Stellar DEV

Premworld

Developer Documentation for Premworld

Overview

All plugins will be made for the minecraft version 1.17.1. The vision of this project will work as a mini game server that has a couple of different gamemodes and unique systems. Tons of games can happen at once as long as there are players to fill up those games. It will be free to enter these games but to get upgrades it would cost WRLD token. No required NFTs to play any of the games.

When a player attempts to purchase an upgrade for the given game, they need to pay using WRLD, using this API to request money from their linked blockchain wallet. Once it is paid, the player will receive the upgrade.

Party System

Overview

The party system will be a way you can create temporary groups with your friends and have the ability of joining the same game as them. You can either /party create or directly invite a player to create a party using /party invite. There will be different rankings in the party such as the Party Owner, Party Moderator, and Party Member. The owner will have full access to the party and all the commands. The Party Moderator should have the ability to kick members from the party, invite other players, and be able to set the party as open. Setting a party as open by the command /party open will open the party for anyone to join without an invite. /party close will close the party, changing the requirement back to invite-only. Invite only should be the default. Party owners should also have access to commands like /party promote, /party setowner, and /party disband. All should be pretty self-explanatory.

Leveling System

Overview

There should be a global leveling system similar to a leaderboard. Players will earn EXP per game win or game played. After you receive a certain amount of experience points, you should level up. There will be a leaderboard GUI of the players with the highest level. Players will also have a hologram over their head but under their in-game name that states their level. Should scale all the way to level 100 but EXP gain should go forever.

Setup Mechanics

The client should have the ability to set up each level specific EXP requirement. Also for every single game there should be a configuration value for the EXP you get for participating and winning. Easiest way I think you can do this is to participate in EXP for everyone and then like a winning bonus that adds on to it. When a player levels up, there should be a configurable chat message that announces it in chat. The hologram that hovers over a player's head should also be configurable so the client can change the colors/looks around. When a player types the command /leaderboard, it should display a GUI that lists all the top players' Levels and XP. After players reach level 100, the EXP bar should still continue to increase.

Shop System

Overview

There should be a global /shop system where players can purchase passes to play specific games. You won't be able to access the games without this specific pass. The passes will be created through the configuration where the client will be able to set the amount of time they want it to go on for.

Setup Mechanics

Everything should be done in a configuration file. You'll be able to set up a GUI which will have control of setting up everything including costs, times, games, etc.

For changing the time, there should be a value you can change (in minutes) of having access to a specific game. You should also be able to choose what type of game (give them unique types like Coliseum, Hunting, Trials). There should also be a cost value for each pass you want to create to change the price of the pass. The GUI should be very configurable so the client can set up the positionings, lores, etc. After the pass expires, it should automatically retract the player's permission to access the games. To recap, open GUI through /shop, multiple passes that can be made by client, change the cost, time limit, game type, player purchases with WRLD token and can access that type of game for a specific time.

Coliseum Games

Overview

The Coliseum Games will be kind of a category of games. The games include 1v1 battles, 2v2 battles, 3v3 battles, 4v4 battles, and Capture The Flag. Each game will have one time purchases for better equipment using $WRLD.

Battles

This section will be in reference to 1v1 battles, 2v2 battles, 3v3 battles, 4v4 battles. It all will realistically be the same thing but just with different player requirements. Each player will start off with basic leather armor and a wood sword. Before each match begins, the player will be prompted with an upgrade GUI where they can trade $WRLD token for upgrades on their gear. Anyone can join a game without any NFT requirements or entree fees.

