SE 3XA3: Software Requirements Documentation Revision1 Project: Flight Shooting Game

Team 15, CLOUD10
Yijun Chen cheny161
Tianxing Li lit20
Zefeng Wang wangz217

October 5, 2018

Contents

List of Tables

List of Figures

Table 1: **Revision History**

DATE	DEVELOPER	CHANGE	REVISION
October 5, 2018	Yijun Chen	Initial Draft	0
October 5, 2018	Tianxing Li	Initial Draft	0
October 5, 2018	Zefeng Wang	Initial Draft	0
Decemeber 4, 2018	Yijun Chen	Revision	1
December 4, 2018	Zefeng Wang	Revision	1

1 Project Drivers

1.1 The Purpose of the Project

1.1.1 The background of the project

~~With the rapid development in the mobile technology, most Mobile Game Development Companies are competing with each other through 3D, 4D, and high-quality graphics games, while 8-bits graphic-style games are getting not popular in supply.~~With the rapid development in the technology, most Game Development Companies are competing with each other through 3D, 4D, and high-quality graphics games, while 8-bits graphic-style games are getting not popular in supply. However, in the market, there is a huge portion of the users who want to play the 8-bits games. Majority of them are born in the 80s or 90s. They think the retro games evoke their childhood nostalgia. [2] Pixel art games aren't retro, they're the future [1], says by the verge magazine, which also believes no matter however advance computer design software becomes, there always be gamers thirsty for the retro games. Our motivation to make this project is to bring back this classic element to the mobile market once again. The project will be based on the original Flight Shooting Game. With adding Pyxel library, the whole game is redeveloped and becomes 8-bits graphic-style.

1.1.2 Goal of the project

Our project is an entertainment-driven experience for users who desire to review the classic. The first goal is creating a python game similar to the original flight shooting game using Pyxel library which transfers the game to the 8-bits style. Secondly, designing new features and interface Thirdly, documenting all processes during the creation for our clients to view. In that case, they will have a brief introduction to the game and know how to play when they access the product.

1.2 The Stakeholders

1.2.1 The Client

The client of our product is an external entity who plays an important role as final reviewer of the project. They are interested in bringing the 8-bits

style game back to the market.

1.2.2 The Customers

The customer of our product is the general public who play games. ~~The archetypal customer would be mobile phone users who think retro games are memory of the childhood, and are able to gain pleasure from the product.~~The archetypal customer would be gamers who think retro games are memory of the childhood, and are able to gain pleasure from the product.

1.2.3 Other Stakeholders

General public – These people will be our consumer base. ~~Anyone from the public can access to our product through their mobile phone and play the game by reading its intuitive instructions.~~Anyone from the public can access to our product through their computers and play the game by reading its intuitive instructions. Due to our project is a redevelop project from the Github, it shall share the same general stakeholders as the original one.

1.2.4 Priorities Assigned to Users

Key Users

The Client
The Customer

Secondary Users

The General Public

1.3 Mandated Constraints

1.3.1 Solution Constraints

Description: ~~The product shall operate on different mobile devices~~The product shall operate on different devices with Python 3 and Pyxel library installed.

Rationale: ~~The client will be preferably using a smart phone with touching screen~~The client will be preferably using an device with Python 3 and Pyxel library installed.

Fit criterion: ~~Our client will be any one with a mobile devices~~Our client will be any one with a PC. They need to install Python 3 and Pyxel library to play the game.

Description: The product shall operate without any delay.

Rationale: The graphics and images in our product should be small, and the algorithm should have the smallest time complexity(O).

Fit criterion: The product shall be approved by running the speed tests.

1.3.2 Off_the_Shelf Software

~~An Android Operating System is all the user needs to run our product.~~A computer installed with Python 3 and Pyxel library is all the user needs to run our product.

1.3.3 Anticipated Workspace Environment

~~The anticipated workspace environment for the product is on any mobile devices~~The anticipated workspace environment for the product is everywhere, as long as the user can have an device with Python 3 and Pyxel library installed.

1.3.4 Schedule Constraints

The schedule constraint is set by the client. A deadline has been set to the week of ~~November 27, 2018~~December 5th, 2018.

1.3.5 Budget Constraints

There is no budget available to us.

1.4 Naming Conventions and Terminology

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

Table 2: **Definitions**

ACRONYM/ ABBREVIATION	INTENDED MEANING
Client	The group that the product is being developed for
Customer	Any individual that utilizes the product after completion
The program	Any individual that utilizes the product after completion
WTF	Write to File
eng	engineer
txt	Text file
git	a version control system
apk	Android Application Package
Google Doc	Google Documentation
overleaf	An online LaTex editor
Pyxel	A retro game engine for Python
Github	An online web-based Git-repository manager
MIT license	A permissive free software license

1.5 Relevant Facts and Assumptions

1.5.1 Relevant Facts

The relevant facts are that there were ~~one thousand lines~~one thousand and five hundred lines of code in the original game, and the game has a MIT license on github.

1.5.2 Business Rules

Everyone one in the team should have the same amount of the work, and the project is owned by the team.

