System Requirements Specifications: Volere
SFWR ENG 3XA3: Revision 0

Cole Blanchard
Ratna Emani
Akshay Mantha
Group 9

December 7, 2015

Contents

List of Tables and Figures

Revision History

Developer	Date	Change	Revision Number
Ratna Emani	Dec 6 2015	Formatting, Use Case Diagram	3
Cole Blanchard	October 10 2015	Functional Requirements, Migration to the New Product, Risks, Costs, User Documentations and Training, Waiting Room, Ideas for Solution	3
Ratna Emani	October 10 2015	Project Drivers, Project Constraints, New Problems, Tasks	2
Akshay Mantha	October 9 2015	Non Functional Requirements, Open Issues, Off the Shelf Solutions	1

1 Project Drivers

1.1 The Purpose of the Project

This project aims to fix and complete the unfished puzzle game Zop. The project was found on GitHub (https://github.com/Zolmeister/Zop). The original creator of the game wanted to create a colored-tile matching game. In its current state, Zop is not playable, it is riddled with many bugs and problems left by the creator. As developers we took the initiative to fix and finish this incomplete project.

1.2 Background of the Project Effort

1.2.1 Content

The scope of this project lies between gamers that are interested in independently developed games (indie games) and people that enjoy quick gaming (i.e. arcade games). The scope has the potential to reach greater audiences in the future and can be hosted on sites like Miniclip or Steam. Furthermore we hope to engage other interested and freelancing game developers to make their own changes and further expand the game. Although originally programed using coffee script. This application will be developed on Python using Pygame.

1.2.2 Motivation

Many simple games developed on Android, or iOS gain great popularity over a short period of time. ~~So much so, that they are being redeveloped to be enjoyed on a desktop experience.~~ We believe such easy to understand game would be simple to implement. Therefore we decided to use this opportunity to create a enriched learning experience for new game developers using Python/Pygame. Games similar to Candy Crush and Bejeweled are liked by people of all ages and become very addictive. The motivation for this project arises from the need of more games like Zop and to encourage new developers to use python. Playing games is a lot of fun, but getting the chance to build our own boundaries and create our own rules, makes it more interesting.

1.2.3 Form

Although not considered a serious issue, there is a great development opportunity here. Our project is an academic application which allows us to explore the work without the consequences of a real world problem.

1.3 Goals of the Project

1.3.1 Content

The main goal of this project is to recreate the original game without the bugs and problems. Reprogram the game using modularization and the Model-View-Controller (MVC) method. Also document the process thoroughly for other developers to change and expand for future development.

1.3.2 Motivation

The motivation for the project is to make changes to the user interface at any time without having to redistribute the application.

1.3.3 Example

When the user makes a move by selecting any combinations of horizontal and vertical pieces of the same colour.

The selected pieces should disappear and the pieces above them must move down by the missing length.

1.3.4 Example

Purpose: to provide a functional and modified game Advantage: room for future expansions and development Measurement: Can be traced by checking if the state of the game board updates correct and cohesively.

2 The Stakeholders

The stakeholders of the project mainly include the group members, group supervisors, and the original creator of Zop.

2.1 Developers

2.1.1 Content

We as a group represent the development team for this game.

2.1.2 Motivation

Our motivation in building this project is to provide the gaming community a functional game with ability to expand.

2.1.3 Form

We used GitHub to look for an open source project that is open for other developers to tweak and modify.

2.2 Client

2.2.1 Content

The client for this project would be our supervisor Dr. Spencer Smith.

2.2.2 Motivation

The clients motivation is to purposefully reconstruct the project with the right documentation and programming etiquettes to better develop the growth of the program.

2.2.3 Form

The client provided us with a variety of open source and crowd sponsored projects with open libraries.

2.3 Customer

2.3.1 Content

The customers can be any users that want to play the game and developers who want to modify and add their own content.

2.3.2 Motivation

The costumers motivation can be a wide variety of reasons. With players who enjoy quick games would indulge the game for its entertainment purposes. While other developers can use this as a programming exercise and apply their creative alterations to the existing project.

2.3.3 Form

The customer can experience the project as a playable game and as a tool for learning the mechanics of simple game design.

