focus more on producing them something that contained these elements and not so much on creating a polished finale piece of work.

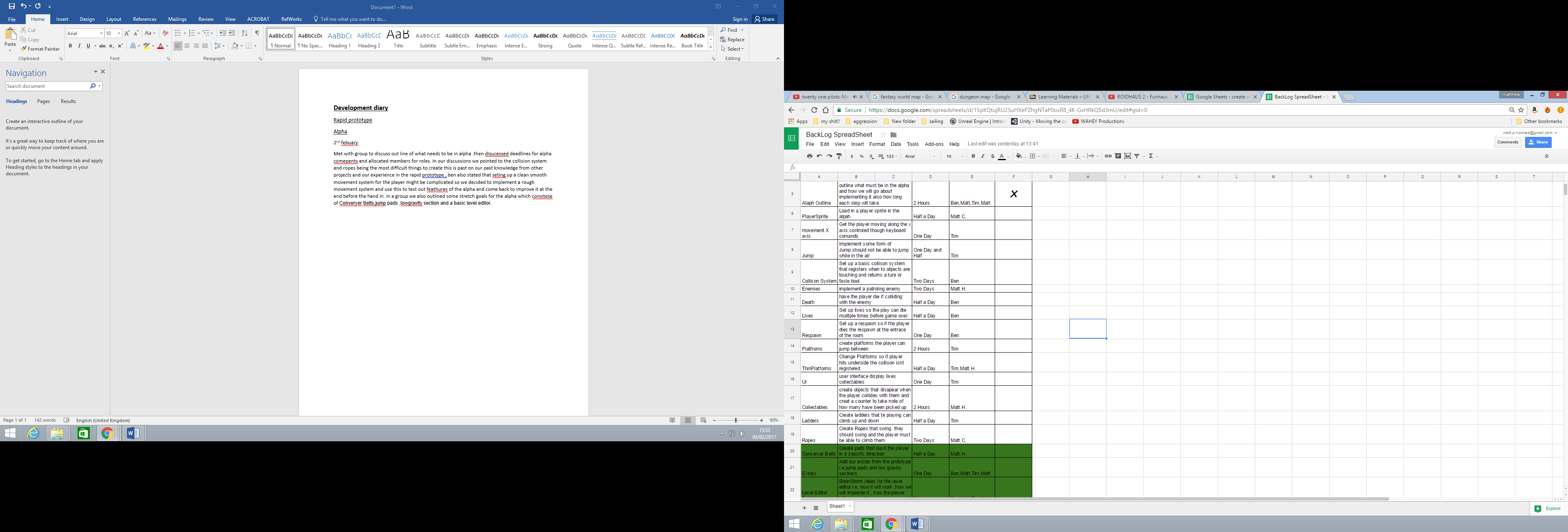
27th January

Most elements of the prototype are finished now only thing left to include is a rope that the player can swing on and stairs. Matt C and Tim have agreed to work on the rope swing together and ben has agreed to work on the stairs with my help as they should work in a similar way to the thin platforms I created.

