|epcc|

Software Development Planning and Risks

B098688 - s1671778

February 9, 2017

Contents

1 Introduction

This report will provide a series of guidelines on how to proceed with the creation of a web application. The web application's code has been initiated by an unknown party, that is written in `html`, `JavaScript`, `python` and uses the web framework `flask`. Said code must be enhanced such that it fulfills the base requirements for the project. The skeleton of the web application has a few elements already built in. It is, however, very rough and needs much work to become a feasible end product.

The first part of the report will briefly outline what is the intended project, and the background needed to understand what it is attempting to achieve. While this is being discussed, some of the important factors to keep in mind are highlighted. After this is done, several comments regarding issues in the provided code are made. Afterwards, solutions for most of these are proposed, and eventually presented in a *time/effort* estimation plan. Finally, once the project has been understood, and its purpose is clear, risk analysis and management strategy are posed. For the latter, a Gantt chart is put together to act as a guide for the next steps: *UI and Evaluation Plan* and *Refactoring and Reflection*.

The management strategy in this small project will help make a successful end product. There are many factors to be analysed in order to properly develop the software development planning. Considering personal inexperience in web security, `flask`, `html`, and `JavaScript`, it is safe to assume that many issues with the program will be overlooked at the beginning. This means that the problems mentioned in this document will be a fraction of those that will actually be fixed in the later stages of the project. This will require allocating time to research on the aforementioned topics as well as choosing carefully the best development model. After careful consideration and taking into account that this project will need to cycle from the current code to an improved version a few times, the *Waterfall with Subprojects* development model is going to be used.

Having mentioned how the lack of experience will impact the development model required, it is important to analyze how it will modify other elements of the project's progress. The best way to do this is most likely through the Gantt chart. In it, tasks that have to be completed and are related to areas of inexperience will need some time allocation for 'research' or 'practice', but this will decrease as more tasks of the same area are performed, i.e. account for knowledge growth.

2 Project Specifications

The goal is to further develop a web based application that allows the user to generate a squad (Warband) for a tabletop game - with certain constraints. The application must allow users to create and edit squads with an unchangeable name, no more than 10 members, and no fewer than 1. The user begins with a Captain, that can become better with time, and has a limited amount of money to hire other regular team members, that

cannot improve with time, and a special member, the Ensign, who, like the Captain, can evolve in time.

The idea of the web application is to check that all constraints are fulfilled for a user's Warband, and that if one or more are violated, a warning message is displayed such that the user is able to modify their decision. The base stats for each team member are shown in the Annex.

2.1 New Warband

A new Warband begins with **one** Captain and **500** credits. For a new Warband to be created, the following conditions must be met:

1. Warband Name.
2. Player Name.
3. Positive number of credits.
4. Captain:
   (a) **ONE** per Warband.
   (b) Starts with **one** EXPERIENCE.
   (c) Must be assigned **one** SPECIALISM (Engineering, Psychology, Marksman, Tactics, Melee, Defence).
   (d) **One** Associated Skill must be assigned to the specified SPECIALISM:
      i. Engineering: [Repair, Sabotage, Augment].
      ii. Psychology: [Bolster, Terror, Counter].
      iii. Marksman: [Aim, Pierce, Reload].
      iv. Tactics: [Squad, Ambush, Surround].
      v. Melee: [Block, Riposte, Dual].
      vi. Defence: [Shield, Sacrifice, Resolute].
   (e) Must be given **one** WEAPONS/EQUIPMENT:
      i. Blaster: **5** credits.
      ii. Needle Gun: **12** credits.
      iii. Blade: **3** credits.
      iv. Cannon: **15** credits.
      v. Whip: **5** credits.

5. Ensign:
    (a) Price of hiring: **250** credits.
    (b) Maximum **ONE** (Warbands can be created without an Ensign).
    (c) Must be assigned **one** SPECIALISM.
    (d) Must be assigned **one** Associated Skill.
    (e) Must be given **one** WEAPONS/EQUIPMENT.
6. The total number of members is **10** (including the Captain and Ensign).
7. Prices of hiring are:
    (a) Augment Gorilla: **20** credits.
    (b) Lackey: **20** credits.
    (c) Security: **80** credits.
    (d) Engineer: **60** credits.
    (e) Medic: **50** credits.
    (f) Commando: **100** credits.
    (g) Combat Droid: **150** credits.

2.2 Edit Warband

When editing a Warband there is also a set of conditions that have to be met. They are outlined here.