3 Project Constraints

3.1 Mandated Constraints

Underdevelopment is the greatest constraint that our team will face. The original developer did not document any of their choices and actions, and they failed to comment the code with

proper protocol. Trying to extract the finer details of the game function and mechanics will be challenging. Due to the lack of comments and description of the functions and methods, not all code can be successfully re-purposed. This can impact the final product of the game by making it vary is some aspects from the original intended content.

3.2 Constraint Solution

The simplest constraint solutions may include:

- Repurposed code from other similar open source projects
- Complete redevelopment of some functions and methods
- Modularization of previous and new code

3.3 Implementation Environment of the Current System

The environment in which the project will be developed in is Python. The original content is programmed using CoffeeScript, JavaScript and HTML. It was available for public use hosted by GitHub. However the implementation environment the development team is most comfortable with is Python 3.5.0. Using Pygame 1.9.1, a set of modules designed for writing games, we will recreate the graphical user interphase.

3.3.1 Implementation Environment of the Current System

The environment in which the project will be developed in is Python. The original content is programmed using CoffeeScript, JavaScript and HTML. It was available for public use hosted by GitHub. However the implementation environment the development team is most comfortable with is Python 3.5.0. Using Pygame 1.9.1, a set of modules designed for writing games, we will recreate the graphical user interphase.

3.4 Naming Conventions and Terminology

There will a use of acronyms and terminology related to the project and the tools being used in the development process. More terms will be added to this list as we explore and learn new technologies to improve our project.

3.5 Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

CoffeeScript: is a programming language that transcomplies into JavaScript.
JavaScript: is a programming language for HTML and web development
HTML: HyperText Markup Language is a standard markup language for web pages
Zop: the name of the puzzle game designed by the developer Zolmeister

GitHub: a Web-based Git repository hosting service for source code management
GitLab: a web-based Git repository manager with code review and issue tracking features.
Python: A high level programming language
Pygame: cross-platform set of Python modules designed for writing video games

3.6 Relevant Facts and Assumptions

The biggest assumption that we as developers have to make is that all the users have/are familiar with using Python. Since users must run the main.py to play the game.

3.6.1 Relevant Facts

We will be using the Zolmeisters game Zop as our starting foundation and base our redevelopment off the original program.

3.6.2 Business Rules

The business rules will be adjusted accordingly as the client gives us more requirement details.

3.6.3 Assumptions

It is assumed that the final product created by our development team will differ a lot from the original content. The goal is to make a replica as close to the original game, however due to development constraints and modifications, there will be differences.

4 Functional Requirements

4.1 The Scope of the Work

Currently, Zop is located on a website in which users can simply just go to https://zop.zolmeister.com/ and start up a game. However, we plan to make the game downloadable with some constraints being that the user needs to have python downloaded in order to run and play the game.

Event Name	Input and Output	Summary
1. user starts	mouse click start button (IN)	user starts new game
2. user restarts	mouse click play again button (IN)	user starts new game (after game ends)
3. user plays again	mouse click play again button (IN)	user starts new game (after game ends)
4. high scores	mouse clicks high scores button(IN). top 3 list of high scores achieved (OUT)	user receives a list of high scores
5. select game pieces	mouse clicks one game piece and drags to others(IN). game score updates with the number of pieces selected (OUT)	pieces that user drags over are highlighted

table 1

4.2 Scope of the Product

1. Product Use Case Name: Start Game
   Trigger: User runs the game
   Preconditions: The game has not already started, this is the first event that should come up when the game first runs.
   Interested Stakeholders: User, teacher, teaching assistants, Zop Creator
   Actors: User, Game Client
   Outcome: A new game of Zop starts

2. Product Use Case Name: Restart Game
   Trigger: The game has started
   Preconditions: The game has to be in progress, not before or after the game has started or ended.
   Interested Stakeholders: User, teacher, teaching assistants, Zop Creator
   Actors: User, Game Client
   Outcome: A new game of Zop starts

3. Product Use Case Name: Play Again
   Trigger: The game has ended
   Preconditions: User must have finished a game of Zop