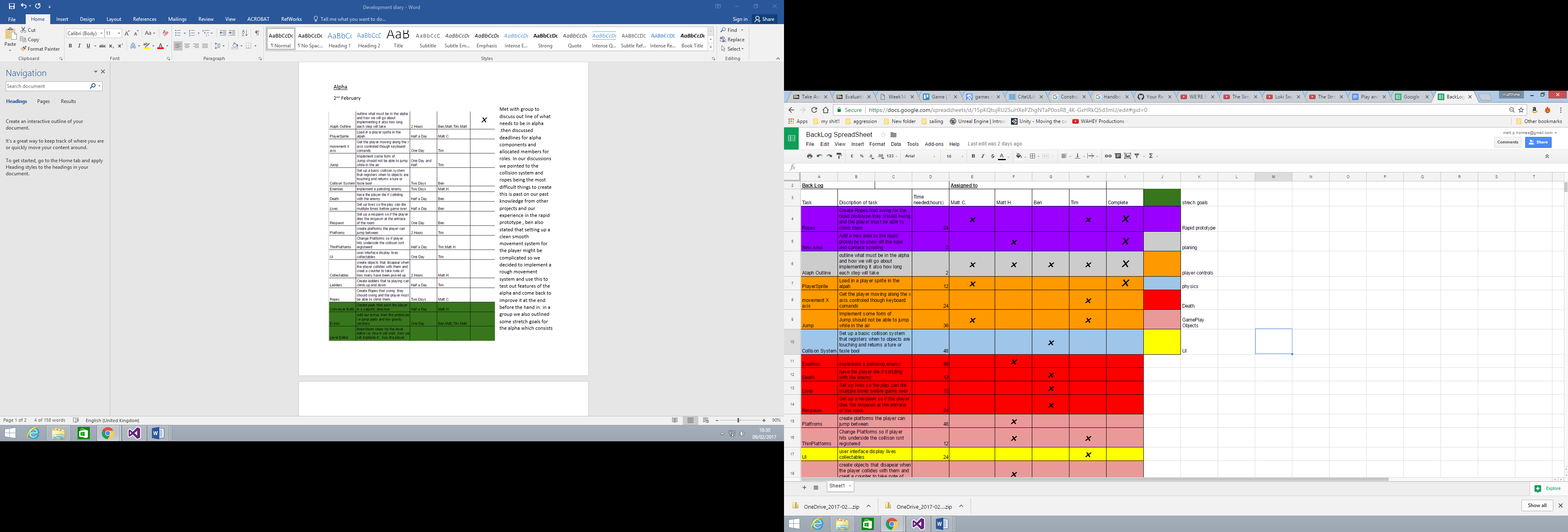
31st January

Today we handed in our rapid prototype and had a debrief of the project and set up times to meet to discuss the alpha. We settled on the second of February.

Alpha

2nd February

Met with group to discuss out line of what needs to be in alpha .then discussed deadlines for alpha components and allocated members for roles. In our discussions we pointed to the collision system and ropes being the most difficult things to create this is based on our past knowledge from other projects and our experience in the rapid prototype , ben also stated that setting up a clean smooth movement system for the player might be complicated so we decided to implement a rough movement system and use this to test out features of the alpha and come back to improve it at the end before the hand in. in a group we also outlined some stretch goals for the alpha which consists of Conveyer Belts, jump pads ,low gravity section and a basic level editor.

7th February

Today we redesigned the back log to make it easier to read and clearer who is doing each task. We then went about starting the alpha Mattock. Implemented a player class that inherited from the already existing 2d image class that displayed our sprite to the screen .Tim then took this new player class and introduced movement along the x axis through key board commands. Using Tim and Matts work as a base I implemented a new enemy class that also inherited from 2d image class. This new class displayed any enemy and moved the enemy right until it reached a pre-defined position at which point it would move left until it reached its starting position and then it would move right again(this is the same as the enemy movement as the rapid prototype). This movement currently doesn’t work as the enemy is not registering the positions. Once this is successfully implemented we can test the collision system Ben is implementing.

14th February

Today while Tim worked on the Games UI and matt implemented a basic jumping system, I cleaned up the Enemy class and finished the Enemy movement for the alpha so that now its start position and farthest left position are passed through to the class and then used to find the farthest right and left positions it has to move to. I then used this new system to give similar functionality to Vertical movement for the enemy. I then added a collectable class and give functionality to the player to keep track of how many collectables they are carrying and integrated this to Tim’s new UI. While I was doing this ben started work on the collision system and raised a concern that the system may taking longer than planned to create so until that is finished we will have to hold off on creating certain object i.e. platforms and try to implement everything without collision and add collision later to the project

21st February

Today ben committed a basic collision system that checks the distance between the centre points of two objects to register a collision. This meant I could finish the collectable and enemy functionality that I had setup a week previous so now if the player hits a collectable the collectable count increases and if they hit a player the player character no longer renders and their functionality stops. I also set up a lives system out line as it was similar to the collectables so now if a player collides with an enemy it checks how many lives the player has left and if it’s above 0 it minuses one from it.

22nd February

Started to work with platforms. Created a new platform class that contains a function that creates an upward force on the player this will be called if the player and platform collide. On collision with the platforms centre the player is repelled from the platform and doesn’t not fall however when drops moved away from the centre the player straight through. Matt has informed me this is due to the collision system we are currently using as it registers collisions between objects by looking at the distance between the two objects centres matt suggested bounding boxes would be a better way to do collision and we agreed to discuss this with ben the next time the team meets up. I also decided to create the platforms similar to how Tim had created the ladders by passing a file name position and number of platform tiles in to a platform class that then creates that many platform tiles in a row. This is a better system as it gives us more control over the length of platforms and can be fed into our level editor at a later stage.

