Dungeon Crawler CO3 Computer Science Programming Project

William Riddell

January 2, 2019

Contents

1 Analysis of the problem

2 Problem Identification

2.1 What will be developed

I will be hoping to develop a Medieval dungeon crawler game in C#, which is using the Visual Studio 2017 IDE. The game will allow the user to move their character around a randomly generated dungeon. This dungeon will look like a maze, with dead ends and loops, where the player will encounter monsters, trap rooms and treasure rooms. This is where the player will grow in power, by upgrading their statistics; like strength, carry weight, stamina etc. As the player moves through the dungeon the enemies will get stronger as the player levels up, until they encounter the Boss Room. This is where the character has to beat the Boss in a final battle to win the game. If the player dies within the dungeon then they lose, and they then must create a new starting character and begin all over again, growing stronger along the way to try to defeat the final Boss again.

The Setting for the game will be in a dark dungeon, with set back grounds which indicate where the dungeon will lead for the character, be it if there is a left turn, right turn, both, if it continues etc. I could have doors drawn into the background which could indicate the different rooms. Parts of the dungeon which haven't been visited yet will be hidden from the player, so they don't know what is inside the room. This means that they can move into a trap room at any time. Once the player is in a room though, they can "run" to the last tile they were at. This means that they have the option to not fight the monster, and instead take a different route.

There will be a small menu in the very middle bottom which will give the player all the options which are open to them, be it to move in a certain direction, and then to the side of their options, but still in the same box there will be the description of the current room. The menu will change when the player encounters an . The menu will show more options, these options are;monster change which weapon they want to fight with. weapon, it changes the icon displayed next to the character menu (Which will be talked about later) showing how much base damage and the multipliers the

current weapon has.The menu will give a description of what is contained in the current tile, and if a monster is in the tile, a description of the monster. This menu will allow the player to change their equipped weapon.

There will also be text which appears. This text describes the damage dealt or taken. This text will be located just right of the very middle of the screen and will keep the player up to date on what is happening in the battle, allowing them to make better decisions when playing through the game. and will be there to tell the user what is happening in the current fight so they can make the correct decisions. It is also there to make sure the user does not miss any important information which may lead to them losing the game.

There will be a small screen just to the left of the bottom menu, which will show the current statistics on the character. This will show the health of the player, their carry weight and then their currently equipped weapon. This will also show the currently equipped armor as well, showing what the player has a resistance too, and then the amount of damage it will deal to each type of monster.

To the right of the bottom menu, there will be the inventory, this will include, for example provision, gold and other treasures, as well as items like a sword and armor, it will also have items like torches. Extending this there will be a finite number of spaces in the inventory, this limits the space which the player has, meaning that they have to prioritise what they want to carry.

There will be a mini-map in the top right which will give a localised view of the area surrounding the player, showing only where they have been, and if they click this it will show a full map, meaning they can plan their route out, or explore areas.

If the player presses a certain button (Escape for example) it will open a main menu, pausing the game in the background and allowing the player to save, overwrite saves or quitting. Having this save feature will allow people to pick it up, and to then come back to it as the game, on the harder difficulties and on a bigger map will last for quite some time, meaning this would be a crucial feature.

3 Computational Methods

3.1 Thinking Abstractly and Visually

I need to begin to think Abstractly and Visually, this is because it would be impossible to accurately simulate everything which would happen in the dungeon, and would also be impossible to code, this is because of my own limitations and ideas. This means that some ideas will be lost for a more simple reconstruction which achieves the same effect. This could be for example when battling monsters, the monsters will have a single simple attack, and then multipliers which will be added to the attack. This means that it simplifies the idea that if the monster was real then it would instead have an array of different attack styles which will all do different "damages". Another example is that different attack styles would do different amounts of damage against different armor. This could be simplified with armor levels, as light armor like leather or chain-mail would absorb less damage and would therefore have a lower armor level. This is a abstract of real life as if you take light armor it allows the person to have more maneuverability, but at the same it would take less damage against certain weapons.

There are certain aspects of real life which I was not sure if I was going to include or not, this were aspects of life like eating and drinking. I was also thinking about including the idea that when you are damaged you deal less damage, as this would be similar to real life. After talking to my Stakeholders I have came to the final conclusion that eating and drinking would take too much effort to keep track of, and would take away from the core idea of the game. They would also think that the idea about decreasing the amount of damage that you do was a good idea at first, and then later Jimbo came back and said that it wouldn't actually be a good idea, so I don't think I will be including any of these aspects as it would be too close to real life and would take away from the main idea of the game, it would also feel unfair for the user to die in battles because of a strong hit by the enemy.