Interested Stakeholders: User, teacher, teaching assistants, Zop Creator
Actors: User, Game Client
Outcome: A new game of Zop starts

4. Product Use Case Name: High Score List
   Trigger: The game has ended
   Preconditions: User must have finished a game of Zop
   Interested Stakeholders: User, teacher, teaching assistants, Zop Creator
   Actors: User, Game Client
   Outcome: Outputs a list of high scores

5. Product Use Case Name: Game Pieces Selected
   Trigger: user selects the pieces they want in game
   Preconditions: game is in progress, more than one piece is selected, only pieces of the same colour can be selected
   Interested Stakeholders: User, teacher, teaching assistants, Zop Creator
   Actors: User, Game Client
   Outcome: pieces the user has selected disappear, random new ones appear, and the score updates with the number of pieces the user selected.

4.3 Functional and Data Requirements

1. Description:The user must be able to start a game
   Rationale: Users will want to play the game
   Originator:Cole Blanchard
   Fit Criterion: Product successfully starts a new game
   Customer Satisfaction:5
   Customer Dissatisfaction:5
   Priority:High
   Conflicts:None
   Supporting Material:None
   History: Created October 9, 2015

2. Description: The user must be able to restart the game
   Rationale: Users may have a "bad game" and want to start a new game mid-game to try for a best score again
   Originator:Cole Blanchard
   Fit Criterion: Product successfully start a new game while a game is in progress.
   Customer Satisfaction:5
   Customer Dissatisfaction:5

Priority:High
Conflicts:None
Supporting Material:None
History: Created October 9, 2015

3. Description: The user must be able to play again
   Rationale: The user will want to play again after the game has ended
   Originator:Cole Blanchard
   Fit Criterion: Product successfully starts a new game after the game has ended
   Customer Satisfaction:5
   Customer Dissatisfaction:5
   Priority:High
   Conflicts:None
   Supporting Material:None
   History: Created October 9, 2015

4. Description: The user will want to see the high scores of previous attempts
   Rationale: The user will want to reach certain scores to be on the leaderboard
   Originator:Cole Blanchard
   Fit Criterion: Product successfully scores the previous 10 highest scores.
   Customer Satisfaction:5
   Customer Dissatisfaction:3
   Priority:High
   Conflicts:None
   Supporting Material:None
   History: Created October 9, 2015

5. Description: The product must display the rules of the game
   Rationale: The user will want to know how to play the game
   Originator:Cole Blanchard
   Fit Criterion: Product successfully displays the rules of Zop
   Customer Satisfaction:5
   Customer Dissatisfaction:5
   Priority:High
   Conflicts:None
   Supporting Material:None
   History: Created October 9, 2015

6. Description: The product must display the amount of time left in the game
   Rationale: The user will want to know how much time is left before the game ends

Originator:Cole Blanchard
Fit Criterion: The product display the time left while game is in progress
Customer Satisfaction:5
Customer Dissatisfaction:5
Priority:High
Conflicts:None
Supporting Material:None
History: Created October 9, 2015

7. Description: The product must allow the user to select game pieces
   Rationale: The user will want to play the game
   Originator:Cole Blanchard
   Fit Criterion: Product allow the user to select game pieces
   Customer Satisfaction:5
   Customer Dissatisfaction:5
   Priority:High
   Conflicts:None
   Supporting Material:None
   History: Created October 9, 2015

8. Description: The product must delete selected pieces of the same colour and generates new pieces
   Rationale: The user will want to play the game
   Originator:Cole Blanchard
   Fit Criterion: The product deletes the game pieces the user selected and generates new pieces in the same column the pieces were in
   Customer Satisfaction:5
   Customer Dissatisfaction:5
   Priority:High
   Conflicts:None
   Supporting Material:None
   History: Created October 9, 2015

9. Description: The product must be able to display the score
   Rationale: The user will want to see the score they achieved in game and after the game
   Originator:Cole Blanchard
   Fit Criterion: Product successfully displays the score for the user to see before and after the game
   Customer Satisfaction:5
   Customer Dissatisfaction:5

Priority:High
Conflicts:None
Supporting Material:None
History: Created October 9, 2015