23rd February

I implemented the new platform system however this It does not register the collisions between the platform tiles and the player I am going to speak to Tim about how the collision and ladder system work and see if I can fix the bug on my own.

1st March

Worked with matt to fix problem with platform collision. We found that the problem was that the current collision system only registers collision between the player and the centre of an object if they are a certain distances from each other. This means when the player moved across the platform the moved out of the collision distance and fell. To fix this matt said he would go change the collision system so it didn’t work off of a distances to the origin of the object.

10th April

The group met up today to discuss what we had left to do and what we needed to focus on now that we had handed in the alpha. We decided that our games physics system needed an overhaul and Tim offered to do this as he had work from his simulated world’s project that would prove helpful. We also delegated people for the systems we had left out of the alpha; matt is going to work on the rope objects , ben is going to implement a state system including menu states pause states restart state and an end game state, I am going to work on implementing a basic level loader system.

12th April

Today I took out all the object creation in the game cpp and created a function that loaded a new rom name in from a file using an f stream. This worked but I decided the file format that’s being loaded in was very clean ad might be confusing to someone who wasn’t involved in the level loaders production so I decide to create a level loading system that worked off of json files instead as they are more understandable to read.

16th April

Today I removed the f stream work I did and created two new classes a level class that loads in all the levels from the file and a room class that displays the current room the players in. I used a json file reader system to read in a room name and a string of symbols that will loaded in different objects from the use of a switch statement in the rooms class that runs through the screen space and creates objects in there needed positions. I used this to get a single row of platforms displaying on screen. This system does work but I don’t like it being one string it reads in from as it makes it hard for a designer to create levels.

17th April

Tested the platforms and found the collision for them wasn’t working so had to change collision system so it checked the objects created in the room class instead of them list of objects in the game class. I also added players to the room creator switch statement.

20th April

Today I added collectables to the level loader switch statement and started to think about how to load in an array of strings instead of just the one.

24th April

Today I added the last of the objects to the object factory i.e. the enemy’s and the respawned.

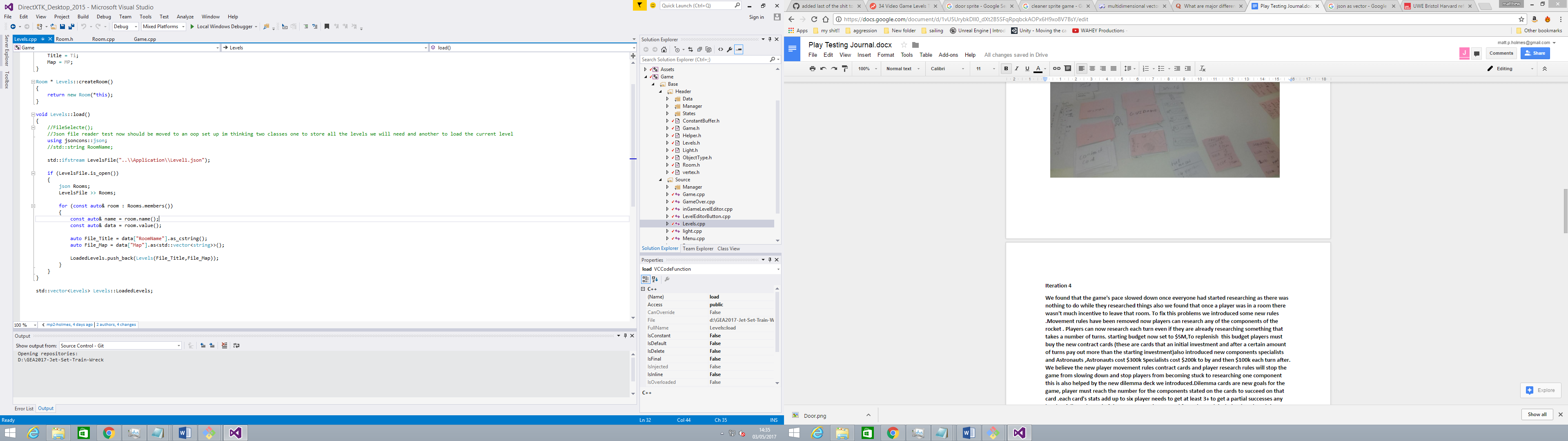
The respawned was fairly simple to implement as it is just creating another object similar to the collectable, The enemy however was much more complicated as along with a position to spawn in to the enemy also needs a point to move to this means I had to set up a goal position then create an enemy and tell it the position of the goal. I did this by create a new vector 2 in the room.h called EnemyStartPos which is created in the header file before the enemy. When the enemy is created it sets the enemy end position to the current TilePos and the enemy satart/goal position to EnemyStartPos. This system is flawed however as it means in the level json the enemy start pos has to be set up on the same row as the enemy or at least before another enemy is set up or else they will swap enemy start positions

