**Resources**

Resources are an essential part of my project. As part of my project I need to identify what I myself need to develop the game and the hardware and software involved in running/creating the game. Hardware is pretty essential since it can actually influence who will buy the game and who won’t. What I mean by this is whose computer can run my game. This isn’t essential to know right now, but my game does need to be optimised by the end of the project as well.

In terms of **hardware**, here are some of the listed pieces of hardware and peripherals that I’ll need in the development of this project:

* A PC (preferably a Desktop PC)
* A Keyboard
* An Xbox controller (x4, two will do but in order to test four player I’ll need four)
* A Mouse
* A Monitor with 1920 X 1024 Resolution preferably (will explain why in description)
* USB (or Storage Device)

In terms of **software**, here are some of the listed software suites that I’ll need in the development of this project:

* Visual Studio 2010
* XNA C# AddOn to Visual Studio
* Adobe Photoshop
* Windows operating System (preferably windows 7, explain in description)
* Internet Browsing Software (chrome or IE)
* FruityLoops (Audio)
* Word

*These are the software and hardware requirements that I will need to design my game. In terms of hardware, here are some in depth explanations of why I chose what i needed.*

**Hardware**

**PC**

Pc is self explanatory, I need a desktop or laptop pc to actually program the game. I decided Desktop preferably because I feel the desktops at the college and desktops in general run things a lot better than laptops. I hope not to have an expensive or particularly fast PC as it only shortens my audience scope.

**Keyboard**

Self explanatory, need a keyboard to input stuff on screen when coding, and to use the PC in general. Keyboard input will be the starter testing input for the game before xbox input is tested.

**Xbox Controller**

The main functionality of my game will be through the xbox controller, so I will need an xbox controller to test what the user will be using to navigate the game. I said I will need 4x controllers because the game will have four player functionality and in order to test four players, I may need four controllers for a temporary time.

**Monitor**

Screen to view and test my game on. I chose 1920x1024 resolution because it’s the standard resolution that’s used on desktop PC gaming at the moment, and I can test what the game will look like on any resolution underneath 1920x1024.

**Mouse**

Self explanatory, used to navigate my screen during testing mode and to navigate desktop UI.

**USB**

Used to store my game and develop portably with.

**Software**

**Visual Studio 2010**

The C# .net coding language I will be using to develop my game. C# object oriented languages are my most experienced coding languages I’ve used to date (Livecode, XNA + Visual Studio). I will be using this coding language for my game and therefore need the software.

**Microsoft XNA**

I will need XNA as its an application framework used alongside Visual Studio for developing games. XNA allows me to choose to develop games on xbox, windows and windows phone if I choose to. for my game I will be developing for the xbox so it’s essential I need this software.

**Adobe Photoshop**

Used for all the art for my game. I have advanced experience with photoshop and pixel sprite art. Things such as my tile system, sprites and sprite sheets will all be developed through adobe photoshop and it’s essential I have this software.

**Windows Operating System**

The windows operating system is what I will need for developing my game. It’s the operating systems I am most used to. I would prefer windows 7 over windows 8 as it is currently more in use and I feel I have better experience using windows 7.

**Internet browsing software (chrome or IE)**

I will need internet browsing software in order to use google for research, help and advice, and I will also need browsing software to upload my game onto the moodle.

**Fruityloops**

Fruityloops will be the audio software I will be using to develop audio for my game. From sound effects to music I will be using this software to develop audio for my game. I have average experience with this software but I believe I will be able to develop at least some audio for the game before the deadline.

**Word**

I will need some documenting software to document my game and desk check algorithms. I may also need excel for documentation as well.

**Purpose and scope**

**Scope**

The project will be a short term project and have limited running costs. The costs associated with this project will largely be electricity, since a lot of the projects media and software/hardware needed is already provided in college and at home. The project will be a short-medium term project run over the course of 4-5 months. Some of the major milestones would be the Planning hand in, the development hand in and the evaluation hand in (of which is my final deadline in june).

This project should have little risks because of it’s size. The project really shouldn’t have many timescale issues as it isn’t a huge project to be working on and it can be something people can make in a few days. My project’s implementation should not exceed 2-3 months.

**Purpose**

The purpose of my project is to provide entertainment in the shape of gaming. The purpose is to create a game that is suitable for everyone aged 10 and over. My main objective of the project is really to find what's suitable for players aged 10-21, but also to find what's suitable to ages 21+ as well. If I do not meet the criteria of creating a game entertaining for both ages the product will be a failure. The product needs to be fun, and worthwhile, engaging and time consuming. The end product also needs to have elements of fun in the shape of friendly competition. If I cannot get the competitive aspect of my game to a good standard and have it become entertaining, I don’t feel my game will be any good to anyone for entertainment value.