5 Non Functional Requirements

5.1 Look and Feel Requirements

- Zop consists of coloured squares on a grid
- Zop will be similar to other puzzle games such as Candy Crush Saga and Bejeweled

5.2 Usability and Humanity Requirements

- Zop can be played by anyone who knows the basic functionality of a computer, including Internet connectivity and browsing
- Zop's web interface is personalized, although the game itself is centralized
- Zop is accessible to anyone who has Internet connectivity throughout the world
  - Zop is played by matching squares of like colour either horizontally or vertically
- To play Zop, the user does not need to possess any particular prerequisite knowledge
- The user will have to understand the matching aspect of Zop in order to play
  - Zop is played by matching squares of like colour either horizontally or vertically
    - The squares are adjacent to each other
    - At least two or more squares have to be matched at once in order to obtain points for a score

5.3 Performance Requirements

- One round of a Zop game should last for a minute on a personal computer with a decent Internet connection
- Playing time for one game of Zop should always be consistent
- Zop is available for playing 24 hours per day and 365 days per year
- Zop shall not have any glitches that hamper the gaming experience

- Zops servers shall accommodate a large number of people for playing at the same time (at times of heavy traffic)
- Zop shall permanently be on the Internet, once published

5.4 Operational and Environmental Requirements

- Zop will be used by users on personal computers in a climate controlled condition
- Zop can be used by users on smart devices in non-climate controlled condition as well
- Zop can be played on all major web browsers such as Firefox/ Chrome/ Safari
- Zop uses Coffee Script
  - Zop requires HTML
- Zop communicates with Third-Party Applications using secure TCP/ IP protocols
- Zops final demonstration will be during the week of November 30, 2015

5.5 Maintainability and Support Requirements

- The games source code will be available for examination or further enhancement
- Zop can be played on all major operating systems (i.e. Windows 7, 8, Mac OS, UNIX, and Linux)

5.6 Security Requirements

- No password shall be required to access/ play Zop
- Zop shall not send a users data to another party (achieved through secure communication networks)

5.7 Cultural Requirements

- Zop shall not use multimedia, which can offend the countries that make use of it
- Zop shall display a disclaimer that states similarity to any political or cultural symbol or figure is merely coincidental

5.8 Legal Requirements

- Zop shall comply with all federal and provincial refrigerator regulation laws, relevant refrigerator standards, and privacy acts.

6 Project Issues

6.1 Open Issues

There are no issues at the moment.

6.2 Off-the-Shelf Solutions

Other games that follow the same mechanics as Zop:

- candy crush
- bejeweled

6.3 New Problems

- implementing Zop into Python
- fix previous bugs/ glitches
- add missing requirements
- after receiving feedback for program, work on revisions.

6.4 Tasks

6.4.1 Effects on the Current Environment

making the program from a web based game to a system based game

6.4.2 Effects on the Installed Systems

Not applicable

6.4.3 Potential User Problems

Users will need to be familiar with python in order to run the game.

6.4.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Not applicable.

6.4.5 Follow-Up Problems

Not applicable.

6.5 Effects on the Current Environment

making the program from a web based game to a system based game

6.6 Effects on the Installed Systems

Not applicable

6.7 Potential User Problems

Users will need to be familiar with python in order to run the game.

6.7.1 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Not applicable.

6.7.2 Follow-Up Problems

Not applicable.

6.8 Migration to the New Product

The product will make a large transition from javascript/HTML to Python. This is a drastic move but, is in the best interest for us since we are comfortable with Python.

6.9 Risks

We risk losing consumers with the migration to python since they have to download the game. Also reimplementing the game in a new language is a large risk.

6.10 Costs

Not applicable to this project.

6.11 User Documentation and Training

There will be a manual supplied and the end of the term containing a problem statement, a requirements document, a design document, a test plan, a test report, a user's guide, and the product source code.

6.12 Waiting Room

Potential upgrades for the game may be wanted for the future. These include levels and powerups.

6.13 Ideas for Solutions

Solutions to get around our risks is creating a fully functioning game to attract the users we may lose to actually download and play the game.

7 Use Case Diagram