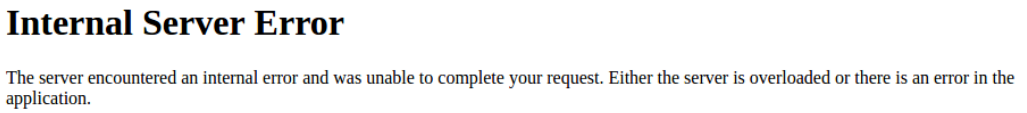
1. Can hire squad members as long as the total number of squad members does not exceed **10**, and as long as the Warband's number of credits does not become negative.
2. Captain and Ensign can gain EXPERIENCE.
3. Acquired EXPERIENCE can be traded for upgrades for Captain and Ensign (**10** EXPERIENCE points can be turned into **1** point for a Stat up to specified maximum values shown in the Appendix section).
4. Can remove squad members, but their hiring price is **not** returned to the number of credits.
5. The Captain can have up to **two** weapons and **four** items.
6. The Ensign can have up to **one** weapon and **3** items.
7. Weapons cost credits. Changing or adding weapons **cannot** result in a negative number of credits.

3 Identified Problems

Now that the constraints of the teams have been covered, it is possible to go through the code to check missing or wrong information. But first, going through the current web application to see errors in its functioning is important. The following is a list of problems that are observed by working though the web application alone.

3.1 New Warband

1. The most obvious deficiency is the fact that Warbands are not linked to a specific player. This means that anyone can edit any Warband. The game must support more than one player, where only the team's owner can edit it, but all other players can see it.
2. There is no way to go from the "New Warband" or "Edit Warband" pages to the "Home" page without using browser back button or address bar.
3. There has been no attempt at testing the current product. For example, when attempting to generate an illegal "New Warband" (Captain without weapon / skill / specialism), an error message is given, but no explanation or way to return and fix the mistakes are provided.



Generic error message.

4. The price of weapons is not discounted from the number of credits from the screen when creating a "New Warband".
5. Neither Captain nor Ensign have "Notes" section, and merged "Weapons and Equipment", whereas the two should be different.
6. The Ensign is called "Apprentice" for some reason - this is confusing.
7. The cost of the "Apprentice" is 200 credits instead of 250 as it should. This and the previous item are actually related, and comprise a larger problem - the information from each team member is actually stored in the `python` code and separately in the `html/JavaScript` code. Due to this, the price on screen is 200 credits, but it actually subtracts 250 credits (as it should).
8. A 10 row table allows the player to fill out the form with up to 12 members (returns the generic error message when done so).

Move	Fight	Shoot	Shield	Morale	Health	Specialism	Weapons and Equipment	Skills
5	2	2	12	4	12	Tactics	Cannon	◉ Squad ○ Ambush ○ Surround

Hire apprentice? ☑ 200gc

Move	Fight	Shoot	Shield	Morale	Health	Specialism	Weapons and Equipment	Skills
7	0	-1	10	2	8	Defence	Blade	◉ Shield ○ Sacrifice ○ Resolute

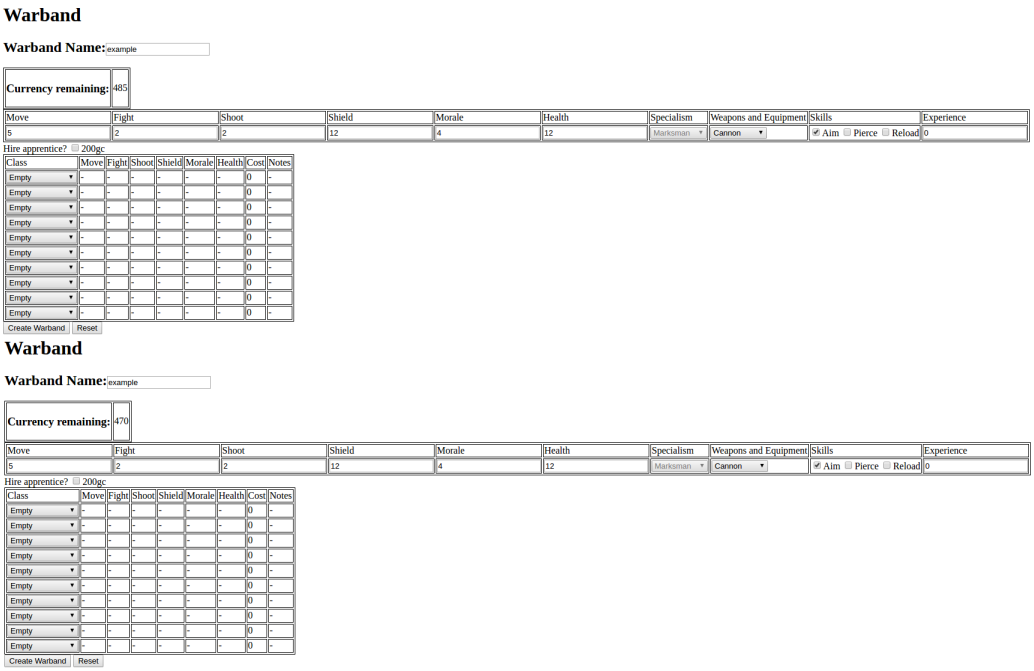
Class	Move	Fight	Shoot	Shield	Morale	Health	Cost	Notes
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Lackey	6	2	0	10	-1	10	20	Melee Weapon

