Project Proposal Draft
for UTM CSCI 352

Mason Bearden Jordan Taylor

Abstract

The user plays as a character and gets to choose a "student" to compete in battles with. They will traverse through a map containing trees, water, buildings (some being interactable), random encounters of "students", and boss fights (the professors). Based on the "student", there is a variety of ways to play the game being that each have different abilities / stats and typing. The boss fights will consist of Mr. Bradley, Dr. An, Dr. Wang, Dr. Ericson, and Dr. Guerin. Each boss fight will be placed in an appropriate area based on their abilities. As an example, Dr. Guerin with artificial intelligence will be placed in a mechanical-like location. The player will gain an attribute from each boss, learning some of their skills. The skills will be randomized as a reward (this skill reward may eventually be reconsidered as it might take too long). The player's "student(s)" can level up to gain new abilities and increased stats. The buildings, being undecided so far, will offer the player special features such as healing. We hope that in making this game we can develop more ideas on this concept. However, we need to get the base of the game started before considering this feature.

1. Introduction

The expected users that would be interested in this game are the students, the professors, and those who enjoy RPGs (role-playing game). At the start of the game, you will be able to pick your starter "student". Students will provide a different experience to the game in terms of combat. The combat will be turn-based involve health and a set number of times an attack be used before being unusable. Level up your "student" and gain attributes to take on the professors, potentially learning something from the battle (unsure yet).

1.1. Subsection Heading Here

Occasionally you need to break your sections into separate parts, you will likely not need a subsection for every section

1.1.1. Subsubsection Heading Here. Occasionally you will need to break your subsections into separate parts, if you find yourself using this often, you're likely going overboard. Don't try to go any lower down than this. (And make sure you remove this!)

1.2. Background

We are interested in learning the concepts of a turn-based combat game with an adventure. It will provide a variety of skill sets that we will be able to use in feature programs. Having the experience of creating an RPG could potentially open up job opportunities.

1.3. Impacts

This game will hopefully provide enjoyment to those who are stressed or ill, making life a little more exciting.

1.4. Challenges

The toughest aspect of our game is managing our time. We have many ideas for implementation and little time for them all. I believe that the toughest portion of the game will be all the small things working together such as: traversing through the map, stat loss / gain / upgrade, capturing the students, leveling the students up, and rewards from combat.

2. Scope

Scope: The bare minimum for the project that we want to accomplish is having at least a character and all types working as expected, along with having the bosses fully developed.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Add item to cart	Shopper	Med	1
2	Checkout	Shopper	Med	1

TABLE 1. SAMPLE USE CASE TABLE

2.1. Requirements

This is where I am putting the stretch goals

2.1.1. Functional.

- Implementing in depth class abilities and type management
- Functioning buildings and balance of the game

2.1.2. Non-Functional.

- Security – user credentials must be encrypted on disk, users should be able to reset their passwords if forgotten
- you'll typically have fewer non-functional than functional requirements

2.2. Use Cases

This subsection is arguably part of how you define your project scope (why it is in the Scope section...). In a traditional Waterfall approach, as part of your requirements gathering phase (what does the product actually *need* to do?), you will typically sit down with a user to develop use cases.

You should have a table listing all use cases discussed in the document, the ID is just the order it is listed in, the name should be indicative of what should happen, the primary actor is typically most important in an application where you may have different levels of users (think admin vs normal user), complexity is a best-guess on your part as to how hard it should be. A lower number in priority indicates that it needs to happen sooner rather than later. A sample table, or Use Case Index can be seen in Table 1.

Use Case Number: 1
Use Case Name: Add item to cart
Description: A shopper on our site has identified an item they wish to buy. They will click on a "Add to Cart" button. This will kick off a process to add one instance of the item to their cart.

You will then go on to (minimally) discuss a basic flow for the process:

1) User navigates to page listing desired item
2) User left-clicks on "Add to Cart" button.
3) User cart is updated to reflect the new item, this also updates the current total.

Termination Outcome: The user now has a single instance of the item in their cart.

You may need to also add in any alternative flows:
Alternative: Item already exists in the cart

1) User navigates to page listing desired item
2) User left-clicks on "Add to Cart" button.
3) User cart is updated to reflect the new item, showing that one more instance of the existing item has been added. This also updates the current total.

Termination Outcome: The user now has multiple instances of the item in their cart.

You will often also need to include pictures or diagrams. It is quite common to see use-case diagrams in such write-ups. To properly reference an image, you will need to use the `figure` environment and will need to reference it in your text (via the `ref` command) (see Figure 1). NOTE: this is not a use case diagram, but a kitten.

After fully describing a use case, it is time to move on to the next use case:

Use Case Number: 2
Use Case Name: Checkout
Description: A shopper on our site has finished shopping. They will click on a "Checkout" button. This will kick off a process to calculate cart total, any taxes, shipping rates, and collect payment from the shopper.

You will then need to continue to flesh out all use cases you have identified for your project.

Figure 1. First picture, this is a kitten, not a use case diagram

2.3. Interface Mockups

At first, this will largely be completely made up, as you get further along in your project, and closer to a final product, this will typically become simple screenshots of your running application.

In this subsection, you will be showing what the screen should look like as the user moves through various use cases (make sure to tie the interface mockups back to the specific use cases they illustrate).

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure 2.

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

Figure 2. Your figures should be in the *figure* environment, and have captions. Should also be of diagrams pertaining to your project, not random internet kittens

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.