Projectmanagement and Softwaredevelopment – year 2022/2023

Duties Record Book

**Team: MeineKraft**

Adam Aigner

Fabian Trenkwalder

**Data: 24th of January 2023**

Table of contents

1. Short Description
2. System Preview
3. Use-Case Diagram
4. Class Diagram
5. TimeTable
6. Short Description

My team partner and me are together developing a plugin for the well-known Java game Minecraft: Java edition. We are rebuilding an existing game mode called “Bedwars” where multiple players or teams are battling against each other. Target is to destroy other players beds because then, they will no longer respawn after their dead

# Trigger

Our plugin starts working as soon as players join the server and then change into the lobby world. Now the plugin changes to “Lobby State” and waits for more players to join. If enough players are available, the plugin changes to “Ingame State” where players can battle. Last state is “Ending State”, which is called when one team won.

# Actors

The actors are only players. When the plugin is done, there is no server administrator needed, to change game states or even reset the lobby. Everything works fine with only players playing

#  Preconditions

Not much is needed before players can start. There must be any bukkit/spigot/paper Minecraft server running on which the plugin works fine. Only the world names are hardcoded, so now you wont be able to just put this plugin on your server and start playing. You must create two worlds called “world\_bedwars\_l” for the lobby and “world\_bedwars” for the battle world. Those can be created with “Multiverse Core” and world changing will be easy with “Multiverse Sign-Portals”.
Spawns of players and spawners can be created by server administrators with special permission.

# Steps

First, a player joins the lobby world, then our plugin changes to “Lobby State”. In this state the plugin is idling and waiting for more players to join. If enough players are available the lobby countdown starts. After that, “Ingame State” starts and players will be teleported to their team-spawns in the battle world.
Now players are able to fight and if one team is the winner, ending state will be started. Now the world will be reset and players are teleported to the normal world. Now the server is available for another round and will be set to lobby state.

# Involved classes

So our plugin has many different classes which are able to handle all of the above steps.
Here’s a list of the most important classes and what they do:

* Commands
This class contains all logic for the commands the server administrator can perform. He is able to create spawns of material and players
* GameStateManager | LobbyState | IngameState | EndingState
The manager class is able to change the current gamestate. The rest contains all logic for the gamestates
* Listeners (e.g. blockBreakEvent, onPlayerDeath, playerWorldChangeEvent)
Listeners are classes which contain functions that are started when something happens on the server. For example: When a player breaks a block in Minecraft, a function in the class “blockBreakEvent” is called. We are checking in which world the player is. Then we check the block and decide if the player shall be able to break it, otherwise the event will be cancled
* Object classes (Team)
This class contains every information about a team. An object Team contains the team color, team players, team spawn many more information

Of course we have many more classes but explaining everything will be to much

1. System Preview

System Preview | Use-Cases

We created a Minecraft plugin with the focus on the game mode Bedwars. Our plugin has three worlds, one of them is the spawn-world, the other one is the lobby-world and the last one is the game-world where you can play the mode Bedwars.

After joining the game you will spawn in a world where you are able to join the lobby by clicking a sign. The sign will teleport you into the Bedwars lobby. You are able to play a knockbackstick-minigame while waiting for other players. If enough people join the game, you and your opponents will be teleported into the main world. Now you are able to enjoy a bedwars game with your opponents.

In the main world we have created 4 places to spawn. Every team has it’s own spawn, marked by colour. If you die, you will spawn at your spawnpoint. In this gamemode you need to protect your bed. If your bed is destroyed (you will be noticed, by the scoreboard and by a message), you won’t be able to spawn again. That means if you die again, the game is over for you. The last person alive wins the game.

1. Use Case Diagram
2. Class diagram

|  |
| --- |
| Team |
| - String color;- int playerCount- int maxPlayerCount- List<Player> players- Boolean bedActive- Location teamSpawn |
| + getColor() : String+ getPlayerCount() : int+ getMaxPlayerCount() : int+ setPlayerCount(int)+ addPlayer(Player)+ setTeamSpawn(Location)+ getTeamSpawn() : Location+ isBedActive() : Boolean+ setBedActive(boolena)+ Team(string, int, Location)+ Team(string, int, int, List<Player>, Boolean)+ getTeamColor() : String |

1. Timetable