3.2 Thinking Ahead

When coding the dungeon crawler I need to look at the inputs which are needed for the game. I think that I would need the mouse for the user to select what options they want and what action they would want the character to do. This would be the first input. I would also need the keyboard as this

would allow the user to input the name and other information about their character. After this I would mainly use the mouse, and the keyboard would become unnecessary.

When coding my game I will need to code in error handling, this is when the player enters the wrong data type for example, and without the error handling the whole code will break, as there wouldn't be the expected data type, and would therefore go wrong when I try to work out values. Null values may also break my code, and I need to catch any errors when they appear.

3.3 Thinking Procedurally and Decomposistionally

My problem can be broken down into sections. I can separate the windows into a main menu, this would allow the player to select what they would want to do, this could be to open the options, to exit or to load up a old save or to make a new one. Each of these options will lead on to the next window. Each window will contain its own menus, only building what it needs and when it needs it. Having the forms like this means that the code will be much faster as it will be more optimised as it will only need to load up the menus which it needs. When the game loads the game make a new maze, this maze will be built from Tiles, and these tiles will contain either a monster or a room. There will be a tile class which means I can have a maze of any size as I can just make more tile objects. When I populate the maze with rooms, some of the tiles will be assigned a room, if it is not assigned a room, the tile will have the room value of null and will not contain a room. The same method will be applied to the monsters, as this will allow a even spread of monsters to be applied to the maze of any size.

As the player moves through the maze he will pick up items from the ground, this will be stored in their inventory. This will be a parent class, and then there will be a class for Gold, armor and then other misc items which for example could be potions. Having inheritance means that I only need to code certain aspects of each item once, as they all share the some of the same properties, such as being a item which can be stored in the players inventory.

3.4 Thinking Logically

When thinking logically, it concerns the idea of thinking with iterations and selection. There will be multiple loops in my code. On a larger scale the game doesn't have too much iteration, the game will have replayability though, and the user will have to make decisions, but the computer doesn't have to make many decisions itself, instead it just reacts to what the user chooses to do in simple ways.

When looking at the code there is a lot of For loops, this is classed under iteration. We can see that it is used when moving through the 2D arrays of tiles. When deciding what to do the computer uses If statements, this is called selection, and there will be multiple of if statements throughout my code, for example when the user has found a treasure room, the computer needs to send back the correct items contained in the room.

3.5 Thinking Concurrently

In my game there isn't code which is called at the same time. Within my game everything is executed in steps, which is dependent on what the user chooses to do.

Even when creating the maze there isn't any code which is executed at the same time as when the grid is created everything which is placed on to of it (The Monsters and the rooms) are all placed individually, I have executed my code this way to make sure that there isn't a monster placed over a monster etc. I could have 2 threads placing monsters throughout the maze if it is large enough, but at the current time there doesn't need to be another thread, so I will code this in at a later date if I have the time.

4 Stake Holders

5 Researching the Proposed Solution

Information about my Stakeholders My Stakeholders are Jimbo Giling, and Peter Cunningham. I think these two people will be suitable for my game as they both bring different angles to my project, and will input different views, ideas and solutions which appear. I know these people personally, and can easily contact them, this will allow me to get a fast response to my

questions even in time outside college. Having this fast response means that I can tailor my game to fit them as they are in the target audience quickly and efficiently.

I asked Jimbo to be my stakeholder because he will bring ideas for the enemies which the player will be encountering throughout the game. He has played many fantasy games which will give him a wide range of knowledge in enemies and encounters which I can code into my game. This will keep the game interesting even when the player is on a larger map.

Peter is more interested with the design of the character. He can bring armor, weapon and other item ideas. This could be the items which they are picking up, or the apparel which they wear as well as the apparel on the monsters. Peter can also help me with the HUD (Heads up Display) which could also be called the user interface, the HUD is a crucial part of the game, as this is what they will be interacting with, so a HUD which is simple to use and aesthetically pleasing would be crucial.

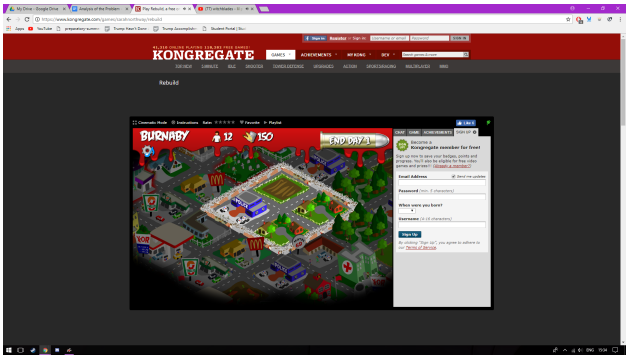
I think Jimbo and Peter alike have felt a dissatisfaction in the current release of RPG games and how alike the game-play is in all of them. This is why they are both looking forward to my randomly generated levels and interesting encounters. Peter has also found a lack of a challenge in current games, and therefore means that he loses interest in the games which he plays, so having a level system would suit him massively.

