Project Plan:

A Tower-defence game for Linux

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.

Description of “Proof of Concepts”:

In preparation for my project, I plan to draft multiple ‘proof of concept’ pieces of code. One such proof of concept would be a basic implementation of the creep path-finding code that I will most likely re-purpose later and then implement into my final project, along with more advanced alterations to said code as I go. Similarly, I also plan to draft the code for the simpler tower AI operations, which, as with the previous code, I will also most likely integrate into my program as I go.

I also plan to test my Python User-Interface integration, as this is an element of the project that I have not used since my 1st year Games module. As such I will writing a basic frame with intractable elements as a proof of concept to revise these concepts, such that I can implement such features into my project early. Finally, I plan to make a conditional sprite-animation class, which, taking in arguments, produces one of a few possible animations using a sprite sheet. This will allow me to revise the basic concept of using sprite-sheets to construct animations in the program, and should hopefully allow me to implement more complicated animations into my project early on.

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.

Upon completion of my project, I hope to have created a fully functional tower-defense game, with 2D graphics and animations, as well as additional features such as a simple map editor that would allow the user to create a custom path for the creeps to follow and place the destination of them thereof, as well as basic abilities to alter the aesthetics of the map.

I also plan to implement the ability to save and load game files, both from within the game’s save file and via user-input of a save code. I also have hopes of adding multiplayer capability to game, either locally or over a network connection – however I imagine this to be a more challenging undertaking than the other features I have planned, and as such, will need to dedicate significantly more time to work towards it.  
Overall, I hope to gain a better understanding of the design and game-making process such that I am better equipped with the skills needed to enter the game industry upon leaving university.

Timeline:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | | | **Start 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 draft | | | | **30th Sept** |
| **Early Deliverables:** | | | | |
|  | First running version with simple game rules | | 5th Sept | 14th Oct |
|  | Allow players to place towers and win or lose on one map | | 14th Oct | 22nd Oct |
| **Interim Report:** | | | | |
|  | Design patterns and code organisation | | Nov 14th | Nov 17th |
|  | Principles of free software and open-source code constraints | | Nov 18th | Nov 19th |
| Interim programs and report hand-in | | | | **1st Dec** |
| Interim review viva | | | **5th Dec** | **9th Dec** |
| Full Unit draft report ready | | | 10th Feb | **17th Feb** |
| **Final Deliverables:** | | | | |
|  | Implement reasonable design (multiple creeps & towers, graphics and animations) | | Nov 16th | Dec 5th |
|  | **Additional Features:** | | | |
|  |  | Save and Load options | Nov 10th | Nov 20th |
|  |  | Several rule sets | Nov 20th | Nov 26th |
|  |  | Map editor | Dec 3rd | Dec 15th |
|  |  | Local/network Multiplayer | Dec 18th | Jan 4th |
| Player testing & feedback collection | | | Jan 10th | Jan 20th |
| **Final Report:** | | | | |
|  | Description of software engineering principle used in software | | Jan 4th | Jan 12th |
|  | Description of message-passing in software, and constraints and benefits associated | | Jan 15th | Jan 23rd |
|  | Test data analysis and discussion | | Jan 24th | Jan 28th |
| Full Unit final programs and report | | | March 1st | **30th Mar** |
| Project Presentation/Demo | | | | **June**(?) |

Bibliography:

Zed A Shaw, “Learn python the HARD WAY”, 3rd Edition (as well as online resources)

Phillipa Avery et al, “Computational Intelligence and Tower Defence Games” (<http://julian.togelius.com/Avery2011Computational.pdf>)

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

Wesley, Advanced Programming in the UNIX Environment

Various resources from:

* <http://programarcadegames.com/>
* <http://codeskulptor.com/>
* <http://codecademy.com/>

Risk Assessment:

The first possible risk that I could encounter upon beginning my project is that one or more of the software that I require could be found to be unavailable. This risk is also the easiest to mitigate, as it simply requires that I put in adequate research into such software to verify it’s availability.

The second possible risk that needs to be kept in mind, is the possibility of code becoming broken or unstable upon transferring it over to and running it in a Linux environment. Again, this risk can be easily mitigated by simply writing all my code in a Linux environment from the beginning of the project.

Another risk that could effect the progress of my project is being unable to find consistent people to test my project in it’s varying states continuously over time. This could come about as a result of being unable to find consistent people to test my project. However, this can be remedied by either convincing my house-mates to test my project for me, as they will be easily accessible, or, if not possible, possibly paying colleagues such that I can reliable get them to give me feedback data on my project.

One final possible risk worth considering is the simple fact that I may not have a suitable level of coding proficiency to complete certain task in my project. The obvious solution to this however is to simply use and correctly reference the source of the solution that I find.