S6

Mom and Dad’s Pizza Pad Pizza Adventure

Supplier Product Evaluation

Team C - COMP 3663 X1
Scott Barnett 100123218, Jimmy Flemming 100116643,
Liam MacKinnon 100114393, Martin Main 100110854
April 11, 2016

Table of Contents

1. Introduction

This document is an evaluation of our product, *Mom and Dad's Pizza Pad's Pizza Adventure*, and how well it satisfied the requirements of our customer Mom and Dad's Pizza Pad, and also consider the effectiveness of our development process. We will first review the primary requirements of our client, and discuss our level of satisfaction with fulfilling those requirements. Then we will take a critical look at the design process we undertook for the process, and comment on what aspects we are proud of, and what needs improvement. We will also be considering the impacts of our integration and test plan on the development stage of the product, and how well we used these resources in the construction phase. We will then make some suggestions for future development and discuss the lessons learned from the project.

2. Satisfaction of Requirements

2.1 Satisfaction of Original Requirements Specifications

2.1.1 Basic Functional Requirements

2.1.1.1 Functional Requirements

Fun and replayable
-The map is randomly generated to guarantee a measure of replayability
Standard time limit
-A timer is shown out of combat
Option to see current progress
-Current progress is displayed on screen at all times outside of combat

2.1.1.2 Plot Requirements

Pizza stolen by anger crew
Try to recover pizza before it's eaten
Deliver whatever was recovered before time runs out

2.1.1.3 Character Requirements

Character can be created by user
-User can select Pizza Boy Joe or Garlic Finger Gina from the start menu
-Character starts with 100 health

2.1.1.4 Item Requirements

Static inventory
-All items are implemented fully except for Mr Michel’s Map

2.1.1.5 Combat

Follows a Rock, Paper, Scissors format
-Pizza Box Smash is strong against Pizza Bazooka
-Pizza Cutter is strong against Pizza Box Smash
-Pizza Bazooka is strong against Pizza Cutter

2.1.2 Human Interface Requirements

2.1.2.1 Map Requirements

Grid style map
Starts on grid tile
Max one anger crew per tile
Each tile has three qualities
-2 of the three qualities are implemented
  -Poison gas reduces energy
  -Pizza whiff restores 10 energy

2.1.2.2 UI Requirements

Records and prints all relevant actions that happen
-All actions outside of combat are displayed to the user
Keep the player up to date

2.1.3 Non-Functional Requirements

Graphics are simple
Runs on all desktop platforms
Easy to open and run
Downloadable from website
Markets Mom and Dad’s Pizza Pad throughout game

2.1.4 Undesirable Requirements

The game is not offensive to any groups of people
Our user manual is not too lengthy, and is easy to read

2.1.5 Time and Cost Constraints

Deliverables must be received on time. No deliverable can be accepted before a previous deliverable has been received and approved.
-All deliverables were delivered on time

2.2 Satisfaction of the Changed Requirements Specifications

There were no changes to the requirements specification.

2.3 Non-Satisfaction of Requirements Specification

2.3.1 Functional Requirements

Time limit cannot be decided by the user.
Saving and loading were not implemented.
High scores were not fully implemented.

2.3.2 Plot Requirements

All plot requirements were fulfilled

2.3.3 Character Requirements

Stats are not unique upon creation.
-Endurance doesn't change
-Speed and intelligence are modified by moves and items
Energy reaching zero currently does nothing.
As energy reaching zero does nothing, there is no alert for when it approaches zero.

2.3.4 Item Requirements

Mr Michel's map currently only shows some text.

2.3.5 Combat Requirements

Speed never goes down, causing a bug where the player will dodge every attack once speed hits 100

2.3.6 Map Requirements

Streets tiles do not have authentic names.
The slipperiness quality is not fully implemented

2.3.7 UI Requirements

Actions in combat are not displayed to the user.

2.3.8 Non-Functional Requirements

No other pizza place is even mentioned throughout the game, let alone dissed.

2.3.9 Time and Cost Constraints

The project costed 350+ hours, exceeding the cost constraint.

3. Analysis and Design Process

The design phase was an important part of the software development process, and it guided the later parts of development in implementation, testing, and documentation. The requirements stage proved useful throughout every stage as a document to ensure that the development was on track with what was planned.

During the implementation stage, we used the design document to create a framework for the rest of implementation, by creating a skeleton framework of all the classes, with comments describing what each module should do. This made separation of work much smoother, and helped to organize the project into a logical format.