Setup Mechanics

For players to join a game, there should be a NPC that is set up by running a command. When that NPC is set up, you can edit their skin, display name and the GUI that is opened when it is clicked in the config. The whole purpose of this NPC is to serve as a way for players to enter the game. There should be a configurable hologram on top of the NPC that supports a placeholder that shows all the users currently playing. There should be 5 buttons inside of this GUI. The first button will represent an information button that includes what the game is, how to play, how to win, etc. The other buttons will be a queue button that will put a player in a queue until other players join. This button should also have a configurable value for whether or not the client decides to have an entry fee. If you queue with a party it needs to be so you can only join the queue that has the same number of party members. So if you have over 4 members you won't be able to compete in the Battles unless you kick players to be 4 or less. Button 2-5 will be for the 1v1s, 2v2s, 3v3s, 4v4s. When players from both sides join and there are enough players to start the match, the players will get teleported to the arena where they will fight. The way setting up the arena will be very specific. In another world, the client will be able to paste in a build as many times as they want. There should be a command process in setting up an arena so first the user will type a command which will begin the process. Next, the user will need to select 2 positions through a command. This will set the arena under all the necessary permissions so users can not build, break, etc. After the positioning has been determined, the user will then prompt a name for some sort of identification in the data file where all this information will be stored. After the name is set, next will be the type. Since there are 4 types of battles, we'll identify types by numbers so Type 1 represents 1v1s, Type 2 represents 2v2s, Type 3 represents 3v3s, and Type 4 represents 4v4s. After the type is recognized, the user will be able to set up the starting positions which pretty much are the positions each team will be in when they get in the match. It should go in order so like if it's type 3 the user should be able to set up 3 of the start locations for one of the teams, go to the other side and set up another 3 start locations for the other 3 competitors. After all this information has been set, it should save it all in a data file and now be complete! The client should be able to repeat this process as many times as necessary. Make sure that when a match is in progress, not to send different players to the same arena. All builds should be randomized so players don't keep getting the same one. All players will start off with full leather armor and a wooden sword. Make this starter kit configurable in case the client's wishes to add food, potions etc. There will be an upgrade system that is match-based which will be described with the details below for the functionality. When a match begins and the players are teleported to the start locations, there will be a cooldown with a configurable time that opens a GUI for the players. Here, the players will be able to purchase upgrades for their armor/weapons. There should be a configurable value for the entry fee of each game that each member needs to pay using WRLD. There should be multiple values for each type of mode. The client should be able to set it as 0 and bypass the payment process. There should also be a configurable pot which is a set amount of money in $WRLD token that is split among the winners. This should also have the ability to be set to 0 to bypass the payment methods.

Upgrades

The client requests there to be an upgrade system where you can buy different types of upgrades in a shop on your armor or weapons. These upgrades will reset after every game and players can repurchase them. The upgrades will be as follows:

- Armor
    - Upgrade your armor for the fight
    - 5 Upgrades - will cost WRLD token
    - Upgrade order is, iron(1), gold(2), chainmail(3) diamond(4), netherrite(5)
    - Should be able to set a multiplier for the cost in WRLD token (higher level more token)
    - Should be able to set a base price for first upgrade
- Weapon
    - Upgrade your sword for the fight
    - 5 Upgrades - will cost WRLD token
    - Upgrade order is, stone(1), iron(2), gold(3) diamond(4), netherrite(5)
    - Should be able to set a multiplier for the cost in WRLD token (higher level more token)
    - Should be able to set a base price for first upgrade

Capture The Flag

Capture The Flag is a gamemode that will also be a part of the coliseum games. The goal of the game will be to travel into enemy territory and attempt to steal their flag and get it back to your base without being killed. When a team has a certain amount of flags captured, the game will end and the team will be considered the winners.

Server Mechanics

Very similar to the one above. For players to join a game, there should be a NPC that is set up by running a command. When that NPC is set up, you can edit their skin, display name and the GUI that is opened when it is clicked in the config. The whole purpose of this NPC is to serve as a way for players to enter the game. There should be a configurable hologram on top of the NPC that supports a placeholder that shows all the users currently playing. There should be 3 buttons inside of this GUI. The first button will represent an information button that includes what the game is, how to play, how to win, etc. The other buttons will be a queue button that will put a player in a queue until other players join. This button should also have a configurable value for whether or not the client decides to have an entry fee. You'll be able to queue with your friends in a party as long as the party doesn't exceed the limit the game’s player count supports. One of the buttons should be 6v6 mode and the other button should be a 12v12 mode. When players from both sides join and there are enough players to start the match, the players will get teleported to the arena where they will fight. The way setting up the arena will be very specific. In another world, the client will be able to paste in a build as many times as they want. There should be a command process in setting up an arena so first the user will type a command which will begin the process. Next, the user will need to select 2 positions through a command. This will set the arena under all the necessary permissions so users can not build, break, etc. After the positioning has been determined, the user will then prompt a name for some sort of identification in the data file where all this information will be stored. After the name is set, next will be the type. Since there are 2 types of CTF modes, we’ll identify types by numbers so Type 1 represents 6v6s, and Type 2 represents 12v12s. After the type is recognized, the user will be able to set up the starting positions which pretty much are the positions each team will be in when they get in the match. The way the number of setup locations is determined is how many players are supposed to be inside of that game. After all this information has been set, it should save it all in a data file and now be complete! The client should be able to repeat this process as many times as necessary. Make sure that when a match is in progress, not to send different players to the same arena. All builds should be randomized so players don’t keep getting the same one. All players will start off with full leather armor and a wooden sword. Make this starter kit configurable in case the client's wishes to add food, potions etc. There will be an upgrade system that is match-based which will be described with the details below for the functionality. When a match begins and the players are teleported to the start locations, there will be a cooldown with a configurable time that opens a GUI for the players. Here, the players will be able to purchase upgrades for their armor/weapons. There should be a way you can set up the flag location for both teams. This will also be the place where you “deposit” flags. There will be a custom texture for this but there should be a way where the client can simply run a command, select the model and it will save it as the deposit/flag block. There will be a flag custom model. When a player wants to deposit a flag, they need to click their own core while having the flag captured. When a player wants to steal a flag, they need to click on the enemy's core which will then transfer the flag to them. When a player tries to deposit a flag but doesn’t have one, a configurable message should pop up in chat stating that they currently do not have the flag. There should be a configurable value for the entry fee of each game that each member needs to pay using WRLD. There should be multiple values for each type of mode. The client should be able to set it as 0 and bypass the payment process. There should also be a configurable pot which is a set amount of money in $WRLD token that is split among the winners. This should also have the ability to be set to 0 to bypass the payment methods.

