

Block Minigame Plugin

COSC412 Individual project

John Mulholland

Professor Broadwater

03/31/2021

Revision Sheet

03/16/2021: First write up.

Preface

The purpose of this plugin is to create a fun minigame for users to participate in. The plugin revolves around the idea of the players in the minecraft server chasing around a randomly assigned block within a five minute window. If a player fails to reach the block in the timeframe, the player will die, the last player left alive is the winner.

The plugin will complete the purpose above, it will be tested for bugs and if bugs are found they will be dealt with swiftly. There should not be anything else that is not related to this plugin added in.

Table of Contents

1.1 Project Overview	5
1.2 Project Deliverables	5
1.3 Evolution of the SPMP	5
1.4 Reference Material	5
1.5 Definitions and Acronyms	5
2.1 Process Model	6
2.2 Organizational Structure	6
2.3 Organizational Interfaces	6
2.4 Project Responsibilities	6
3.1 Management Objectives and Priorities	6
3.2 Assumptions, Dependencies, and Constraints	7
3.3 Risk Management	7
3.4 Staffing Plan	8
3.5 Monitoring and Controlling Mechanisms	8
4.1 Methods, Tools, and Techniques	8
4.2 Software Documentation	8
4.3 Project Support Functions	8
5.1 Work Breakdown Structure	9

1.1 Project Overview

The purpose of this plugin is to create a fun minigame for users to participate in. The plugin revolves around the idea of the players in the minecraft server chasing around a randomly assigned block within a five minute window. If a player fails to reach the block in the timeframe, the player will die, the last player left alive is the winner.

The plugin will complete the purpose above, it will be tested for bugs and if bugs are found they will be dealt with swiftly. There should not be anything else that is not related to this plugin added in.

The plugin will be coded through the use of Eclipse, Bukkit, and Minecraft.

1.2 Project Deliverables

Everything in this project will be delivered/presented through Discord or Github.

A presentation of the plugin will be given before March 31st this will include some Use Cases, Sequence Diagrams, Requirements, and the SPMP.

A working version of the plugin will be ready for play testing before May 7th. The plugin will also be presented around this time as well.

1.3 Evolution of the SPMP

The SPMP will always be kept up to date with any changes that stray from the requirements stated by the SPMP. John Mulholland will be incharge of keeping the SPMP up to date.

1.4 Reference Materials

1.5 Definitions and Acronyms

Bukkit: The plugin software that builds the plugin for the server.

Plugin: A file that changes the games original files.

Server: Locally ran server that houses the plugin and will be used for testing the plugin.

Eclipse: Java IDE used to code the plugin.

Minecraft: Video game where the base code will come from.

2.1 Process Model

The developer will use bukkit commands in Java in order to complete the game. The programmer will then test and run the program using a minecraft server that is run through .jars. The server will be built using a combination of bukkit and standard minecraft server files.

2.2 Organizational Structure:

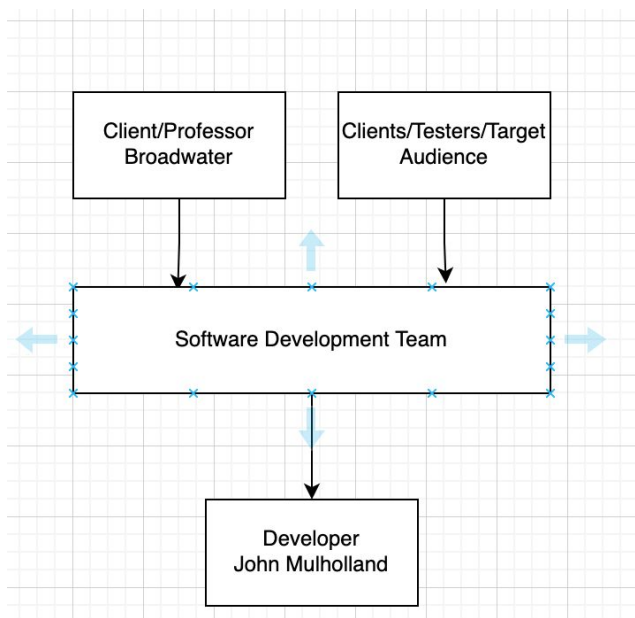


Figure 1: Organizational Structure

2.3 Organizational Interfaces

The developer will use bukkit commands in Java in order to complete the game. The programmer will then test and run the program using a minecraft server that is run through .jars. The server will be built using a combination of bukkit and standard minecraft server files.

2.4 Project Responsibilities

John Mulholland will be responsible for all events/actions that occur during the testing and creation process of this plugin.