The User Manual also proved useful in the front end design, to ensure that the user interactions with the software lined up with the user documentation.

In the future, if I were to do something different for the design stage, we would want to flesh out more of the implementation details, which would require a better working knowledge of the libraries we were using. One of the biggest source for implementation change was the use of our game library we used for the project, which provided many classes that made certain parts of implementation easier, but sometimes caused more work to interface the new classes.

4. Integration and Test Plan

There were certain aspects of the Integration and Test Plan documentation that did have a positive effect, though the test plan went mostly unused. We made effort to follow the schedule laid out in our test plan, and that made it much easier to stay on track to finish in time.

Our unit test cases went largely unused during integration, and we did not compare expected outputs in any case. Though, our example cases were referenced for their logic in design.

Most of our testing during implementation was in the form of System Testing. We did reference our test cases for preconditions, and postconditions of certain classes, such as the Street Tile class.

5. Real vs Planned Team Structure

5.1 Our original proposed team structure

Scott Barnett
- Customer Side Documentation
- Non-Combat System Design
- Quality Control

Jimmy Flemming
- Combat Systems
- Documentation Diagrams

Liam MacKinnon
- Client Side Liaison
- Meeting Minutes
- Non-Combat System Design

Martin Main
- Project Lead
- Documentation Editor
- Webmaster

5.2 Differences

- All four of us collaborated for our Customer Side Documents
- Liam played more of a quality control role
- Documentation diagrams were originally lead by Jimmy but it became more of a team effort in later documents
- Liam was involved in the combat system design
- Martin played the role of lead developer
- Jimmy was the Supplier Liaison

5.3 Similarities

- Scott and Liam did most of the non-combat design
- Jimmy did most of the combat design
- Liam remained the client side liaison
- Liam remained the minute taker for our meetings
- Martin remained the project lead, editor, and webmaster

6. Lessons Learned

- The time committed to preliminary design can pay off when implementation begins
- Document decisions made to remove ambiguity during implementation.
- Keep track of requirements at each step throughout the process. Always check and recheck requirements.
    - It’s easier to spend more time on requirements in the beginning than to backtrack to complete them.
- Prioritize requirements so that more integral factors of the game are finished before less important aspects.
- Learn your team members schedules. This is something that our group did quite well. We had a good understanding of each other's regular general schedules so that we could plan weekly meetings in the same time slot.
- Have a plan for your meetings. It wastes time if a large portion of a meeting is just figuring out what needs to be done, instead of delegating.
- There’s nothing wrong with cold pizza, and sometimes it’s even better than microwaving it.

7. Suggestions for Future Study

- The project should be worth a larger portion of the final grade. As of now, it sits at 50% of the term, yet the project is the largest focus of the class. The final exam is still worth 35%, which is difficult, since we have little testing on theory prior to the exam.
- Fewer base requirements for the games, so that the games are easier to implement. It would seem quite difficult for any team to complete a fully working game in the timeframe, given the standard requirements that each group worked off of (Project Definitions - Basic Requirements from Plato). Perhaps make it more clear to students that the Basic Requirements listed on the web page are only a guideline, not necessary requirements to be used, as most teams seemed to treat them.
- Introduce the class to some useful libraries, or languages for implementation. It was very difficult for our team to get over the initial learning curve of creating a video game, since none of us had any experience in this type of development. Some basic hints of a good direction to go in, or libraries to use would end up shaving a lot of time off of our implementation.

8. Real World Hypothetical Differences

- It would have been advantageous in a real world scenario to have someone involved who has a background in game design
- More effort into advertising
    - If this product needed to be marketed to user in the real world, there would have been more effort put into the aesthetic of the project and into advertising
- Relations with our customer and supplier would have been a concern in a real world scenario. Since this scenario involved us dealing only with people who have a technical background, our relations with customer and supplier were much easier

9. Conclusion

This project was an excellent experience overall, and we feel that it has better prepared us for going to work on real world projects in the future. We have a much stronger grasp on the concepts presented, including requirements analysis, design, testing, implementation, and documentation. The biggest takeaways included learning how to work well as a team, learning how to design a product well, and learning how to manage time as a group to successfully implement a final product. Though we didn't finish every requirement listed, we feel like the project was a satisfying product of our efforts, and we are proud with the work we produced. We hope the customer has felt a reasonably level of satisfaction with the product, and feel that we developed a piece of software up to the standards that were expected.