SE 3XA3: Software Requirements Specification MineCraft

Group 307, 3 Craftsmen
Hongqing Cao 400053625
Sida Wang 400072157
Weidong Yang 400065354

February 10, 2020

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Jan 27	1.0	Team information
Date 2	1.1	Notes

This document describes the requirements for MineCraft. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (**?**).

1 Project Drivers

1.1 The Purpose of the Project

1.1.1 The User Business or Background of the Project Effort

In the current gaming market, the age limitation on video games builds boundaries between children and adults. Most 3D games available in the market are not available to children since they contain violent content. It is expected to see more video games that are available and beneficial to children and teenagers.

1.1.2 Goals of the Project

We want to give a solution to provide an interesting and inspiring video game for children to unleash their creativity.

1.2 The Stakeholders

1.2.1 The Client

Since this project is considered as a reimplementation of an existing open-sourced software game, it cannot be invested and be used for financial purposes. Therefore, the client of this project will only be Dr.Bokhari at the current stage. For further development, with the authorization from the original software designer, this software game project is expected to be deployed on PC gaming service agencies such as Steam. By that time, the client will include gaming service agencies.

1.2.2 The Customers

In general, the customers of this project will be PC gamers who have a passion for Sandbox games. Due to the simplicity and user-friendliness, the main customer group specifically refers to young gamers aged 7-13.

1.2.3   Other Stakeholders

Other stakeholders aside from the clients and customers are listed below:

- The software development team, including Sida Wang, Hongqing Cao, and Weidong Yang

- Legal experts in the domain of youth game design

- Usability experts

- Software game testers

1.3   Mandated Constraints

The project should be designed and developed conforming with the following constraints:

1.3.1   Solution Constraints

- **Description:** The product shall operate using either Windows or Linux OS.
  Rationale: The user uses either Windows or Linux OS.
  Fit Criterion: The executable file shall be approved running on either Windows or Linux OS by the development team.

1.3.2   Implementation Environment of the Current System

- The processor of the hardware system where the product is to be installed must be capable to run a simple 3D game.

- The product is developed using Python, so the development environment must have Python installed.

- The executable file of the product will be independent of Python, which means the OS that runs the game will not necessarily have Python installed.

1.3.3   Partner or Collaborative Applications

- The GUI of the game will be generated by Pyglet. Therefore, the development environment will require Pyglet package installed.

- The executable file will be generated by Pyinstaller. Hence, the development environment will require Pyinstaller package installed too.

1.3.4   Off-the-Shelf Software

- The development of this product shall follow and refine the original design, which is Fogleman's Minecraft software program[1].

1.3.5   Anticipated Workplace Environment

- The sound experience will be negatively affected if the workplace is noisy.

1.3.6   Schedule Constraints

- The project must be completed within a 3-month period starting from Febuary to April.

1.3.7   Budget Constraints

- N/A

1.4   Naming Conventions and Terminology

The naming conventions and terminology section will aid readers from different backgrounds to clearly understand the content of this document. The naming conventions and terminologies used in this document are listed below:

- **OS:** Operating System

- **GUI:** Graphical User Interface, which allows the user to interact and visualize the program by graphics instead of text

- **PC Gamers:** The individuals who play games either on a desktop or on a laptop

- **Sandbox Games:** A type of games that allows player to create, modify, and destroy the environment

- **Python:** The programming language that is used for the development of this project

- **Pyglet:** The Python library for the design of graphical user interface

- **Pyinstaller:** The tool for containing a Python application and generating an executable file to that application.

- **Player/Gamer:** The person who controls the PC to play the game

- **Character:** The fictional character who is controllable by the player/gamer.

- **3D Game:** A game in three dimensions.

1.5   Relevant Facts and Assumptions

1.5.1   Facts

- The original project is 902 lines of Python code.

- The project is built with Python and its GUI library Pyglet.

- Tests feedback will come from the assigned game testers.

- This is not a profit-oriented project unless the financial purpose is approved and authorized by the original project owner.

