**To Do Decide if you really want to use local P2P or if you want to start a server**

|  |  |  |  |
| --- | --- | --- | --- |
| P2P |  | Server |  |
| Its easier |  | I could learn more |  |
| Conflict problems |  | I would have to host a database and server |  |
|  |  |  |  |

**Multiplayer**

* the host owns the world
* Player orientation, skin names, and movement are sent to directly to all other players
* When getting a new chunk, the host gives it
* If a player modifies a chunk, it doesn’t get sent to another player unless they are close enough to it, that includes the host.
  + Changes eventually get sent to the host. If the host is not around the change, the change is cached.

I want a world that is shared between peers

**Unless I have a shared server, I will have to deal with conflicts.**

**Hosting a server is technically more straightforward, however it is probbably also more difficult to set up**

* + I **will choose the peer to peer solution**

**Option 1: host a server:**

* + We will need a server to update changes, as well as needing to connect to the internet to play thus avoiding conflicts.
  + This would definitely make online play across the internet safer and easier
  + If I have a server, there is no need for a p2p system
  + One caveat would be that the world information would be mostly stored online. You won’t be allowed to join unless you are connected to the internet.
    - Changes that other players have made that you
  + The server contains the entire world in its database
    - All changes made are transmitted to a servers database
    - When a player reaches a chunk, the server gives them the chunk from its database
    - Any changes made when a player is within viewing range is **also** transmitted directly to said players

**Option 2: peer to peer solution:**

* + The worlds are stored on everyone’s computers, when played as multiplayer, everyone’s world is like an online server, but that also means that the world can be played just like any ordinary local world when played alone.
  + When joining in multiplayer, If the users world already exists, merge changes with its peers and use that, otherwise, make a world containing all the information from the game master (prompt user before doing this)
  + Players
    - Players transmit their positions instead of having position handling on them in game
    - The players still have collision boxes that we can bump into
    - No more player model, just player

* + The chunks of the entire world are shared on each players computer
  + Each player has a list of changes made (LOC), it looks like this:
    - <blocks changed>, player1,player2
    - The first column is the change, the second column are the players that have not received this change yet.
    - Rows represent chunks
      * If a block is set in an existing row (chunk), it is added to the row and the list of unvisited players is set to everyone

* + All changes made are added to the LOC
    - for players within viewing range, the changes are **also** transmitted instantaneously
    - for players that come towards changes I modified, the changes are sent to them if they have not yet received the updated changes

* + If 2 people make changes to the same place, when they are not connected to each other, it will create a conflict.
    - We can somewhat prevent conflicts by allowing players to lock chunks that they don’t want others to modify during online multiplayer
    - If a conflict is presented, both changes will overlap each other
      * The person that made the latest change wins

* + When I leave, all changes made will be broadcast to all other players

### **Option 3: Host and clients**

* The owner of the world hosts the server and the clients join
* This gets rid of all issues with block conflicts

**Local P2P Approach**

Each players chunk is its own change file

Note Local P2P is easier, Multiplayer server will handle conflicts and is more straightforward

Note Server will allow me to review some of my server skills

**Handling conflicts**

* + solid blocks take priority over non-solid blocks

**When I join the game**

* + The host gives me the world info
  + If I don’t have it, make a new world.
  + Exchange chunks with other players
    - All joined players send me all of the chunks that have changed
      * We know a chunk has changed if it was modified after my world chunk was updated or If said chunk on my world doesn’t exist.
    - Send all of my chunks (that are changed for them) to other players
      * Only send voxels and entities (dont include light)
  + Apply ALL the changes from the change files of other players

**When I start a local game**

* + Apply ALL the changes from the change files of other players

**During the game session**

* + If i make changes to a chunk while other players are around, those changes are sent directly to them
  + If there are players that were not around but own a copy of the world, a change record is written with the changes on the chunk and a list of players who have not seen the change yet
    - Change records only apply to the current session of the game.
    - Send these immediately or after interval

* + If a new client joins, send all changed chunks to him and receive and install the changed chunks from him

---------------------------------------------------------------------------------------------------------------------

To Do Create a basic server-full peer to peer network system

To Do Start by making a system to connect all players

To Do Connect each player by tapping into a single computer

To Do Only 1 client per IP address

To Do Connect all players using the p2p servers

To Do Each computer should have a model of each player

To Do Each player has a name and color

To Do Make player color

To Do Add third person view

To Do Add a progress bar for the world loading

To Do Add multiplayer to ui

To Do Play multiplayer on load world button

To Do The play multiplayer will open a popup showing this computers IP address and players joined along with a continue button

To Do Join multiplayer button

To Do Goes to a window where we can enter the target ip and join

To Do Top menu character options settings menu

To Do Test 2 player connection with real computers

To Do It even works in simulation

To Do Test 3 player connection with real computers (IT WORKS!!!)

To Do Simulation fails (Leave it. Just do real networking)

To Do Not only is simulation a limitation due to the inability to separate computers, it is also possible that something that works in simulation will not work in real life

To Do **KEEP THE OLD CODE!** If we make the old code “obsolete “ it could be permanently deleted

To Do Build on top of playerServers

To Do Connect all players in the Xbuilders project

To Do Share the project across computers

To Do Start with connection of players

To Do Share worlds when connecting

To Do