**Functional and Nonfunctional Req**

*See file in folder provided for Functional and nonfunctional requirements.*

**Timescale and risk**

In terms of timescale, my entire project will be completed from January to the middle of may, a total of around 4 and a half months. The planning deadline is the 16th of February, the development deadline is in May the 18th and evaluation is in 8th of June. If you’re looking at this in terms of length, this is a pretty small project, however I plan to do more than 80 hours at college and do more than 200 hours at home. My project will be split into three aspects of design. The first is the design process that I want to go through, deciding what may be best for my project and what is a particularly bad idea. This will be my planning stage. The second aspect of my development will be the coding part of my project. This in theory should be the longest section of my development time but the third stage could actually take the most time. The third stage of design time is art. I think art will take the most time for me in the project since I want the project to be aesthetically pleasing to the most people possible and keep the game fun to play.

With timescales there are always risks that come with them. There are thousands of risks that can possibly happen but I have have limited my selection of risks down to the most likely to happen. These are some of the risks I’ve found :

* Home Computer breaks down
* Student server goes down on deadline day (believe it not, this can happen).
* One Part of the project takes a longer time than originally scheduled
* One Part of the project is too difficult to implement/out of my skill range.
* Bugs

Like I said these are only **a few** of the risks involved in this project. There are so many risks in this project due to timescale but I won’t list them all. I’ll take time to talk about each risk in turn and say why I believe these are the ones with the biggest risks.

**home computer breaks down**

This is self explanatory but a pretty big risk. As highlighted above I plan to do 200+ hours of home working. However if my computer broke down this would affect The development of my game drastically. In order to counteract this risk, I’ve taken action to spend as much time at college to do my work, so that if this risk does come true, then I would have sufficient time to finish my project in the timescale planned.

**Student Server Goes down on Deadline Day**

Although not an obvious risk, it’s still one that can happen. After speaking to my classmates, I actually found out last year during one of the deadlines at college, the student server went down. As part of the risk, I’ve decided that I set my deadlines a few days before the scheduled deadline, that way if the server DID go down, I would be able to say that my project is finished and handed in on time.

**One part of the project takes a longer time than scheduled**

Something that happens often with me is that I overestimate how great I am at programming. As part of the timescale risk this CANNOT happen. In order to counteract the risk I’ve done two things. First thing I’ve done was learn a bit more about c# programming, the more I learn the less likely this risk will happen. The other way I’ve solved this risk is I’ve created replacement steps in order to fill in the stuff I can’t do with ease if it came to that point. For example instead of the map screen warping, I can add a closing gap of lava from the left and right side of the screen to make it more interesting.

**One part of the project is too difficult to implement**

This risk comes under the risk of a part of a project taking longer than scheduled, but I can explain why I put it under this header. The reason it’s under this header is that a part of the project might just be too difficult for the timescale decided on. In particular this comes under light sources, since developing a lightsource generator may be above my skill level in C# programming, and just be too time consuming to develop under the 4 months I’ve been given to develop. In order to counteract this risk I have to develop something to put in it’s place to keep it interesting, whether it be bloom or shaders.

**Bugs**

Bugs are the source of all that’s unholy in programming. Bugs are the most time consuming thing that will affect my programming time. Bugs can turn a five minute job into an hour or more if I’m unable to debug it properly. The only way I feel I could get round this risk is if I give myself more time than I think would be necessary on a particular section of code. Instead of giving myself five minutes to implement something I could instead give myself maybe half an hour to get an understanding of how my code works, and desk check each algorithm so I understand how it works and if it works efficiently.

**Artificial Intelligence (AI)**

My game is a multiplayer game. Due to this, my Artificial intelligence will be **very** limited. almost, if not all of my games events are player driven events. Things such as the camera doing a screen warp is something driven by player combat and isn’t done by itself. Things such as power ups which shrink players is a powerup driven by the player obtaining it. The functions in the game that are **not** player driven are the blinking tiles that disappear and reappear elsewhere. These Tiles are time driven and controlled through an algorithm by the computer. None of the projectiles in my game are AI driven, or if they are, they are only driven by gravity and not directly controlled by the computer.

**Planning**

**Software Development Method**

My software development method will be a mix between the waterfall method and the agile method (a bit like Agile-Waterfall). In the beginning of my project, planning and analysis will all be done in order, one after each other in the order they are required. However in the Development/implementation stage, I will be doing a sort of agile method. I will be doing the test document and the test log and documentation along side the implementing of my game. In other words, I will be programming my game and testing my game as it goes along, and adding to my test log and creating a test strategy for each task I come across.

When I talk about it having waterfall features, I mean that I will iterate if something goes wrong during a stage. For example if I was in the development stage and I didn't feel like the menu had a user friendly interface and was clunky to use, I would iterate back to the design stage and redesign my UI. Another example goes along with my testing method of doing testing as I develop. If I was testing something and it didn't work the way I wanted it, I would iterate back to the implementation or even further back to the design stage and redesign my idea.

I chose this method because I feel I am most comfortable with it, and I've been working with this sort of method for almost three years of programming. In my project plan (ganntt chart), you can see how there's a difference between my development stage structure and my planning and evaluation stage structure.

A particular risk with this type of developing is that I could be landed with too much work at once during the development stage. In particular I could be forever stuck doing too much designing when really I could be doing a lot more in coding. Another example would be doing the test log and the implementation at the same time only proves to slow my implementation down, however it would be more reliable code since it is tested thoroughly.