Upgrades

The client requests there to be an upgrade system where you can buy different types of upgrades in a shop on your armor or weapons. These upgrades will reset after every game and players can repurchase them. The upgrades will be as follows:

- Armor
    - Upgrade your armor for the fight
    - 5 Upgrades - will cost WRLD token
    - Upgrade order is, iron(1), gold(2), chainmail(3) diamond(4), netherrite(5)
    - Should be able to set a multiplier for the cost in WRLD token (higher level more token)
    - Should be able to set a base price for first upgrade
- Weapon
    - Upgrade your sword for the fight
    - 5 Upgrades - will cost WRLD token
    - Upgrade order is, stone(1), iron(2), gold(3) diamond(4), netherrite(5)
    - Should be able to set a multiplier for the cost in WRLD token (higher level more token)
    - Should be able to set a base price for first upgrade

Hunting Games

Overview

There will be a single hunting game which is played solo. You'll be able to queue up with friends in a party but overall you're playing by yourself. Essentially what the game will include is 6 huntable creatures (Bear, Fox, Deer, Boar, Turkey, Bison). The way this will work is the player with the most points killed in a specific time limit wins. Each animal will be assigned a "weight class" where animals with a higher weight will be harder to kill but give more points. You'll get points by killing animals.

Setup Mechanics

For players to join a game, there should be a NPC that is set up by running a command. When that NPC is set up, you can edit their skin, display name and the GUI that is opened when it is clicked in the config. The whole purpose of this NPC is to serve as a way for players to enter the game. There should be a configurable hologram on top of the NPC that supports a placeholder that shows all the users currently playing. There should be 2 buttons inside of this GUI. The first button will represent an information button that includes what the game is, how to play, how to win, etc. The other buttons will be a queue button that will put a player in a queue until other players join. This button should also have a configurable value for whether or not the client decides to have an entry fee. You'll be able to queue with your friends in a party as long as the party doesn't exceed the limit the game's player count supports. When players from both sides join and there are enough players to start the match, the players will get teleported to the arena where they will fight. The way setting up the arena will be very specific. In another world, the client will be able to paste in a build as many times as they want. There should be a command process in setting up an arena so first the user will type a command which will begin the process. Next, the user will need to select 2 positions through a command. This will set the arena under all the necessary permissions so users can not build, break, etc. After the positioning has been determined, the user will then prompt a name for some sort of identification in the data file where all this information will be stored. After the name is set, the user will be able to set up the starting positions which pretty much are the positions each team will be in when they get in the match. The way the number of setup locations is determined is how many players are supposed to be inside of that game. After all this information has been set, it should save it all in a data file and now be complete! The client should be able to repeat this process as many times as necessary. Make sure that when a match is in progress, not to send different players to the same arena. All builds should be randomized so players don’t keep getting the same one. There should also be a command to set up spawn locations for the creatures. It should be random and not spawn all at once. There should be configuration values that alter the amount of creatures that should spawn per minute. Since the creatures need a different weight or as in this case “health”, there should be a configuration option for each creature that you can change the health/weight of it so it’ll take longer to kill. There should also be configuration options for the points that are given when each creature is killed. All players will start off with a wooden sword. There should be upgrades for weapons that you can buy before the game. A list of weapons that you can purchase in the upgrade shop will be a bow, rifle, and shotgun. These will use custom textures and all have different perks. The rifle will work better from long range, the shotgun will be better at close range, and the bow should be a balance of both but do a lot more damage compared to the wooden sword you start off with. The damage values should be configurable for balancing purposes. There will also be 2 utilities you can purchase. One of them being a trap that will catch animals when they walk over it. The player who purchases it for this game will be able to place one on the ground every certain amount of time. This time should be configurable. It should instantly kill any type of animal. The trap should autoregenerate in your hotbar after the cooldown ends. There should also be a configurable chat message that alerts the player when their trap is off cooldown and they can place another one. Next utility is a Lure item. What this item does is it lures animals to your location to kill them easier. There should be a configurable radius for this effect to work. There also should be a configurable cooldown when you right click the item in your hotbar. There should be a configurable value for the entry fee of each game that each member needs to pay using WRLD. There should be multiple values for each type of mode. The client should be able to set it as 0 and bypass the payment process. There should also be a configurable pot which is a set amount of money in $WRLD token that is split among the winners. This should also have the ability to be set to 0 to bypass the payment methods.