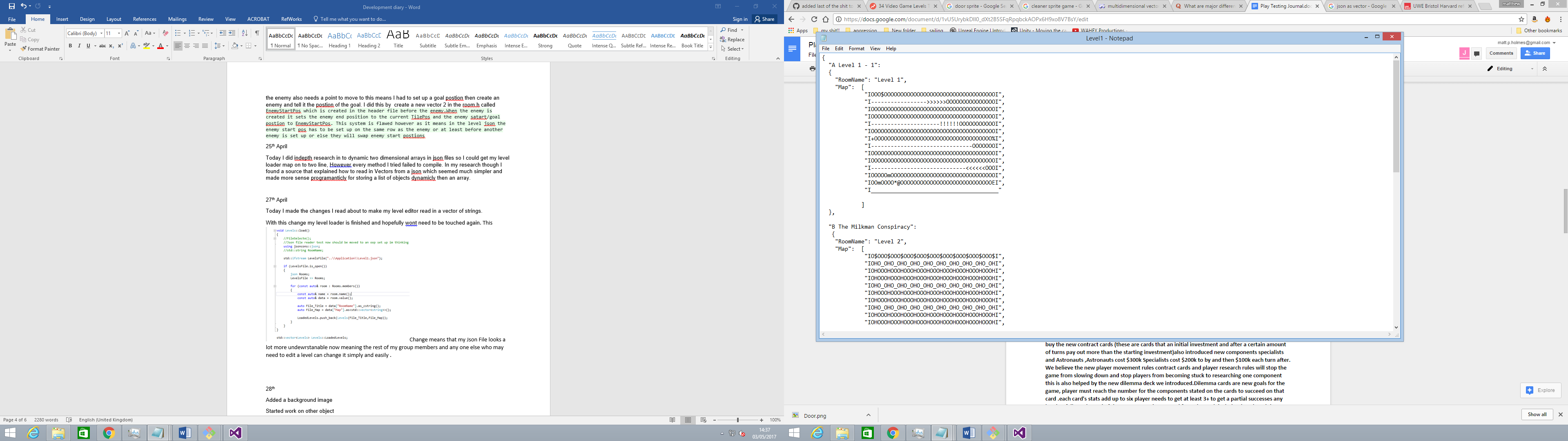
25th April

Today I did in-depth research in to dynamic two dimensional arrays in json files so I could get my level loader map on to two line, however every method I tried failed to compile. In my research though I found a source that explained how to read in Vectors from a json which seemed much simpler and made more sense programmatically for storing a list of objects dynamically then an array.

27th April

Today I made the changes I read about to make my level editor read in a vector of strings.

With this change my level loader is finished and hopefully won’t need to be touched again. This Change means that my Json File looks a lot more undewrstanable now meaning the rest of my group members and any one else who may need to edit a level can change it simply and easily .



28th

Today I added a new Background Class. The Background object is set up the same as other objects such as the collectable however it is not created in the room instead it is created in the game where its position is set to centre screen and its scale is set to high as to cover the whole gameplay environment. This Background is drawn before the room only if the player is alive. I also started work on the mother object today basing the logic off of the platform collisions laid out in Tim physics system however they weren’t working to push the player back they were only slowing the players movement.