Jimbo took GCSE art, this will allow me to ask him about my art work, and will take ideas which he produces, making sure that the HUD and the monster design is to a certain quality. Peter plays a lot of single player games, this forces me to keep the game to a high quality so it keeps him interested.

5.1 Similar Solutions

5.1.1 Random Generation

I looked at a game called "Rebuild". In this game you are placed in a post apocalyptic world surrounded by zombies. The goal is to grow your base and survive until you reach a win condition. Every time you load into a new City the terrain around you will change, for example in this game I have a Mall next to my base:



In this game there are constants, you are given 12 people and 150 food to start with (This is in the top bar in each image). This is a good idea as this means when you start a new game you always know that you will have a strong start, and then from them you

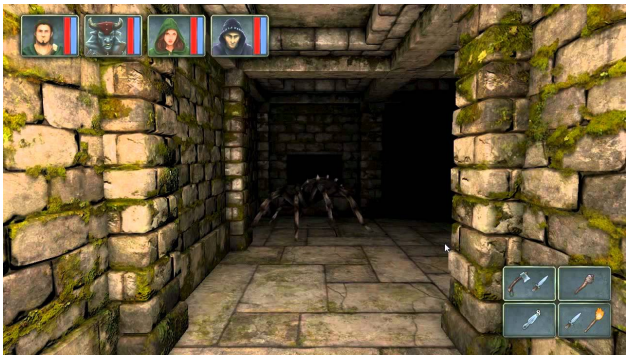
5.1.2 Ramping Difficulty

Furthermore in Rebuild, the zombie attacks on you city ramp over time, meaning it is always getting harder and more challenging. I would like to include this in my own project so the game doesn't get too easy in the late game, and also meaning the final boss will be around the same difficulty no matter when you face it in the game. I can either do this as time goes on, which is how this game works, or instead the average XP of the players characters. This will be standard for all difficulties, so when you select easy, the game will be easy thought out, or when you select hard, it is still hard until the very end.

5.1.3 Appearance and Audio

I much prefer having my game first person, this will increase the connection the player has to the characters, as well as increasing immersion. Even though the graphics are not realistic, I would like the audio to be convincing. So I will be looking for copyright free audio samples which I could use on my game. I wouldn't be able to achieve graphics like this so I would like to do basic pixel art, as this is more accessible for me. This will allow me to use skills which I have learned in GCSE art, and also to ask Jimbo, my stakeholder with advice for this as he also took GCSE art.

I have found a game called the Legend of Grimrock. This is what it looks like:



As you can see, it has items in the bottom right, and characters in the top left. This is somewhat what I would like my game to looks like referencing the HUD, but the background is exactly what I would like it to look like, although the graphics would be in a pixel art style, and will be 2D. I would like enemies to appear in front of the player when the player enters that monsters square. I don't know whether I will indicate what is in the next tile at this point in time. Furthermore I would like the enemies to have an indicator for the Health and other statistics, to give the character information about what they are facing. This is contained in to top left with each Character, but I will only have one character which the player will play as. This means I think I will have a separate menu section for the health and statistics which will be in the bottom left.

After talking to my stakeholders I should not indicate what is in the next rooms, but instead indicate where they can next go. This will keep the idea that everything is random until discovered and entered.

The inventory on the right will be exactly like that, but with an increased number of slots which is dictated by the characters carry weight, and inside the slots the item which they are carrying.

5.1.4 Combat

The game above is Pokemon, I will be taking the combat system from this game. As you can see there is the Players Pokemon in the foreground and then the enemy's Pokemon in the background, and to the side of each Pokemon there are statistics which are displayed to the player. This tells the name of the two Pokemon battling, the level of the two Pokemon and the health of the Pokemon. Furthermore there is a menu at the bottom of the screen which displays text to the player and then options which they can select.

When you enter a battle with an enemy you decide what to do on your first turn. This could be to "Fight" and then you can select the attack which you will use. You could use something from you bag, this could be to increase the HP of the Pokemon on the field. If your current Pokemon is low, "KO" (This means it can't be used) or too weak you can swap it out for another Pokemon. Finally, if you are going to lose this battle you can run, allowing you to safely move on. Pokemons combat is turn based.