1.5.3 Assumptions

~~Some assumptions for the project are that all the images and audios used in the project are free from the internet, the web-hosting server is free for launch and the written code are encapsulated so that the code would bot be used for personal interests.~~Some assumptions for the project are that images used in the project are free from the internet, the written code are encapsulated so that the code would bot be used for personal interests.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

The work, in the form of a project, is going to be completed in a period of three month by a team of three members and is composed of project code implementation and project documentation. The work of the project is essentially a re-development of an existing project and the core of it is to make it even more advanced by fixing mistakes and adding new features. The files of the work is added and saved on GitLab. The three members are of the role of maintainers.

2.1.2 Work Partitioning

Table 3: **Work Partitioning**

Event Number	Event Name	Input	Output
1	Game Creation	Developer code	An runnable game
2	Game Audio	Audio file	Sound Effect
3	Player Plane	Graphics and Developer Code	Graphic Animation
4	Game Collisions	Developer Code	An runnable game
5	Final Improvements	Developer Code	An runnable game

2.1.3 Individual Product Use Cases

- ~~The user plays the game in endless mode.~~The user starts the game by pressing "space". Then the game goes from the start page to the game page. User controls the flight by pressing "wasd" or directions keys on keyboard.

- ~~The user plays the game in stage mode.~~User presses "u" to use ultimate when the ultimate condition is full

Scenario:
a) A user downloads the game file and installs Python 3 and Pyxel library on his/her device (usually a computer).
b) ~~A user runs the file and plays the game.~~A user starts the game by pressing "space". Then user controls the flight to collide with the boss
d) ~~Game over because of the exhaustion of life point.~~A user starts the game by pressing "space". Then user controls the flight to collide with the boss. The game over because of the exhaustion of life points.
e) ~~The user exits the game.~~A user starts the game by pressing "space". Then user controls the flight to collide with the boss. The game over because of the exhaustion of life points. Then user presses "r" to restart the game.

2.2 Functional Requirements

- FR1

  ~~The python file should create a window when the project is still in progress~~A window should be created while the game is still in progress, i.e. the game should be playable on computer.

  Fit Criterion or Test Case:
  ~~Is a kivy window created by executing the python file?~~Run the game file on different devices with the same environment for several times

- FR2

  ~~The python file should be executable with Pyxel library.~~The main game file should be executable

Fit Criterion or Test Case:
~~The file is executable, it does something.~~ Crash is stricted.

- FR3

  ~~The game should have a welcome stage before the game begins.~~The game should have a start stage before the game begins.

  Fit Criterion or Test Case:
  Without giving any input, the game is not started, ~~and the window stay unchanged.~~and the game remains at start page.

- FR4

  ~~The welcome page should include buttons for mode options.~~The start page should come with a retro music, and accept "space" from user to start the game

  Fit Criterion or Test Case:
  ~~Once the file is executed, a welcome page with buttons shows up.~~Try input "space" and other keys when the game is at start page, and check whether the game gives the anticipated response

- FR5

  After the game starts, the plane should shoot automatically. Fit Criterion or Test Case:

  a) Is the plane shooting without giving any input?
  b) Is the plane shooting when it is manipulated (or give input)?

- FR6

  ~~If the plane has a collision with obstacle objects, i.e. enemy planes, bullets from enemy planes, the game terminates.~~If the plane collides with enemy objects such as enemy planes, enemy bullets, player must lose health points.

  Fit Criterion or Test Case:

~~Give a collision test by set the value of "collision" to True to see whether the game terminates or not.~~Control the flight to collide with each of the enemy objects ten times, and record the results

- FR7

  ~~The player's plane should respond to user's input.~~The player's plane should respond to "wasd" or direction keys on keyboard for direction control

  Fit Criterion or Test Case:
  ~~Set the value of "direction" and "speed" to some value within the constraints, check whether the plane acts correctly.~~Press those keys and any combination of them each for ten times, and see if whether the flight is moving in the anticipated direction.

- FR8

  ~~Enemy plane should be eliminated if it is hit by the player's bullet(s).~~Enemy flights should all move downwards from the top of the screen in a same pattern.

  Fit Criterion or Test Case:
  ~~Set the value of "life" of an enemy plane to False to check whether that plane is eliminated or not.~~Play the game in developer mode which is a mode to set health to maximum for developing convenience. Look and record the enemy flights behaviour for 5 minutes.

- FR9

  ~~In endless mode, the game should not terminate until the player's plane has a collision with an obstacle object.~~The game should go to end page with an ending retro music when the player's health point is zero.

  Fit Criterion or Test Case:
  ~~Set the number of enemy to 0 to check whether the game is running consistently before force it to stop.~~Control the flight to collide with enemy bullets. Record the result when the player's health point goes to zero, and then test again for 10 times.

- FR10

  ~~In stage mode, the game should terminates in two cases:~~
  ~~a) The player's plane has a collision with an obstacle object.~~
  ~~b) The stage is cleared.~~Boss1(The Yellow Head) should show up and shoot automatically towards the player when the player score increases 5000 every time. Similarly, Boss2(Chramp) should show up every 12000 points.