3.1 Management Objectives and Priorities

John Mulholland will be responsible for all events/actions that occur during the testing and creation process of this plugin. Once the game has begun working, bug hunting will be the biggest priority, followed by adding additional functionality if possible. The project will be ready for play testing and presentation before May 7th. All funding will come from John Mulholland and will be evaluated when funding is deemed necessary.

3.2 Assumptions, Dependencies and Constraints

This project will be dependent on the opinion of the play testers and the developers decision making.

3.3 Risk Management

In the event of

- Lack of funding: The project is free to complete and will be completed
- The project is too large: In the event the scope is too difficult to complete, I will complete the project to the best of my ability and attempt to complete it past the presentation time.
- Requirements creep: If the requirements become too much, I will decide which requirements are considered vital to project completion and complete them in the order.
- Staff: This is an individual project and as such no staff will be hired, and there will be no danger of individuals leaving the project.

- Client Disapproval: If the client/play testers do not like this product their opinion will be noted.

3.4 Staffing Plan

There will be greater than 2 play testers and one developer.

3.5 Monitoring and Controlling Mechanism

The play testers will be in charge of finding bugs and reporting them to the developer for swift fixes.

4.1 Methods, Tools and Techniques

The developer will use bukkit commands in Java in order to complete the game. The programmer will then test and run the program using a minecraft server that is run through .jars. The server will be built using a combination of bukkit and standard minecraft server files. The information necessary to use all these files will be learned by using a variety of youtube videos and other developer files.

4.2 Software Documentation

There will be inline comments and any necessary changes will be committed in the SPMP.

4.3 Project Support Functions

Plan for quality assurance: There will be play testers and constant communication amongst the client and the developer. Problems will be regularly assessed.

Configuration management plan: Any changes in the plan will be kept track of in the revision list on the SPMP. Requirements will also be regularly changed as the developer sees fit.

Verification and validation plan: Whenever major changes occur to the program the developer will ask the play testers to interact with the game in hopes to find bugs in the changes.

The developer will also run regular bug testing using commands. These commands will allow the developer to test out individual parts of the game without actually running through the game and are solely for bug testing. They will be removed from the final product when the developer deems the project to be bug free.

5.1 Work Breakdown Structure

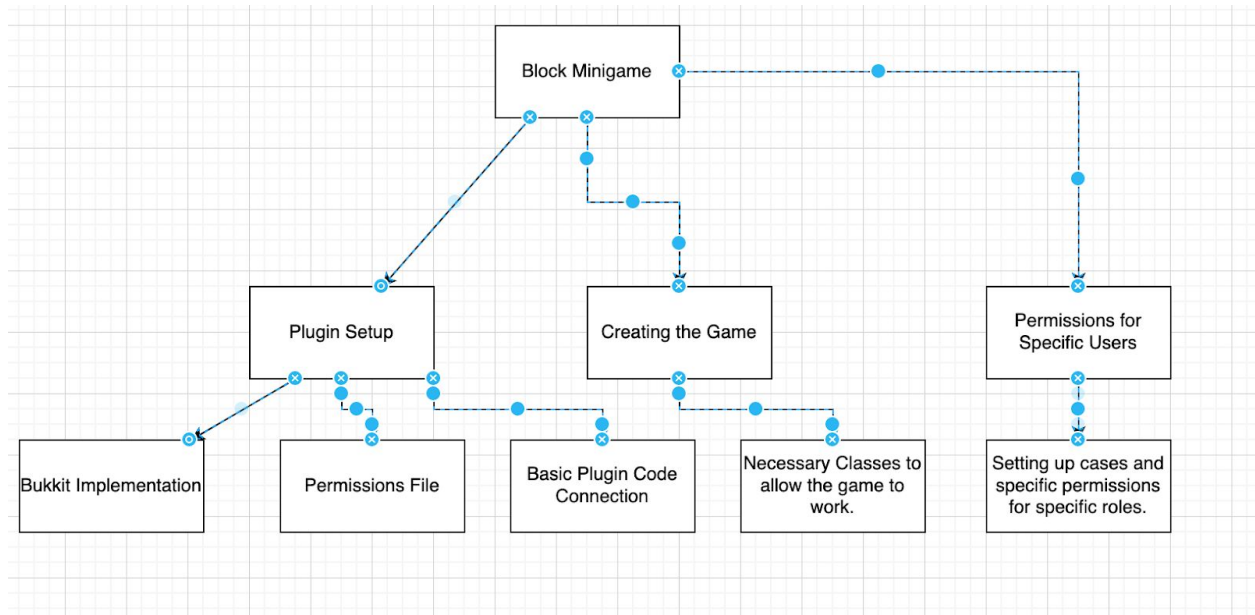


Figure X Work Breakdown Structure