- The executable file will be built by Pyinstaller and requires up to 2GB PC memory.

1.5.2   Assumptions

- To compile the source code in the terminal, Python 3.7 and Pyglet package will be required to be installed on the OS.

- To open the game by clicking the executable file icon, no specific software is required to be installed.

- The executable file generated by Pyinstaller will keep all the features from the compiled application.

- The Pyglet library will provide sufficient functionalities for the reimplementation.

- **User Characteristics:**

- The user is capable to play 3D video games with no dizziness.

- The user is passionate about 3D video games.

- The user has a PC that can run a simple 3D game.

- The user has knowledge of Python compilation and OS operations in the case of running the program in the terminal.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work
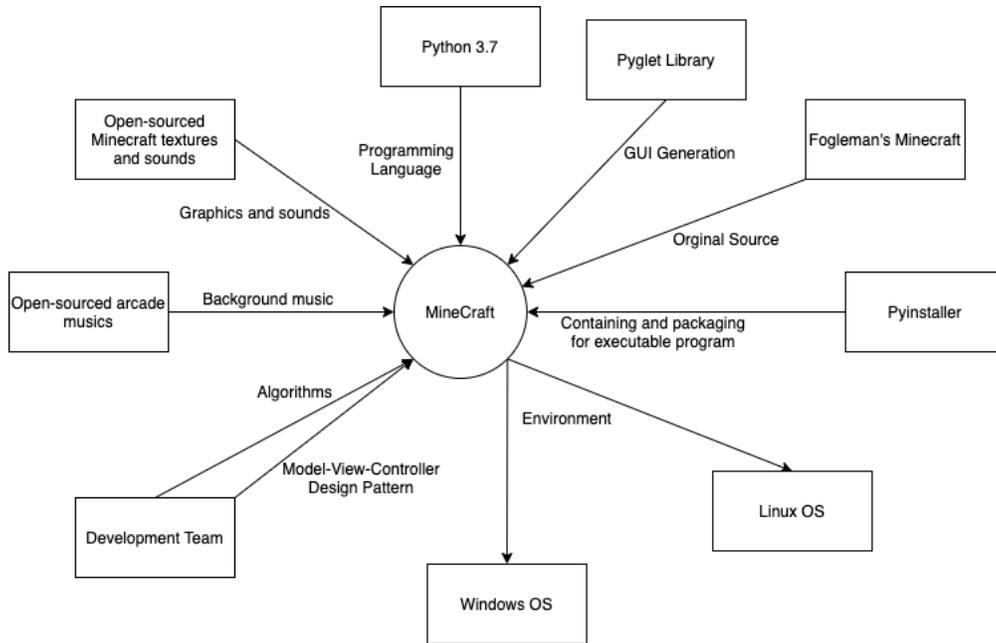
Figure 1: Context Diagram

2.1.2 Work Partitioning

Event Name	Input and Output	Summary
Player performs action on mouse/keyboard	Player action(in) GUI updates(out)	The system modifies the game state and updates the GUI

2.1.3 Individual Product Use Cases

Since the game has Nonlinear Gameplay Style and different players may have different ideas, it is not possible to determine all use cases from the user actions. The main use case of the player is to control the character and explore the game world.

2.2 Functional Requirements

- **FR1:** The software game shall start(open the GUI)when the player clicks on the icon of the game.

- **FR2:** The software game shall load the scene of the game on the GUI when the game starts.

- **FR3:** The software game shall load the character of the game on the GUI and place it on the ground when the game starts.

- **FR4:** The GUI shall place a crosshair at the center when the game starts.

- **FR4.1:** The crosshair shall be kept at the center of the GUI from the start of the game to the end.

- **FR5:** The software game shall move the character when the player press "W", "A", "S", and "D" keys on the keyboard.

- **FR5.1:** The software game shall move the character forward when the player presses the "W" key on the keyboard.

- **FR5.2:** The software game shall move the character to the left when the player presses the "A" key on the keyboard.

- **FR5.3:** The software game shall move the character backward when the player presses the "S" key on the keyboard.