29th

Added invisible walls to side of level but need to rotate them set rot didn’t work spoke to matt

He said it will when he has finished his tile stuff

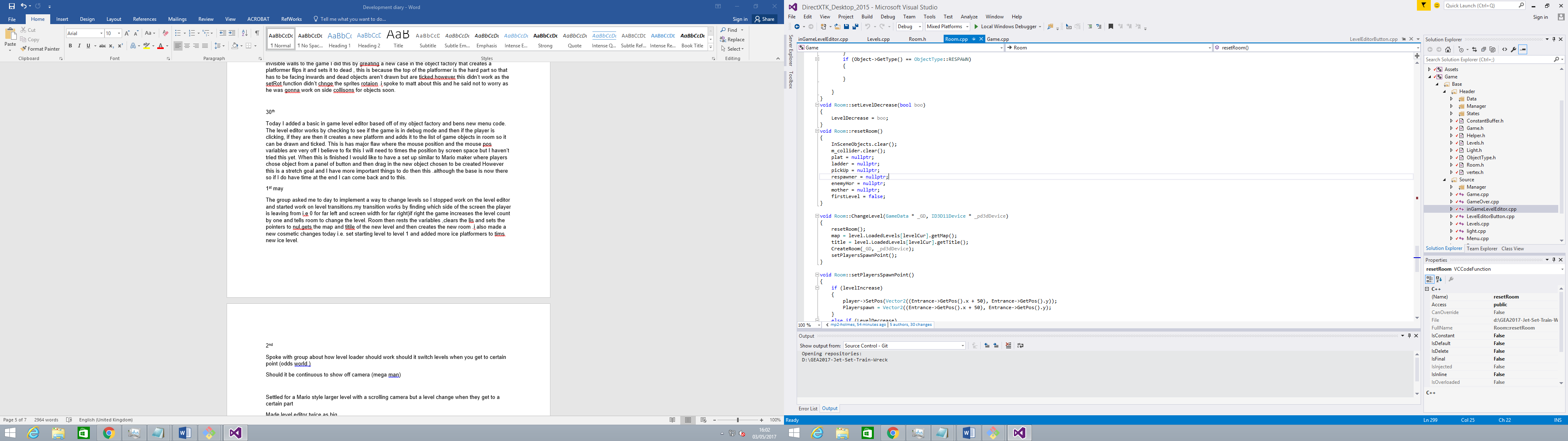
Fixed bug that stopped you from being able to get more than one collectable per level

Was setting it in the collectable class meaning when one got it they all registered as collected set it to use alive instead of collected

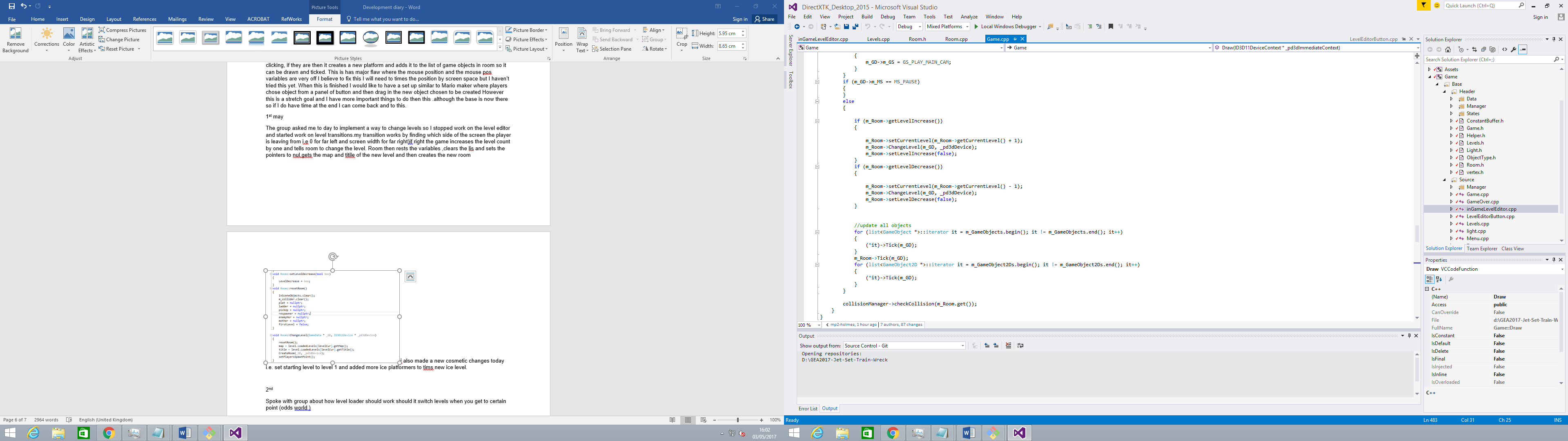
I spoke to Tim about this error he explained that his collision works by get the force of which the player is moving and applying the opposite of the is what my code was close to however I wasn’t getting the speed of the player. Tim said seeing as he didn’t have any other work to do on the spread sheet and because it was based in his physics system he could do it, so the mother has been delegated to Tim so I can get to work on bug fixes and adding to the room loader. Today I also added invisible walls to the game I did this by greeting a new case in the object factory that creates a platformer flips it and sets it to dead , this is because the top of the platformer is the hard part so that has to be facing inwards and dead objects aren’t drawn but are ticked. However this didn’t work as the setRot function didn’t change the sprites rotation .I spoke to matt about this and he said not to worry as he was going to work on side collisions for objects soon.