Chance to include 12 members.

9. There is no method for handling attempts at creating a new Warband with a name that already exists. This has to be carefully thought out when including multi-player - no two teams can have the same name, even if they are built by different players - involving the client is important for this decision.

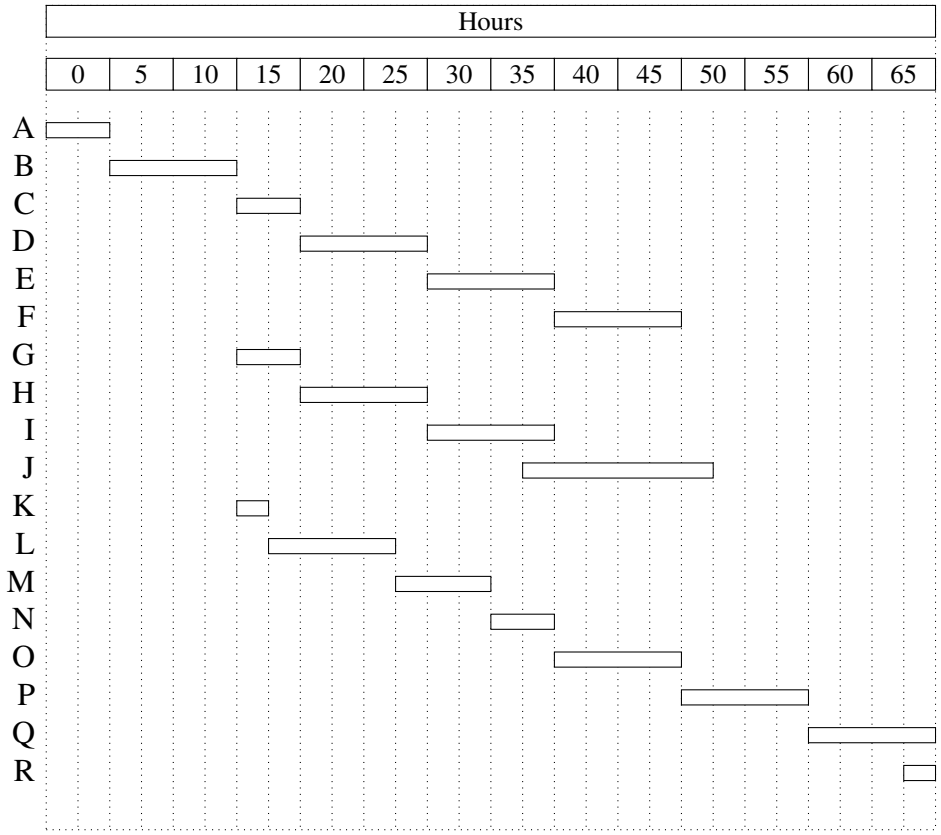
3.2 Edit Warband

1. The most serious problem in the "Edit Warband" menu is the handling of credits. When editing the Warband the number of credits is modified erroneously:
   A new Warband with only a Captain with a cannon (15 credits) is generated. Then, in the "Edit Warband" page, clicking "Create Warband" decreases the number of credits by 15 (shown in the figures below). Like this, there are other errors with the credit handling in the "Edit Warband" page.



2. It is not possible to hire an Ensign after creating the Warband.
3. Squad members are treated like slaves, i.e. it is possible to sell them for the original amount of credits that they were "hired" with. This needs to change according to the game rules.
4. The rules for increasing the Captain's and Ensign's skills is completely off - it is unlimited and completely uncorrelated to the experience points.
5. The Captain and Ensign maximum values of skills (Annex) are not taken into consideration.
6. Captain is not able to have **two** weapons.
7. Neither Captain nor Ensign can obtain equipment.
8. When attempting to make an illegal Warband nothing happens - not even the generic error message returned previously.
9. Impossible to add number of credits.