I would like to implement all of these features into my game. For the "Fight" button, you could select the style of attack to use, for example you could "Lunge" this, for example could be a high damage, but low accuracy attack, compared to a "Stab" where it could be very accurate, but low damage. I could further this with multipliers, for example a "Slash" attack is weaker on armored opponents, yet stronger on less armored enemies. This can further develop the depth of the game, as as you move on and spot the boss,you can prepare your character to deal with that Boss type, which is dictated from the multipliers. For example there could be an Undead Boss,

which means you should try to find weapons and armor which is more adapt to dealing with undead characters, which could be a fire based attack, or the slash attack we talked about earlier.

The player should have an option to use their turn by selecting an item their inventory. This could be switching weapons, or using some provisions to heal up their character if they are low on HP. Finally, you should be able to run away from the enemy, this means that you move to the room you were last in. This will be used to take a different path way if there is a strong monster in the way.

5.2 Stakeholder Questionnaire

This is the Questionnaire which I will be sending to my stakeholders. The answers to the questionnaire will be stated below.

1. Would you like random monsters to be generated through out the grid?
    (a) Yes
        i. If so, what statistics should be randomised?
        ii. Should I make every enemy be around the same difficulty?
        iii. Should I have multipliers? For example, this monster has a specific weakness to fire.
        iv. Other Ideas
    (b) No
        i. How many monsters should I set?
2. Should I limit my inventory?
    (a) Yes
        i. The carry weight integer should be the number of items a single character should hold
        ii. Some items should not be carried, for example arrows.
    (b) No
3. Should I add a scoring system?
    (a) Yes

(b) No

4. Should I add a constant music, maybe ambiance?

   (a) Yes

   (b) No

5. Should I add audio cues, for example when you reach a treasure room, there is a small jingle which plays?

   (a) Yes

   (b) No

6. Should all the dungeons be the same, or should I change to a different background, for example it could be inside a pyramid?

   (a) Yes

   (b) No

7. Should the user be able to see all of the map?

   (a) Yes

   (b) No

      i. How much should I limit it by? Or should I only allow the player to view the tiles which trhey have been on?

8. Should I have a constant mini-map in the main form, or should I have a map form which they can always keep open?

   (a) You should have it in the Mini Map style

   (b) It should be contained in a separate form

9. Please enter further ideas on the game.

5.2.1 Stakeholders Response

I have kept this questionnaire short, and specific. I believe this will be better as I already have a good view of what I would like the game to look like. Although on some areas, im not too sure what to do and this is why I have asked my stakeholders. This will allow me to clarify what to do and how to attack these problems, as well as giving me new ideas to develop and move into the game. I can also ask my stakeholders throughout my project, as they are easy to contact them to get a quick answer.

Peters response:

1. Yes, I think you should have range where the monsters health, attack damage etc should be.
   This response was originally in line with my own opinions, as this will keep the mystery when moving into another tile, and facing a random boss.

2. *Yes, I think the inventory should be a visual grid, here there is a limited amount of squares.*
   Peter gave the example of the game Runescape, the inventory looked like this:

   This is the inventory which I will be going with, but I will have the spaces grow with the characters carry weight.

3. *I think there should not be a underlying score, and the win condition should only be if you have killed the final boss. I also think when you die, you die.*
   I don't think a underlying score would be good either, as this wouldn't be fair as longer games would yield a higher score, and would therefore be unfair.

4. Yes, I think there should be music and ambiance
   Making and finding music is hard, but I think I will be able to find some as I really want to have music in my game, as all of us want music in the game.

5. *Yes*
   I want to make my own jingle here, as I want it to be specific to the game.

6. There should be a variety of different biomes, for examples barbarian, Arabic / Indian as well as the Egyptian one.
   I really like the idea of the Egyptian and the Indian biomes, these will both be contained in the final game if I have time.

7. *No, when the player first spawns there should be a certain amount of tiles which are allowed to see, and then the rest should only become visible when the player visits them*
   This will allow the user to get a strong start, and not get killed at the beginning of the game. A problem with this is that you may be spawned next to a treasure room, and they won't have the luck of finding one, as it would be displayed that they were spawned next to it.

8. There should be a small mini map, and the a big main form. The main form will show you the whole size of the map, and the mini-map only a small amount of the map
   I like this idea a lot but I don't see the need of it if you can just have the main map form open in another window.

9. *I think there should be a Shop Room, this is where the player can sell his Gold which he has earn't from killing monsters.*
   This is a very good idea and I want to contain it in the final version of the game as it encouraged the player to tailor their character to the final Boss.

Jimbo's Response:

1. No, there should be set monsters, as when you encounter them, you know if you can take them on or not, and if you can't then you can run away.
   This makes sense as this allows you to predict what you are going up against which allows you to calculate if you can take the battle or not. This will allow consistency for the player.