30th

Today I added a basic in game level editor based off of my object factory and bens new menu code. The level editor works by checking to see if the game is in debug mode and then if the player is clicking, if they are then it creates a new platform and adds it to the list of game objects in room so it can be drawn and ticked. This is has major flaw where the mouse position and the mouse pos variables are very off I believe to fix this I will need to times the position by screen space but I haven’t tried this yet. When this is finished I would like to have a set up similar to Mario maker where players chose object from a panel of button and then drag in the new object chosen to be created However this is a stretch goal and I have more important things to do then this .although the base is now there so if I do have time at the end I can come back and to this.



1st may

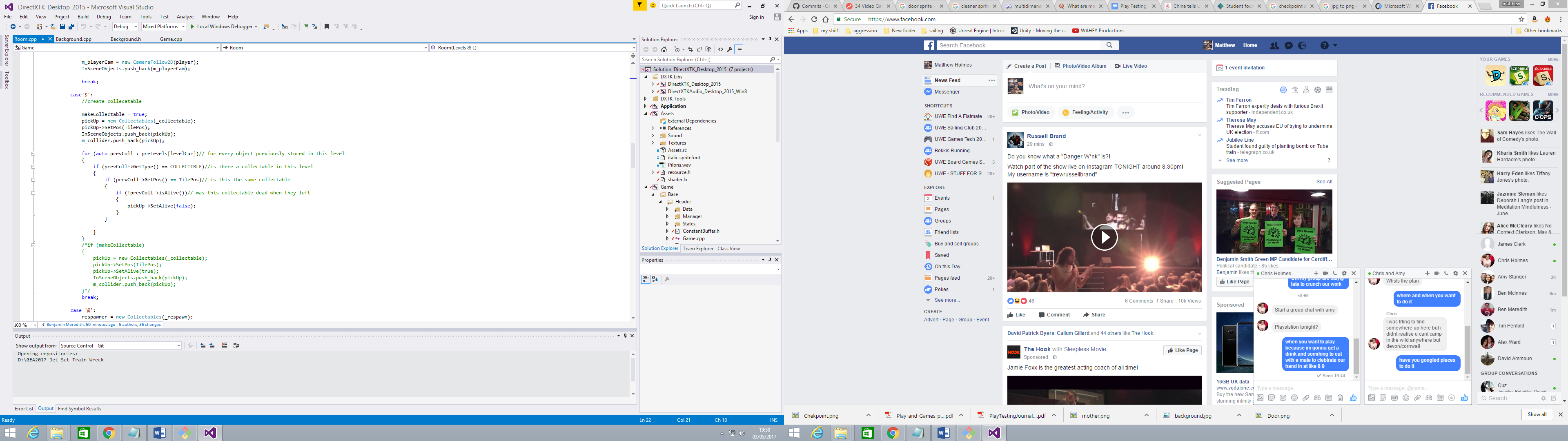
The group asked me to day to implement a way to change levels so I stopped work on the level editor and started work on level transitions.my transition works by finding which side of the screen the player is leaving from i.e. 0 for far left and screen width for far right) if right the game increases the level count by one and tells room to change the level. Room then rests the variables ,clears the list and sets the pointers to null ,gets the map and title of the new level and then creates the new room .I also made a few cosmetic changes today i.e. set starting level to level 1 and added more ice platformers to Tim’s new ice level.