- **FR5.4:** The software game shall move the character to the right when the player presses the "D" key on the keyboard.

- **FR6:** The software game shall change the direction of the character according to the mouse movement controlled by the player.

- **FR7:** The software game shall let the character jump when the player presses the space key on the keyboard.

- **FR8:** The software game shall toggle the flying mode when the player presses the tab key on the keyboard.

- **FR8.1:** The software game shall enable the flying mode when the flying mode is off and the player presses the tab key on the keyboard.

- **FR8.2:** The software game shall disable the flying mode when the flying mode is on and the player presses the tab key on the keyboard.

- **FR8.3:** The software game shall allow the player to control the character to fly using "W" and "S" keys when the flying mode is on.

- **FR9:** The GUI shall show the outline of a block when the crosshair is pointing to the block.

- **FR10:** The software game shall remove a block when the player moves the mouse to point the crosshair to the block and left-clicks on it.

- **FR11:** The software game shall place a new block next to or on the top of an existing block when the player moves the mouse to point the crosshair to the existing block and right-clicks on it.

- **FR11.1:** The type of the block to be built shall be changed to the brick type when the player presses "1" on the keyboard.

- **FR11.2:** The type of the block to be built shall be changed to the grass type when the player presses "2" on the keyboard.

- **FR11.2:** The type of the block to be built shall be changed to the sand type when the player presses "3" on the keyboard.

- **FR12:** The software game shall release the mouse movement from the GUI(stop the character direction changing) when the player presses "ESC" on the keyboard.

- **FR13:** The software game shall quit(close the GUI) when the player clicks on the close button("X") on the GUI.

- **FR14:** The software game shall play the background music when the game starts.

- **FR15:** The software game shall play the effect sound when a block is being added or removed.

- **FR16:** The player shall not be able to move through blocks.

- **FR17:** The Stone block shall not be removed or added.

3 Non-functional Requirements

3.1 Look and Feel Requirements

3.1.1 Appearance Requirements

- **NFR1:** The software game shall be attractive to a teenage player.
  Fit Criterion: A survey of players asking for attractiveness and get the average as ATTRACT_AVG and it is above ATTRACTIVENESS.

3.1.2 Style Requirements

- **NFR2:** The software game shall give the player a feeling of playing Minecraft.
  Fit Criterion: A survey of players asking for similarity to Minecraft and get the average as SIMILAR_AVG and it is above SIMILAR.

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirements

- **NFR3:** Players aged MIN_AGE+ shall be able to play the game with no difficulty.
  Fit Criterion: A survey of players asking for difficulty and get the average as DIF_AVG and it is below DIF.

3.2.2 Personalization and Internationalization Requirements

- **N/A**

3.2.3 Learning Requirements

- **NFR4:** The learning curve of this game shall vary from LEARN_MIN to LEARN minutes depending on the age of the player.
  Fit Criterion: A survey of players asking for learning time and get the average as LEARN_AVG and it is below LEARN.

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

- **NFR5:** The response shall be fast enough to avoid interrupting the user's flow of thought.
  Fit Criterion: The software game shall respond in less than RESPONSE second for 99 percent of the interrogations. No response shall take longer than FALSE_RESPONSE second.

3.3.2 Precision or Accuracy Requirements

- **N/A**

3.3.3 Reliability and Availability Requirements

- **NFR6:** The software game should be available for play HOURS per day, DAYS per year.
  Fit Criterion: 100 times access tests at different time will be applied and the result should be 100 successful accesses.

- **NFR7:** The software game shall be able to keep running for a maximum of MAX_HOUR hours.
  Fit Criterion: A MAX_HOUR long time program duration test will be applied 3 times and the result should be all pass.

3.4 Operational and Environmental Requirements

3.4.1 Requirements for Interfacing with Adjacent Systems

- **NFR8:** The software game shall not perform negative actions on other programs running on the same system.
  Fit Criterion: The software game will be tested with different types of other software programs running and it will not bring negative effects to other programs.