4 Project Planning



One of the important restrictions of the model is that neither hares nor pumas can live in sections that represent water. Therefore it is straightforward to expect that if a landscape is entirely covered in water it should run faster than one completely covered in land, since no computations related to the population densities should be required. This is the motivation behind the first performance test.

For an input file of fixed size the percentage of land to water is varied from 0% to 100%. Using a bash script, the time it takes to run the program with different l

The output confirms the intuition that the program's run time should increase as the percentage of land in the input landscape file is increased. Furthermore, it shows that the aforementioned growth has a direct

5 Code Profiling

In order to actually know which functions make up for most of the time in the run time for the program, the python native profiler "`cProfile`" was used. First, two input files of the same size (100x100) one with 30% and the other 60% land-to-water ratios were profiled. The output shows how much time each function took and how many

times they were called. These results are

These figures provide a more complete picture of what can be done to optimize the program under study. The first thing that is evident is that `ppm.py` is using most of the run time in all four cases. This file holds the functions that are used to generate the `.ppm` files. Hence this is where it would be most beneficial to attempt some type of optimization. Additionally, the second set of pie charts makes it clear that the files `population.py` and `landscape.py` are also good candidates for possible optimization.

6 Conclusions

After performing a series of performance and profiling tests on the program written by *Callum Black, Andrés Cathey, and Eskil Joergenssen* for the HPC course **Programming Skills** a few good features can be made about the run times of said program. Additionally, it was possible to know what functions should be optimized in order to do a significant impact on the program's run time.

The first feature of the code that was studied was how the program's run time increases as the land-to-water percentage is increased. A linear relation between these was found, confirming that the computation time increases when the percentage of land in a fixed size file is increased.

Afterwards, it was shown that the program's run time also grows linearly when the input landscape file's number of land/water squares is increased linearly. This confirmed the intuition that the most important factor for the program's run time is the landscape's area.

Finally, the profiling part of the tests provided information as to what functions could be optimized in order to decrease the run-time of the program. Said optimizations should be done for writing the `.ppm` file, since this always represents most of the run time regardless of input file size (the percentage of the run time that it represents should also grow linearly).

It can be concluded that performance testing and profiling are important to know what are the weaknesses and the strengths of large, or complex, codes that cannot be studied by simply going through them. These methods are also very useful to know what sections of a code is taking up most of the run time and, thus, best suited for optimization.

7 Annex

7.1 Base stats

Besides the stats shown in table 1, both Captain and Ensign start with empty Skillset (that has to be filled when creating a squad), as well as an Associated Specialism, and no Items (one Weapon has to be added when creating a squad). All other stats are summarized in the following table.

Squad Member	**Move**	**Fight**	**Shoot**	**Shield**	**Morale**	**Health**	**Cost**	**Experience**
Captain	5	2	2	12	4	12	0	0
Ensign	7	0	-1	10	2	8	250	0
								Notes
Augment Gorilla	8	3	0	10	2	8	20	Animal, cannot carry treasure or items
Lackey	6	2	0	10	-1	10	20	Melee Weapon
Security	6	2	1	12	2	12	80	Blaster, Blade
Engineer	4	0	3	12	2	10	60	Blaster, Repair Kit
Medic	5	0	0	12	3	10	50	Blade, Medkit
Commando	8	4	0	10	4	12	100	Stealth Suit, Blade, Needle Gun
Combat Droid	3	2	4	14	0	14	150	Mechanoid, Dual Blaster, Claws

Table 1: Base stats for each squad member.

7.2 Maximum Stats

As the Captain and Ensign are able to increase their stats, it's important to state their maximum stats. In order to increase any of these stats by **1** point, the corresponding squad member (Captain or Ensign) will loose **10** EXPERIENCE points. In one edit no stat can be increased by **more than 1** point.

Squad Member	**Move**	**Fight**	**Shoot**	**Shield**	**Morale**	**Health**
Captain	5 (+0)	2 (+8)	2 (+8)	12 (+0)	4 (+8)	12 (+8)
Ensign	7 (+0)	0 (+6)	-1 (+6)	10 (+0)	2 (+4)	8 (+8)

Table 2: Maximum stats for Captain and Ensign.