2nd

I Spoke with the group about how the level changer should work should it switch levels when you get to certain point on screen (e.g. odds world) or Should it be continuous to show off camera follow functionality (e.g. mega man).we Settled for a Mario style with large levels with a scrolling camera but a level change when they get to a certain part. I then went about increase the levels size I did this by change the json file (making them two times bigger) and multiplying the variable used to get anew row in the level loader (screen width) by 2.While I was in the level editor I used matts new sprite work to get the height of the sprites and add half of that to the tilepos y instead of the magic number I was using. While play testing the game I found the player would spawn in floating this was because it was being set to grounded on creation so changed this to the opposite so it would not be floating when the players started the game. I then added new levels to the game to show off all aspects of the game however there was a bug when loading in files where level 4 was being loaded before level 1. I found this was because the json file reads data from the file alphabetically so I changed the data structure names as to get them loading in in order. I then added an end game Criteria so if the player has three collectables and touches the mother object they are taken back to the menu as a basic win set up. After this Added door objects so the player only changes level when they touch the door this works by having two door type exit and entrances if player touches the exit door the go forwards in the level vector if the entrances door the go back in the Level vector I also set it up so if the player dies and they haven’t touched a checkpoint yet they respawns at the last door they went through and set it up so if player walks in to a room it sets the spawn position to 50 units in front of the entrances (this magic number is to stop the player falling back through the door ).After I had the player smoothly changing levels I found that when I player picked up a collectable left the room and came back the collectable respawned. I tried to fix the bug by creating a vector of vectors that would store the level and then the position in that levels map of where the collectables where and push to this vector whenever I picked up a new collectable. However this failed so I changed it to be a vector of vector of collectable pointers and then adding to it when I created a new collectable object, this also failed though. Matt said he would have a look at the problem tomorrow so I commented my work out and recommitted to the repository.

3rd May

I fixed the collectable bug this morning. I did this by setting up a vector of list of game objects, this vector stores the previous levels and checks to see if the collectable has already been picked up in this room. At first this wouldn’t work if the player picked up the collectable left came back left again and came back as it would check to see if there was already a collectable in this level and if it was picked up if it was it wouldn’t create a new one which meant that if players did the above the prevlevel list would not have a collectable in it so would fail on the checks. I fixed this by setting it to create the platform and then do the same checks as before if true it would set the collectable to dead. I also spent some time today bug fixing. the first bug was that if you died on a level without a mother object the game would crash this is because the game was trying to reset the mother without checking if it existed first, so I added a check to see if it was null. The second was that for some reason the first level was loading in two mothers on close examination it turns out we had put two mothers in the json file map so we took one out and fixed it.



**Post-mortem**

Group reflection

Overall the group worked well together and was well organised were at every point in development every one new what everyone else was working on however I believe the project management was not without its flaws

Rapid prototype

I think the rapid prototype went well and as a group we produced something of relatively good quality for are time frame. However I believe that we relied too heavily on the inbuilt features of unity and if we had tried to do certain mechanics like ropes without the unity features we would have been more prepared for the challenges of those mechanics later on in the production

Sprint to alpha

I believe that are alpha submission was of a high quality and contained most of the aspects that would be expected to be seen in the actually game but I think the group focused too much the gameplay features of the game instead of them more low level aspects of the game such as the level editor which would have made the sprint to hand in easier if it had been completed sooner.

Sprint to hand

After the alpha hand in the group took a large break that is understandable as people had other hand ins to work on however I feel like if we had met a couple of times in this break period to do group work we would have been able to include more features in the final game and refine the aspects of the current game. In saying this I think the level of team work in the sprint to hand in was incredible and the group pulled together to finish all the requirements ahead of time.

**Individual reflection**

I believe at the start of the project I wasn’t pulling my weight as much as other members but during the sprint to hand in I have really come in to my own and took on a larger work load. I think the majority of my work came out ok However there are still flaws with systems like the enemy that can only move in certain direction and if set up in the file in very particular way also I have used a large amount of a magic in the room creator that I wish I had changed in to something more OOP based. One of my major irritations with my work is that I left aspects like the in game level editor which I believe I could have finished if I had prioritised it. I would have done this by setting it up to create an object when the player has clicked on that objects corresponding button and then attached its position to the mouse until the player let go of the left click.