Initial Meeting Project Plan (3/4 pages, plus diagrams)

Description of project:

For my project I chose to develop a Tower Defence Game in a linux environment. Over the summer I decided on a few key features of this project that I plan to implement beyond the just the foundation game. I decided on implementing an isometric game view, an unobtrusive and intuitive User Experience [/Interface(?)], a system of varying types of towers and enemies and a points system that can be used to buy/upgrade these towers. As well as these, I am also considering the possibility of either an easy-to-use map creator, or possibly a multiplayer element where in one player defends the objective, while a second player deploys enemies.

Look more at the general aims of the project (TD)

Description of “Proof of Concepts”:

[programs that prove you can code all of the algorithms required]

how we begin with limited capability and then iderate on this later one. (first running versin with game rules, then added features afterwards)

Abstract:

I chose this project as I have an avid interest in games and the gaming industry, and found myself passionate about my work during my first year Games module. As such, my belief that I wanted to enter the games industry after my degree was reaffirmed. For this reason, I felt choosing a project in which I could further expand on my python and explore game design possibilities would be beneficial upon applications for work. Also, it occurred to me that I would perform better on my chosen project if it was in a subject area that I was a passionate about, and therefore I feel like the quality of my project overall will benefit because of this.

What you expect the final project to achieve…

Timeline: break down to more granular and more descriptive. What was your rational? Why did I do this then. Consider slight overlap between coding and beginning of report writing. May be better to leave task if hit brick wall and then come back to later.

|  |  |  |
| --- | --- | --- |
| Milestone | Starting Date | End Date |
| Gather research materials | Aug 1st | Sept 26th |
| Set up Linux environment (the os, importing any libraries necessary) | Sept 22nd | Sept 26th |
| Obtain access to any materials possibly required for project (eg: Git, Pycharm, TeXStudio, Ubuntu etc.) | Sept 19th | Oct 3rd |
| Submit project plan |  | **30th Sept** |
| Foundation system | 30th Sept | **14th Oct** |
| Write interim report (one for each easy deliverable etc) | Nov 14th | **1st Dec** |
| Implement final graphics |  | Jan |
| Work on additional (extra) features/polish [list now or once decided on?] | Dec | March (mid) |
| Write final report | March 1st | 30th March 2017 |
| Project Presentation/Demo |  | June(?) |

Bibliography:

Question: what reference style should be used? IEEE referencing

Zed A Shaw: Learn python the HARD WAY, 3rd Edition (as well as online resouces)

Ian Millington. Artificial Intelligence for Games.

Ian Millington, Game Pysics Engine Development (?)

Either: Al Sweigart, Invent your own Computer games with Python OR Al Sweigart, Making games with Python & pygame.

Mark Lutz, Programming Python (O’Rielly)

Ivan Idris, Instant Pygame for Python Game Development How-to

Quick start guides, [LINUX: 2nd Edition! Beginner's Crash Course - Linux for Beginners Guide to: Linux](https://www.amazon.co.uk/LINUX-Beginners-Commands-Programming-Operating-ebook/dp/B014ECOU50/ref=sr_1_3?ie=UTF8&qid=1474981868&sr=8-3&keywords=python+linux" \o "LINUX: 2nd Edition! Beginner's Crash Course - Linux for Beginners Guide to: Linux Command Line, Linux System, & Linux Commands (Computer Science, Linux ... Programming, Linux Operating System Book 1))

Christopher Negus, Linux Bible OR William E. Shotts Jr., The Linux Command Line: a Complete Introduction

Various resources from:

* http://programarcadegames.com/

Risk Assessment:

(simply ideas atm, let me know if this is not the correct way of thinking)

* Software required for project not available
  + Remedy: research what will be required in advance of development
* Code becomes broken/unstable when operating in a linux environment
  + Remedy: develop and test code in a linux environment to begin with
* May not be able to find consistent people to test project against continuously over time
  + Remedy: house-mates/pay them/~~bored first years~~
* May not have a suitable level of coding proficiency to complete certain tasks
  + Mitigate by using and correctly referencing the source

What do you plan to do if this happens, how can they be mitigated? – either paragraphed or tables. Try to identify that which is relevant to project.