2. *No, you should be able to carry as much as you like, but there should be a limit, and when you reach that limit you have a de-buff, this is when you do less attack damage for example*
   This is unlike Peters idea, what Jimbo is saying is that you can carry as much as you like, but the character has a certain value which, when exceeded your character has a multiplier to them, maybe making them weaker, or to tier easier in battle. I think I will still choose Peters idea, as this seems to make the most sense and is the closest to reality.

3. Yes, this should count monsters killed, your characters level, the amount of Gold you have carried etc. But defeating the final Boss should be the only win condition.
   I disagree with this, as if you make a scoreboard, the person with the biggest maze can get a infinitely big score as the maze could be infinitely big.

4. *Yes, I think there should be both*
   I think Jimbo wants a constant ambient sound in the background, this will add immersion to the game, but I think that making the ambiance would be too hard, and therefore I am going to try and find a copyright free ambient track.

5. Yes, there should be a jingle for finding treasure, and killing the Boss
   I think this would be a nice idea as well, so I am going to try to contain some small jingles in my game, although I will try to make my own.

6. *Yes, There should be one set in Hell, under water etc, I also like the Pyramid idea a lot.*
   It is good that we all like the pyramid idea, and the Hell and Underwater ideas would be interesting to make. Hopefully Jimbo could help me create the backgrounds.

7. No, only tiles which the player has visited should be visible on the mini-map
   I agree with this point as it means that the player has to travel to the surrounding tiles, making the boss room a surprise, and also the trap rooms hidden.

8. *I think it should only be in a separate form*
   This would be better for the user as it allows the to have it open on a different screen, or to have it constantly open in another window. This keeps the main form from being cluttered and will allow me to fit everything in with room to see the current tile.

9. There should be battle pets to help you along the way, you should be able to carry gear, rings etc which all buff you, for example make you deal more damage.
   I never through about pets, this is a good idea but will be hard to code, so I think I will add it in later iterations. Other gear would be fairly easy to add but I also think I will add it in later iterations as it isn't needed.

5.3 Explaining the Solution

"So what will the main menu look like?" "I would like my main menu to be a full screen application, with 4 options. The first option is to make a new save, the second one is to load a save, the 3rd one brings up options and then the 4th option is to quit."

"So what will happen when you select to make a new save?" "When you make a new save, it will change the screen to another full screen form, which will allow you to select what statistics you want for your character, the size of the grid and also the difficulty."

"And then what will happen if you select Load a save?" "When you load a save, you will be asked to give a path to a file. This path will be given to you when you last saved the game, after this is would of then loaded you into the last tile which you were on"

"So what is a Tile?" "The map is separated into Tiles, and in each tile is a area, or room which may contain an event, such as the Boss room, or having a monster inside of it etc. Some events may be good, like a treasure room, or bad like a trap room. This will be indicated on the map when it is opened."

"What sort of monsters will there be? And what sort of Trap Rooms will there be?" "Each monster, trap room, treasure room etc will all be randomised, although, I do want the randomised values to be in a bracket, so the player wont be facing any super hard or super easy monsters, or finding a treasure room, and getting nothing from it. The monsters will be from a set"

"So What will the actual game look like, once the player has loaded up their game?" "I want my game to be in full screen, with buttons in the top right, which open the menu, and in the top left to open the map form. I also want there to be, in the bottom left, a image which displays what the current characters health is. I also want any buffs which he is given from the item which he is holding contained in this picture. To the right of the character window, in the middle of the screen, but still on the bottom of the screen, will be the main bar, this will contain the current options for the player, and also any information which they are given. To the right of this will be the inventory, which will contain all the items in a list form, and also indicating when the player is over encumbered, which will then cause de buffs to appear in the character window. In the main window, which will take up the rest of the screen will be the background, which is dictated by the tile type, and anything extra which is contained in the tile, like a chest or a monster."

"What is the recommended audience?" "I want my audience to be young adults, so around 17 to 18. This is because the game will be quite challenging and will also take up a lot of time. It could though appeal to a younger audience, as the game will include a setting to make the monsters easier. This is why I have chosen you three as the stakeholders, as you are all inside of my target audience age bracket."

5.4 Limitations

I think my limitations are the graphical side of the project, as this will be incredibly time consuming, and will also mean that I have to learn how to draw pixel art. Another problem will be the audio, as I have no idea where to get the samples, or how to make them, so that will also be a limitation of the project as both of my stakeholders have agreed that there should be music. I think that I will try to get a copyright free ambiance which I can use for the background, and then make my own jingle as a audio cue.

A coding limitation would be the user interface of the solution, with the menus, and the buttons and drawing the character and the inventory windows

in the main game screen and achieving a professional look will be a challenge. I think getting the exact look that I want would be a struggle, but I think I could get a GUI which would be usable working.