Upgrades/Utilities

The client requests there to be an upgrade system where you can buy different types of upgrades for your weapons and purchase utilities.. These upgrades/utilities will reset after every game and players can repurchase them. The upgrades/utilities will be as follows:

- Bow
    - Access to a Bow weapon that shoots arrows and does immense damage
    - Configurable cost in WRLD token
    - Configurable damage values
- Shotgun
    - Access to a Shotgun weapon that does colossal damage the closer you are to an animal.
    - Configurable cost in WRLD token
    - Configurable damage values
- Rifle
    - Access to a Rifle weapon that does supreme damage the farther away you are to an animal.
    - Configurable cost in WRLD token
    - Configurable damage values
- Trap
    - Access to place traps every minute that when an animal walks on top of it, it instantly kills them.
    - Configurable cost in WRLD token
    - Configurable cooldown
- Lure
    - Lure animals from all over to your location to find and kill them easier every minute..
    - Configurable cost in WRLD token
    - Configurable cooldown
    - Configurable radius

Timed Trials

Overview

This gamemode will use the Jet Ski custom models to compete in a race course against other players. It will list the people who complete the race course in order. Very similar to Mario Kart, just with the Jet Skis in Minecraft.

Setup Mechanics

For players to join a game, there should be a NPC that is set up by running a command. When that NPC is set up, you can edit their skin, display name and the GUI that is opened when it is clicked in the config. The whole purpose of this NPC is to serve as a way for players to enter the game. There should be a configurable hologram on top of the NPC that supports a placeholder that shows all the users currently playing. There should be 2 buttons inside of this GUI. The first button will represent an information button that includes what the game is, how to play, how to win, etc. The other buttons will be a queue button that will put a player in a queue until other players join. This button should also have a configurable value for whether or not the client decides to have an entry fee. You'll be able to queue with your friends in a party as long as the party doesn't exceed the limit the game's player count supports. When players from both sides join and there are enough players to start the match, the players will get teleported to the arena where they will fight. The way setting up the arena will be very specific. In another world, the client will be able to paste in a build as many times as they want. There should be a command process in setting up an arena so first the user will type a command which will begin the process. Next, the user will need to select 2 positions through a command. This will set the arena under all the necessary permissions so users can not build, break, etc. After the positioning has been determined, the user will then prompt a name for some sort of identification in the data file where all this information will be stored. After the name is set, the user will be able to set up the starting positions which pretty much are the positions each team will be in when they get in the match. The way the number of setup locations is determined is how many players are supposed to be inside of that game. After all this information has been set, it should save it all in a data file and now be complete! The client should be able to repeat this process as many times as necessary. Make sure that when a match is in progress, not to send different players to the same arena. All builds should be randomized so players don't keep getting the same one. There should be configurable values for the minimum and maximum players that are required to start the game. The client should be able to set up the start points for each Jet Ski. They can set up as many as possible and it will randomly choose one when a player enters the game. To set up the end point, the client will need to run a command and then select 2 positions. The area of the 2 positions will be considered the end of the race and will record the positions and time it took for the players to complete the course. There should be a configurable value for the entry fee of each game that each member needs to pay using WRLD. There should be multiple values for each type of mode. The client should be able to set it as 0 and bypass the payment process. There should also be a configurable pot which is a set amount of money in $WRLD token that is paid to the winner of the race. This should also have the ability to be set to 0 to bypass the payment methods.

Gameplay Mechanics

The race is going to be something similar to Mario Kart in terms of gameplay. As the race begins, all the participants can start moving. The way this will be done is it will have the same controls of a boat but look like custom textures of the Jet Ski. It needs to be a lot faster by default as a boat but still controllable. There will be a start and end point, after users meet the end point their place in the race is determined. After everyone has crossed the finish line, it should display the placements of the competitors in chat.