  Fit Criterion or Test Case:
  ~~a) is passed because it is the same as one stated before.~~
  ~~For b), set the value of stage length to zero to check whether the game terminates as soon as it starts.~~Play the game in developer mode which is a mode to set health to maximum for developing convenience. Record the Bosses' behaviour for 10 times and check if whether the behaviours are as anticipated.

3 Non-functional Requirements

3.1 Look and Feel Requirements

- ~~NF1~~
  ~~The icons of three types of enemy jets should clearly represent their difficulty levels of beginner, intermediate , advanced by different sizes and patterns.~~

- NF1
  The colour of background scene should not be more saturated than the colour of jets. Also, it should be in contrasting colour from jetss colour.

- NF2
  The pattern should not appear too much in the background. Players need to be able to see their jet and incoming enemy jet clearly from the scene.

- NF3
  Background scenes and jets should be in pixel style.

- ~~NF5~~
  ~~The background scenes and jets should not be complicated as pixel style is the theme of this product.~~(deleted due to duplication with the former one)

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirements:

- NF4
  The app should be applicable for player using any device with Python 3 and pyxel library installed.

- NF5
  The game should be simple for a player aged 12-65 in able condition to understand.

3.2.2 Personalization and Internationalization Requirements:

- Not applicable

3.2.3 Learning Requirement:

- NF6
  The player shall be able to read the game rules written in English provided in the User Guide

- NF7
  The game rules should be written in short and clear logic form and it should provide visual aid.

3.2.4 Understandability and Politeness Requirements:

- NF8
  ~~The player shall be able to read the game rules written in English.~~The rule, purpose of the game should be easy for users to understand. The content, especially textual information, should be in a simple style.

3.2.5 Accessibility Requirements:

- NF9
  The game shall be able to be accessed and executed on the majority of ~~Android smart phone devices~~ computers with python3 installed.

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements:

- NF10
  The app must response to any input from user within 0.1 second when the game is not started yet.

- NF11
  When the game has started, the response to a user input should not have latency that can be felt by user.

- ~~NF12~~
  ~~Generate pop-up message indicating the status of app(ie, loading) if result can not be generated immediately.~~

3.3.2 Safety-Critical Requirements:

- NF12
  The product should not collect data from user or device.

3.3.3 Precision or Accuracy Requirements:

- NF13
  Game score and other numerical information that will be shown to player are ~~all integer values~~ in the appropriate format.

3.3.4 Reliability and Availability:

- NF14
  The game should be fully functioning without internet connection after the necessary content is installed successfully.

3.3.5 Robustness or Fault-Tolerance Requirements:

- NF15
  The game requires very limited modification from player and should be operate without consideration of the physical state of end users.

3.3.6 Capacity Requirements:

- NF16
  The game should not overuse device storage space when running.
- NF17
  The size of stored data and running log should be limited.

3.3.7 Scalability or Extensibility Requirements:

- NF18
  The code will allow for scalability.

3.3.8 Longevity Requirements:

- NF19
  The product shall be able to be executed for the lifetime of a regular program.

3.4 Operational and Environmental Requirements

3.4.1 Expected Physical Environment:

- NF20
  The product is intended to be used on device with Python 3 and pyxel library installed.

3.5 Maintainability and Support Requirements

- NF21
  Maintenance is conducted upon user's feedback and update on Python and pyxel library version.

3.6 Security Requirements

- Nf22
  The product should not access user's personal data.

- NF23
  The only data generated will be ~~game record and log~~the file storing the high score.

3.7 Cultural Requirements

- NF24
  The product shall show a disclaimer explaining that any similarities to any cultural or political symbol or figure is coincidental.

3.8 Legal Requirements

- NF25
  The product shall comply with all national and federal laws.

- NF26
  The product shall comply with all relevant software standards.

- NF27
  The product shall comply with all relevant privacy acts.

3.9 Health and Safety Requirements

- Not applicable.

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

4 Project Issues

4.1 Open Issues

N/A

4.2 Off-the-Shelf Solutions

4.2.1 Reusable Components

Modularized code components

4.2.2 Products that can be copied

As an open source project with MIT license, it is allowed to be copied and redeveloped.

4.3 New Problems

4.3.1 Effects on the Current Environment

~~The updates of Android system may cause the failure of the game.~~
~~The product mat not be compatible on all Android-based operation systems.~~
The updates of Python version and Pyxel library may cause the failure of the game.

4.4 Tasks

4.4.1 Project Planning

This is a link to our GanttProject file.

4.5 Migration to the New Product

4.5.1 Requirements for Migration to the New Product

None.

4.5.2 Data That Has to Be Modified or Translated for the New System

None

4.6 Risks

N/A

4.7 Costs

potential cost: image editing software

4.8 User Documentation and Training

Not applicable

4.9 Waiting Room

- further functions and features
- Audio effects

4.10 Ideas for Solutions

- Proper hierarchy
- Documentation of python code

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

References

[1] Bob Biber. Logic simplified, Oct 2018.

[2] Sam Byford. Pixel art games aren't retro, they're the future, Sept 2018.