

Abstract

This project is a Minecraft server modification that allows for the Minecraft players and administrators the ability to create a support an in-game support ticket to report any issue observed within the gameplay. The modification (referred to as "mod") involves the development using the IntelliJ framework in conjunction with the Fabric API. The mod will write to a MariaDB and the data entry will produce a Discord server message which is the KSU Esports main interface and communication outlet.

Introduction

Minecraft is an online based game where players can create and modify their own "block" world. The premise is adventure-based experience that can be played solo or with a community of other online players. Kennesaw State University uses a shared Esports server for players to collaborate in these block worlds to play mini games and design their own maps. Mods are loaded to the server to influence and change the way the players interact with/ experience the game. In this project, the moderation modification is a server loaded Java based package that allows the player to use the in-game chat functionality to create a ticket and gives the administrators additional functional such as claiming and closing resolved tickets.

Frameworks / Materials

IDE/Languages: The IDE used to develop the code for the Minecraft Mod was the Student Version of IntelliJ. This programming language used was the 17th version of Java.

API/Systems: For internal game data use, the Fabric API is utilized. The script writes to a server-controlled database called MariaDB. The notification of a schema data entry is sent to the Discord server.

Methods

Command Strategy: The Minecraft Mod uses the name "/modrequest" for the player to initiate the script in the chat. The Mod has three subcommands: Create, Claim, and Close. The Claim and Close subcommands are only available to the server administrators to solve the issue to which the ticket is made. The user will type in "/modrequest create help me I need help" to create a ticket for review.

Database Strategy: When a user creates a ticket, there is data such as map location, player ID, etc that is logged from the reporter. This data and the input of the ticket descriptions is then send to the MariaDB to for the information to be stored.

Integration Strategy: There are two parts of the integration strategy. 1) The command script has an access code to where the commands can send data to the MariaDB. 2) The MariaDB will send a notification to the Discord chat for the admins to be notified of the ticket.

Results



Figure 1: The player creates a ticket by calling the Minecraft Mod by typing "/modrequest create" then following the phrase with their issue. This is illustrated in the picture above where the player is creating a ticket to report to the admins that the Minecraft world is lagging.

#	ticket_number	user_name	user_input	location_x	location_y	location_z	timestamp	admin_assigned	status
1	1	insaneocrab	test	706	223	1,396	08:23:22	(NULL)	open
2	2	insaneocrab	help my game is lagging	708	223	1,396	08:25:42	(NULL)	open

Figure 2: The command pulls user name, any user input, (x, y, z) player location, and time from the game and stores in Mariadb.

Conclusions

The conclusion of this project shows how a ticketing system was needed to bridge the gap between the players and the administrators. This Mod allows the players to report one off events but will allow the administrative team to make deterministic decisions based on patterns seen in the type of tickets reported.

Acknowledgments

- Dr. Ying Xie – Faculty Advisor and Professor
- Kylie Nowokunski – Project Owner and Advisor
- Alla Kemelmakher – Director of CSEE Partnership and Events
- Kennesaw State University College of Computing and Software Engineering

Contact Information

- Jay Lambert – [LinkedIn Profile](#)
- Dasiane Taplin – [LinkedIn Profile](#)
- Kathy Nguyen - [LinkedIn Profile](#)
- Hayden Scarbrough – [LinkedIn Profile](#)

References

- Fabric. (n.d.). <https://fabricmc.net/>
- Incendo. (n.d.). *Incendo/Cloud-Minecraft-modded: Cloud implementations for modded minecraft.* GitHub. <https://github.com/incendo/cloud-minecraft-modded>.
- Eldoria. (n.d.). *Resource - connecting to databases - mysql/mariadb | SPIGOTMC - high performance minecraft.* Connecting to Databases - MySQL/MariaDB. <https://www.spigotmc.org/threads/connecting-to-databases-mysql-mariadb.501928>.