SE 3XA3: Development Plan
CraftMaster

Group Number: 307
Group Name: 3 Craftsmen
Members:
Hongqing Cao 400053625
Sida Wang 400072157
Weidong Yang 400065354

Contents

List of Tables

List of Figures

Date	**Editor(s)**	**Change**
Jan 27	Hongqing	Added team information
Jan 30	All	Discussed about the content
Jan 30	Sida Wang	Finished the content of all parts according to the discussion
Feb 11	Hongqing	Renamed the project name
Mar 15	Sida	Finished the refinement for Revision 1
Mar 19	Sida	Fixed a minor error

Table 1: **Revision History**

1 Team Meeting Plan

Meetings will primarily be held twice a week in the lab room according to the lab schedule. Additional meetings will be held in Thode Library at any time when necessary.

- **Monday** 12:30 - 14:20, ITB 236
- **Thursday** 12:30 - 14:20, ITB 236
- Occasionally, Thode Library 1st Floor

1.1 Roles

Chair: Weidong Yang
Responsible for selecting topics and creating the meeting agenda.

Timekeeper: Hongqing Cao
Responsible for ensuring the meeting starts and ends on time and keeping the distribution of time on each topic balanced.

Notetaker: Sida Wang
Responsible for recording valuable information and summarizing the meetings.

1.2 Rules For Agenda

Pre-meeting:

- The chair will create an agenda sheet for a specific meeting including general topics, activities, and questions to be discussed. Other team members should preview the agenda sheet before the meeting.
- An item checklist may be produced by the chair when multiple topics are to be discussed.
- Any absence for meetings should be reported before the meeting starts.

During meeting:

- The timekeeper should ensure that the meeting starts and ends on time.
- The chair will record the attendance of each member at the beginning of the meeting.
- The meeting will start with a review of the agenda sheet. This will ensure that all team members are fully prepared.
- The meeting will end with a review of the meeting's effectiveness. Each team member should give at least one piece of advice on how to improve the effectiveness of the next meeting. The notetaker will summarize the meeting at the end.

2 Team Communication Plan

- **Facebook Messenger** group chat will be used for internal communications including inquiries, the share of resources, and discussion.
- **Gitlab** will be the main tool for communications of development changes and updates.

3 Team Member Roles

Team Member	Role(s)
Weidong Yang	Software Developer, Tester, Team Leader, Pyglet & Algorithm Expert
Hongqing Cao	Software Developer, Tester, LaTex & Documentation Expert
Sida Wang	Software Developer, Tester, Scribe, Git Project Manager

Table 2: **Team Member Roles**

4 Git Workflow Plan

4.1 Overview

The **Git Master and Feature Branch** will be used to manage the software development. One team member will create his/her local branch and work on it. This workflow allows different modules to be developed and tested in each team member's localized branch and do not cause conflicts. Sida will be responsible for the git master controls and will ensure all the files are completed and tagged before the deadline.

4.2 Millstones with Gitlab Tags

Milestones	Deadline	Tag
Requirements Document Draft	February 7	SRS-Rev.0
Proof of Concept Demonstration	February 13	N/A
Test Plan Draft	February 28	TP-Rev0
Revision 0 Design and Documentation	March 13	DD-Rev0
Revision 0 Demonstration	March 19	N/A
Revision 1 Demonstration	April 2	N/A
Revision 1 Design and Documentation	April 6	DD-Rev1

Table 3: **Millstones and Gitlab Tags**

5 Proof of Concept Demonstration Plan

5.1 Scope and Feasibility

The original project is implemented using Python within one module. To optimize modularity, our reimplementation will follow the software architecture MVC(Model, View, Controller) model. Since Python is an object-oriented programming language, the modular design is feasible using different classes to implement.

5.2 Potential Challenges and Risks

The hardest part of the reimplementation will be adding new categories of blocks. In the original project, all blocks are static and have no other interactions with the player besides being built or destroyed. In our expectation, new blocks with unique properties such as Lava, which will burn the player out, will be added to the game. These new types of blocks will require more complex interactions with the player, which is difficult to implement. To mitigate this challenge, we plan to model the game blocks as objects, and to program unique algorithms to apply to different blocks. Also since the original game is using the content of Minecraft, the customization of the reimplementation heavily depends on the texture resources from the internet. The most difficult part of testing is to test the interaction between the player and the world within the game. Similar to most 3D games, it is hard to mitigate the risk of bugs using traditional testing methods. To make the game stable and enjoyable, our testing team will dynamically test the game by spending a significant amount of time playing it.

5.3 Software Resources

The Pyglet package provides cross-platform windowing and multimedia library. With Pyglet, visually rich small games can be feasible to build. Pytest provides powerful unit testing and functional testing but does not fully support solutions to integration testing. All the libraries using by this project will be easily installed on either Windows or Linux machines. The game will be delivered as an executable file(generated by Pyinstaller) in order to optimize the portability.

5.4 Demonstration

The demonstration will be done on both Windows and Linux systems. On each machine, one team member will click on the icon of the executable game file and play the game. The player will travel the world for a certain distance and have some interactions between different types of blocks. The flying mode will also be activated to show its functionality. To overcome the risk of 3D bugs, edge case testings will be performed.

6 Technology

- **Programming Language:** Python
- **Graphical User Interface:** Pyglet
- **Testing Tools:** Pytest
- **Documentation Tools:** Doxygen, Latex
- **Version Control:** Gitlab
- **Other Tools:** Pyinstaller

7 Coding Style

The Coding Style is **PEP-8** for this project.

8 Project Schedule

The project schedule can be found **HERE**.

9 Project Review

9.1 Overview

In general, the development process of CraftMaster was successful. All the documentations and code implementations were completed before their deadlines. Team meetings went smoothly according to the meeting plan. Feedbacks were gathered from demonstrations. Refinements on both documentations and code implementations were proposed according to the feedbacks and were completed on time. The team believes that the development process of CraftMaster is a practical experience of software project management on Gitlab. By developing this project, all team members became more familiar with the software development process and git usage.

9.2 Open Issues

During the development process, there were some open issues and they are listed below:

- Due to the time constraint, the initially proposed feature "Dynamic Block Types" was not implemented and has been added to the "Anticipated Changes" section in the **Module Guide(MG)** for future implementation.

- The team find it is difficult to give hardware-related specifications, such as the content of “Hardware-Hiding Modules” in the **MG**. The reason is that Pyglet leaves the hardware design concern out. By using Pyglet, the design and development team can design and build the system without knowing the hardware-related specifications. Therefore, the hardware-related specifications in all documents were poorly done.

- The team also find that the MIS is difficult to be implemented for software games like CraftMaster. The media-related modules cannot be described by mathematical notations. For those modules, the access routines have to be described by natural language, which violates the rigour principle and increases the ambiguity of the documentation.

9.3 Future Reference

This section acts as a reference for further development. Since Pyglet is a high level software library and the modification on hardware controls is limited. The design team will attempt to find a better substitute that allows more freedom on the hardware controls. The anticipated changes specified in the MG will be the primary goal for future design. The team will also learn the Gitlab feature **Continuous Integration(CI)** and apply it to further development process.