

03-T2 Minecraft

Test Plan and Report

CS 4850 - Section 01 – Spring 2026

April 12, 2026

 Matthew Pflieger Team Leader Developer	 Andrew Burgos Developer	 John McDowell Developer	 Hezekiah Holloman Documentation	 Chase Holt Documentation
---	---	---	--	--

Team Members:

Name	Role
Matthew Pflieger (Team leader)	Developer
Chase Holt	Documentation
Andrew Burgos	Developer
Hezekiah Holloman	Documentation
John McDowell	Developer
Sharon Perry	Project Owner or Advisor

Table of Contents

Table of Contents.....	2
1. Overview.....	3
1.1. Purpose.....	3
1.2. Scope.....	3
2. Testing Summary.....	3
2.1. Scope of Testing.....	3
2.1.1. In scope.....	3
2.1.2. Out of scope.....	3
3. Test Cases.....	4
4. Test Procedures.....	5
5. Test Environment.....	6
6. Software Test Report (STR).....	6

1. Overview

1.1. Purpose

The purpose of this document is to define:

1. How the team will test the new gamemode
2. How the team will test server deployment of the gamemode
3. How we will test gamerules and mechanics
4. The basis of the test estimates
5. Any risks, issues, assumptions and test dependencies
6. The test schedule and major milestones
7. The test deliverables

1.2. Scope

This document details the testing that will be performed by the project team for the Minecraft project. It defines the overall testing requirements and provides an integrated view of the project test activities. Its purpose is to document:

- We will be testing the gamemode and gamerules and how they deploy to the server.
- We will first test locally independent of the server to test gamerules and find bugs.
- Then we will test on a server with multiple clients.

2. Testing Summary

2.1. Scope of Testing

2.1.1. In scope

Testing server and gamemode are running properly with multiple clients connecting.

Testing lobby functions (connecting, starting, etc.)

Testing gamerules are working (players cannot cross into the opposing side of the arena, one arrow kills, blocks and arrows are received periodically, etc.)

Testing gamerules bounds (is the arrow Interval too frequent, what about blocks?) I.e. are these mechanics as fun as they can be?

2.1.2. Out of scope

Minecraft mechanics. Our game is built in Minecraft, we are not Minecraft developers and cannot change the inherent gameplay.

Wider KSU server architecture. The main hub, and other areas of the KSU playable server are not ours to change/test.

3. Test Cases

- List scenarios to test (functional and non-functional if applicable)

Test Case ID	Description	Expected Result
Turf Tracking	Tracks the divide line between both teams.	Tracks the divide line and updates when a team gets a kill.
Kill-Based Advancement	When a player kills an opponent, divide line will move 1 block into enemy territory.	The territory will update once a player is killed by moving one block into the corresponding territory.
Build Restrictions	Players are restricted when placing blocks.	Players will only be able to place blocks within their territory and not one the enemy team's.
Win Condition	The condition needed to end/win the game.	The game will end once a team claims 100% of the enemy team's territory/ game arena
One-shot Kills	The combat mechanics for the players.	Any arrow hit will be counted as an instant kill and will immediately respawn the player upon a hit/kill.
Respawn mechanics	Players will respawn when killed/hit by an arrow.	After a player is eliminated the will immediately respawn within their appropriate team spawn area.
Materials/Ammo	Players will be provided blocks and a bow and arrow.	Players will be given blocks whose color corresponds to the appropriate team and a bow and arrow and will gain more arrows and blocks after 5 seconds pass.

4. Test Procedures

Test case	Steps
Turf Tracking	1. Start the match 2. Observe the divide line 3. Kill a player for both teams 4. Observe turf line progress and scoreboard
Kill-Based Advancement	1. Start the match 2. Kill players on both sides 3. Observe turf line progress and scoreboard
Build Restrictions	1. Start the match 2. Attempt to place blocks on the enemy team's side
Win Condition	1. Start the match 2. Repeatedly kill players until the line reaches the endpoint 3. Observe if the game correctly ends
One-shot Kills	1. Start the match 2. Shoot a player with a bow 3. Observe if player correctly dies in one shot
Respawn mechanics	1. Start the match 2. Have other player shoot you with a bow 3. Observe whether you respawn instantly after death and skip Minecraft's respawn screen
Materials/Ammo	1. Start the match 2. Deplete blocks and arrows by placing and shooting them respectively 3. Wait and see if the in-game clock refills these resources periodically as expected

5. Test Environment

- Minecraft version 1.20.1-1.21.11
- Pre existing Minecraft server (Oracle Linux 8 ARM VPS)
- IDE capable of running Java
- Java Development Kit (JDK) version 17 or higher
- Java version 21
- Gradle

6. Software Test Report (STR)

- Record results for each test case
- Indicate Pass/Fail and Severity (e.g., Minor, Moderate, Critical)
- Simple table format is fine – recall our example from Requirements

Requirement	Pass	Fail	Severity
Turf Tracking	☒	☐	
Kill-Based Advancement	☒	☐	
Build Restrictions	☒	☐	
Win Condition	☒	☐	
One-shot Kills	☒	☐	
Respawn mechanics	☒	☐	
Materials/Ammo	☒	☐	