Jimbo gave and idea for magic and battle pets, but at this point in time I don't think that that would be feasible with the limited time of the project.

6 The Proposed Solution

6.1 Requirements for the Solution

Requirement No: For The Maze:	Requirement	Justification
	To get the maze to be randomised, without any glitches or dead ends	This will allow a interesting game, every game. It allows the player to think about pathways, and to decide what monster to kill if there are multiple blocking their way.
	The Maze needs to be as big as the player wants it to	This allows the game to take as much time as the player wants, meaning that
	There should be monsters randomly scattered across the map	This will allow the player to move around the map without knowing what is next
	Each monster needs to be chosen from the list of monsters	This will allow the player to remember what monsters there, but also for it to be less repetitive

	The Boss room should be randomly placed	This will mean that the user can with the game.
	The maze shouldn't have any tiles which you can't visit	This allows the user to fully explore the grid, and allow them
For the Main Game Form:		
	The window needs to take up the whole screen	This means that the player has a more immersive experience, where everything is more ordered as it branches from this one main window
	There should be a bottom bar, where the options and description is displayed	This will allow the player to select their options, it also allows them to get information for the tile which they are in This will allow the player to select their options, it also allows them to get information for the tile which they are in
	The options button should be functional and opens another form over the main form	This will allow the user to exit the game, or save at any point

There needs to be a menu which displays the current characters stats, like health and carry weight	This will allow the user to judge whether they can take the next fight
The map needs to be displayed in a different form when the user clicks the map button	This will allow the user to see the map, allowing them to choose their path accordingly
There should be an inventory window in the bottom right	This will show the player what the have in their inventory, showing what items they have and how many they have of each item

Monster:

Each monster needs to be from the class monsters, meaning its attacks and statistics are set.	This will allow the user to predict if they can take the fight, and know what they are going to fight.
The monster needs to be draw to the form	This will allow the user to see the monster, allowing them to quickly recognize their enemy.
The form's options needs to change so he player can decide what to do	This will show the options to the user, allowing them to select what they want to do

	The monster inflicts damage and can take damage	This allows for combat which means the user can pass through the tile to the next area.
For the Character		
	The statistics need to be shown correctly, and the be edited / modified correctly.	This will allow the user to see what their health and other statistics are, meaning that they can make educated decisions
	The character needs to have their modifiers displayed to the user	This will allow the user to see what each item does, and the effect it has on them. This will allow them to choose what they want to equip
Items		
	The items need to be correctly rendered in the inventory	This will allow the user to see what they have, and if it is in use or not
	The Items need to stack	This will allow the user to store a lot of one item without filling up their inventory.

6.2 Hardware and Software Configurations

CPU	You will need a 2 Ghz or faster processor
Memory	You will need at least 4GB of RAM

Storage	You will need at least 2GB of storage
Graphics Card	You will need a DirectX Compatible 9 Card
Operating System	You will need to run a Windows Operating system

7 Iteration 1

7.1 Introduction to Iteration 1

In this section of Iteration 1 I will be talking about what I want to achieve.

I started iteration 1 with discussing what I want to include in my project. We leaned towards the generation of the grid and creating the monsters and the rooms which will go on the grid, from this we can then move towards making the user interface to display the grid which we have just made. This will allow us to see if my code is working correctly or not. This is what we have came up with:

Requirement	Time Cost
For the Maze:	
To get the maze to be randomised, without any glitches or dead ends	3 Hours
The Maze needs to be as big as the player wants it to	1 Hours
Monster and Rooms:	
There should be monsters randomly scattered across the map	2 Hours
There should be 2 Shops with in the map	1 Hour

There should be a Boss room inside the map	1 Hour
There should be the trap rooms inside the grid	1 Hour
There should be treasure rooms inside the maze	1 Hour

For the Character:

The player should be able to move around the grid one space at a time obeying the grids walls	2 hours

This is what we have came up with, it is 12 hours work and is achievable.

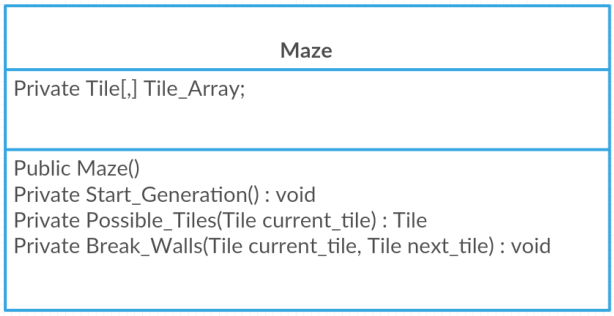
7.2 Making the maze

I started iteration 1 with the maze generation. For the maze generation I decided that I will use a recursive backtracking algorithm. This algorithm seemed simple to create and effective, it ended with a simple grid which will allow me to efficiently implement player movement into. This is the algorithm:

This is a over view of the algorithm, but from this I could create classes and make my procedures and functions. I made 2 classes in my code. The first of which was the Maze class, and the second was a Tile class (Called Cells in the algorithm above). These are the class diagrams for only the maze generation:

Algorithm 1 Recursive Backtracking Maze Generation

This will create a random maze of n size:
Make the initial cell visited
while There are unvisited cells **do**
  if $Current - Cell$ has unvisited neighbors **then**
    Choose a random $Unvisited - Neighbor$ of the $Current - Cell$
    Push the $Current - Cell$ to a $Stack$
    Remove the walls between the cells
    Make the chosen cells the $Current - Cell$
    Make the $Current - Cell$ visited
  else
    if The $Stack$ has cells in it **then**
      Pop the cell from the $Stack$
      Make it the $Current - Cell$
    end if
  end if
end while



In my Maze class I created a private tile array, this will contain all of the tiles which are in this instance of the grid. Having this array will allow me to call back on the tiles I have created. For example when I am populating the grid I can call on each individual tile to see if it is containing anything else. Furthermore having all of the tiles in this array means that when this instance of the maze is removed all of the tiles contained inside its tile array will also be deleted. This is more optimised than if I kept all of the tiles in the Game class.

In my Maze class I have the Constructor, this constructor uses nested for loops to populate the 2 Dimensional tile array I talked about above. This

allows me to better visualise the maze when I am breaking through walls later in the class. From here I also start my maze generation by calling the start generation procedure after I have made all of the tiles I will need.

The start generation procedure follows the recursive back tracking algorithm above. This moves through the algorithm calling the other functions and procedures as it goes. Within the start generation procedure there is the integer which tracks the amount of tiles which have been visited. This is used in the while loop which is the break condition of the algorithm. I will also create the stack which is used in the algorithm here. I have used a stack as it is suitable for the recursive algorithms, as the first item on the stack must be the last one off. When I start my algorithm I start the current tile in the top left of my grid. This is position (0,0).

This is my Start Generation procedure:

```
private void Start_Generation()
{
    int visited_tiles = 0;
    Stack<Tile> tile_Stack = new Stack<Tile>();

    // This starts us in the top left
    Tile current_tile = tile_Array[1, 1];
    current_tile.visited = true;


    while (visited_tiles < (Settings.grid_size * Settings.grid_size))
    {
        Tile next_tile = this.Possible_Tile(current_tile);
        if (!(next_tile == null))
        {
            tile_Stack.Push(next_tile);
            Break_Wall(current_tile, next_tile);
            tile_Stack.Push(current_tile);
            current_tile = next_tile;
            next_tile = null;
            current_tile.visited = true;
        }
        else
        {
            current_tile.visited = true;
            try
            {
                current_tile = tile_Stack.Pop();
            }
            catch
            {
                break;
            }
        }
    }


}
```

From here the next function which I use is the Possible Tiles function. This function takes in a tile and returns a random possible tile, if there are no possible tiles then it returns a null value. The way I made this procedure is by working out all of the x and y co-ordonates, then going through each one

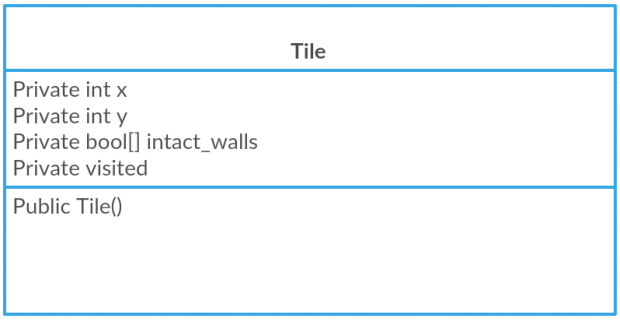
making sure that it is within the bounds of the array, this includes negative numbers and the max values of the array. After this I store the possible co-ordonates as points and find the corresponding tiles and adding them to a array of "possible tiles to return". Once I have finished adding all of the tiles I will check the size the array, if it is empty then I will return a null value, and if it has tiles then it will return a random tile from the list. The algorithm for this function is:

Algorithm 2 Possible Tiles

int $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4 \leftarrow$corresponding x and y to the current tile
Point $point_1, point_2, point_3, point_4 \leftarrow null$
$List < Point > valid_p oints \leftarrow List < Point > (Max \leftarrow 4)$
for all Possible points **do**
  $point \leftarrow newpoint(correspondingx, correspondingy)$
  $ValidPoints.Add(currentpoints)$
end for
for all Points in Valid Points **do**
  $tile \leftarrow tilearray[points.x, points.y]$
  if $tile.visited = false$ **then**
    $PossibleTiles.Add(tile)$
  end if
end for
if $PossibleTiles.Count = 0$ **then**
  $Return \leftarrow null$
else
  $PossibleTiles[random]$
end if