3.5   Maintainability and Support Requirements

3.5.1   Maintenance Requirements

- **NFR9:** An update must be applied to the software game when a bug is revealed.
  Fit Criterion: N/A

3.5.2   Supportability Requirements

- **N/A**

3.5.3   Adaptability Requirements

- **NFR10:** The software game shall be able to run on Windows and Linux OS.
  Fit Criterion: The software game will be tested on both a Windows OS and a Linux OS and the test cases should all pass.

3.6   Security Requirements

3.6.1   Integrity Requirements

- **NFR11:** The software game shall protect itself from intentional abuse.
  Fit Criterion: Specific threat test cases will be made and the threats should be prevented.

3.6.2   Privacy Requirements

- **NFR12:** The software game shall delete all the data generated by the player when the player exits the game.
  Fit Criterion: Specific test cases will be applied and they should all pass.

3.7   Cultural Requirements

3.7.1   Cultural Requirements

- **NFR13:** The software game shall not be offensive to any religious or ethic groups.

Fit Criterion: A survey of players asking for cultural politeness and get the average as CUL_AVG and it is above CUL.

3.7.2 Political Requirements

- N/A

3.8 Legal Requirements

3.8.1 Compliance Requirements

- **NFR14:** The product shall not violate the Digital Millennium Copyright Act[1].
  Fit Criterion: A survey of a legal expert asking for standard meeting and the result is meeting.

3.9 Health and Safety Requirements

- **NFR15:** The product shall not generate any mental or physical threat to the safety of the players.
  Fit Criterion: A survey of a safety domain expert asking for meeting standard and the result is meeting.

4 Project Issues

4.1 Open Issues

Our investigation into whether Pyglet is the best fit GUI library for the design of this game product is still an open issue.

4.2 Off-the-Shelf Solutions

The original program designed and developed by Folgleman[2] will be considered as the candidate component for the development of this software game. Online open-source Minecraft textures and sounds will be considered as source elements to create graphics and sounds of the game.

4.3 New Problems

One potential problem of this software game could be that the player feels it is too similar to the well-known Minecraft game released by Microsoft. Another problem is that the further development for game features extension could be difficult due to the limitation of Pyglet.

4.4 Tasks

The project development plan can be found **HERE**.

4.5 Migration to the New Product

N/A

4.6 Risks

4.7 Costs

Since all the sources(including textures, sounds, original program) and tools(including Python, Pyglet, Pyinstaller) are open-source, there will be no monetary cost for the development. The time cost of the development will be 3 months.

4.8 User Documentation and Training

A user beginning manual will be added to the game window as a guideline of how to play the game. The beginning manual will include all the basic operations that the player could do. With the help of the manual, the player will be able to discover the game.

4.9 Waiting Room

The toolbox feature is still under consideration.

4.10 Ideas for Solutions

N/A

5 Appendix

5.1 References

- [1]Fogleman, \fogleman/Minecraft,"
    GitHub, 16-Feb-2019. [Online]. Available:
    https://github.com/fogleman/Minecraft. [Accessed:
    10-Feb-2020].

- [2]"Video Games and the law: Copyright,
    Trademark and Intellectual Property", NewMediaRights, 2020. [Online].
    Available: https://www.newmediarights.org/guide/legal/Video_Games_law_
    Copyright_Trademark_Intellectual_Property. [Accessed: 06- Feb-
    2020].

5.2 Symbolic Parameters

- **ATTRACT_AVG:** 0 to 10

- **ATTRACTIVENESS:** 7

- **SIMILAR_AVG:** 0 to 10

- **SIMILAR:** 7

- **DIF_AVG:** 0 to 10

- **DIF:** 3

- **LEARN_AVG:** 30 minutes

- **LEARN_MIN:** 2 minutes

- **RESPONSE:** 0.1 second

- **FALSE_RESPONSE:** 0.5 second

- **HOURS:** 24 hours

- **DAYS:** 365 days

- **MAX_HOUR:** 5 hours

- **CUL_AVG:** 0 to 10

- **CUL:** 7