The next procedure which I use is Break_Wall, this function takes two tiles, these tiles are the current tile and the next tile. This procedure is simple and will check if the two tiles are either on the same x value or on the same y value, from there it will then check for the other axis and remove the two walls for each tile. It has to remove two walls because the tiles have an array of boolean values which tell the program which walls are intact, and to make a path way between the two tiles you need to remove the wall stopping the movement from each tile. For example; if you have current tile to the left of the next tile, and if you remove current tiles right wall you will still have next tiles left wall blocking you.

That is everything contained in the Maze class which is to do with the Maze generation, the tiles class' UML looks like this:



This class doesn't do anything as far as the maze generation is concerned, and inside the constructor it just stores the x and y parameters into the private x and y integers.

7.3 Monsters and Rooms

Once the maze has been generated I spawn the rooms and the monsters in a random location through out the grid. There must not be any overlap with the rooms or monsters, and they need to be able to have a custom spawn rate.

7.3.1 The Rooms:

I decided that every maze will have to have at least 2 shops inside of it, this is where the user can buy goods with the gold which he collected, and then a random amount of treasure rooms, which is where the user gets free goods and gold, and trap rooms which is where the user gets a certain amount of damage given to them and / or gets gold taken from them. For this I made a static class which is used through out the project which contains multipliers and statistics about the current instance of the game. This is what the class looks like:

```
public static class Settings
{
    //These are the enums which are used through out my project
    public enum enum_item_type
    {
        armour,
        wepons,
        misc,
        gold
    }

    public enum enum_weakness_type
    {
        none,
        metal,
        fire
    }

    //These are the global variables:

    //MAZE VARIABLES:
    public static int grid_size = 5;
    public static double monster_ratio = 0.4;
    public static double trap_ratio = 0.1;
    public static double treasure_ratio = 0.1;

    //Current Game in play:
    public static Game current_game;
}
```

In this class you can see that I have added the maze variables to it, these can also be set through the user interface, but if the player doesn't set their own settings then it will use these presets. From here I created an abstract class, this class will be used in the future, but for know it only holds the containing_tile, this links every room to the tile which it is contained in. Every Tile will also have a *object* which it holds either a room or a monster in.

I then made 3 more classes which are all rooms, currently they only contain a constructor which takes a tile parameter and sets the containing_tile to it. Containing_tile is contained inside the abstract class which is a parent to all of these classes. I have done it this way to allow me to easily see what is contained inside every tile, and to be able to expand it in the later iterations, as if they were only different enumerated values then it would be harder to implement different functions when a player enters the room.

I created a procedure called Generate_Rooms inside my Maze class, this was called on once the grid generation was completed. The algorithm looks like this:

Algorithm 3 Generate Rooms

```
shops ← 0
while shops < 2 do
  Tile tilearray[random, random]
  if tile.containing == null then
    tile.Add(newshop)
    shops ← shops + 1
  end if
end while
```

As you can see it has a integer value which is contained outside the while loop, so it will keep looping until 2 shops have been made, this is because the first 2 random tiles may already have a object inside of it. I have added this if statement into every room generation, as with out this the rooms may overwrite existing rooms or monsters, and if it overwrites a tile which contains the Boss in then it would eliminate the win condition from the game meaning it will never end until the user looses. Having this if statement will also allow me to make sure the possibility of the tile being generated and it actually being generated is as close as possible.

7.3.2 The Monsters:

I did mostly the same for the monsters, there is a abstract class which is the parent class for every other monster class, but contained in the abstract class are values which I can foresee myself using in the future iterations. An example of this would be the *hp* and *ap* for each monster, as well as its weakness and the multiplier which goes a long with that weakness. Furthermore the abstract class contains the same containing_tile which the abstract room class contains.

The Boss class and the Monster class are very basic though, as they only currently contain a simple constructor like the room classes, taking in a tile and setting it to the parents containing_tile value. To finish off the monster generation I matched my Room generation algorithm, but began all the generation (before the room generation as well) which dealt with generating the Boss room. This guarantees that the maze will have a Boss room in it. From there I placed down my basic Monsters through out the grid obeying to the multiplier contained inside the static Settings class.

7.4 The Character Movement

8 Used Websites

https://creately.com/app/?tempID=h165rwt81&login_type=demo#
https://en.wikipedia.org/wiki/Maze_generation_algorithm