

MARCH 17, 2023

# MULTIPLAYER CREATIVE SANDBOX

DRACOTT, PERCY  
HILLS ROAD

# Analysis

## Project Overview

In this project, I will be exploring a Solution to a problem, to fulfil the needs of my client. Video game, since their creating have been a means of socialising, connecting and relaxing. A recent boom in popularity, owing to people spending more time gaming since the recent Covid-19 pandemic and lockdowns, has resulted in games becoming a staple in many people's lives. This ever-growing popularity creates a strong demand for new and unique games to enjoy.

## Project Outline

My client is an avid video game enjoyer, who like many enjoy the social aspect of video games. They require a multiplayer game in the "Creative Sandbox" genre, so that they can play and create with their friends.

While this may be available through other games, their requirements for a game that runs well on very low powered hardware so that they can run the game on their laptops in a LAN configuration, vastly limits the choices of game. This excludes games such as terraria or Minecraft.

With all this in mind, the client has made it clear that they want a complicated and optimised map generation algorithm as well as multiplayer capability, as connecting on the same session is the most core component, and the most decorated feature.

## My Client

My client in question is an avid video game fan. They particularly like retro style games, with simple and well executed features.

## An Interview with Elliot - My Client

Q1. How would you like the game to be played?

A1: The Game needs to be playable on PC, as my friends and I all have laptops which we play our games on. Therefore, the controls need to be suited to a keyboard and mouse, and with a trackpad friendly control scheme.

Q2. How would you like the game to be used and by who?

A2: The game will be played by me and my friends using multiplayer. In either a LAN with all players on my Wi-Fi, or over a WAN hosted by me as I know how to set my router up for Port-Forwarding.

Q3. Which aspects of a "Creative Sandbox" do you find most enjoyable?

A3: For me I find the building aspect of the game most enjoyable. Coming up with an idea for a build. then creating it in 'reality' most appeals to me. I also enjoy exploring the game world, as I can often find inspiration of new builds in the landscape.

Q4. Which features would you most associated with a “Creative Sandbox”?

A4: In my opinion, gathering resources for a build through mining is a key feature of a creative game. As it rewards the player with a real sense of achievement and satisfaction, knowing the work it takes to complete their build. I also enjoy the sense of progression, as getting new tools makes once difficult tasks seem trivial.

Q5. Which areas of a “Creative Sandbox” map are you most interested in?

A5: Of course, the landscape of a map is important, as the majority of the gameplay occurs there. However, I find exploring the thrill of exploring cave systems in games to be the most enjoyable.

## **Objectives**

01. Have functioning game element, which:
  - 1.1. Allows the player to move around the world,
  - 1.2. Contains hostile Mobile-Entities (mobs) in the world,
  - 1.3. Has some form of combat between the player and the mobs,
  - 1.4. Has a unique procedurally generated terrain and cave system,
  - 1.5. Has a destructible world where blocks can be mined by the player,
  - 1.6. Has a building system where the player can build structures,
02. Have a functioning Menu System, which:
  - 2.1. Allows for new games to be created,
  - 2.2. Allows for players to connect online,
  - 2.3. Displays the game title,
  - 2.4. Allows the user to close the program,
03. Have a functioning Multiplayer System, which:
  - 3.1. Allows multiple players to play in the same world,
  - 3.2. Allows the world to be synced across each player,
  - 3.3. Allows interactions between players,
  - 3.4. Allow players to connect through LAN or WAN
04. Satisfy the clients brief by:
  - 4.1. Have a save-able and load-able world,
  - 4.2. Have a Day / Night cycle,
  - 4.3. Include fantasy mobs such as skeletons, or zombies,
  - 4.4. Be playable on PC,
  - 4.5. Be playable on a mouse and keyboard or trackpad,
  - 4.6. Have a tunnelling cave generation algorithm,
05. Be appealing to my End-Users by:
  - 5.1. Having a consistent art style throughout,
  - 5.2. Using a consistent pixel by pixel tile size throughout,

## **Similar Games**

In order to create my program and explore possible features, I have researched into 2 existing “Creative Sandbox” games. Minecraft and Terraria.

Minecraft	Terraria
<ul style="list-style-type: none"> <li>Made for PC and other platforms.=</li> <li>3D</li> </ul>	<ul style="list-style-type: none"> <li>Made of PC and other platforms</li> <li>2D</li> <li>Can create free multiplayer games</li> </ul>

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Able to create LAN multiplayer games, however there's no inbuilt way to play WAN multiplayer for free.</li> <li>• Uses 16x16 pixel art style.</li> <li>• Procedurally generated map with cave system.</li> </ul> | <ul style="list-style-type: none"> <li>• Uses pixel art textures.</li> <li>• Side Scroller</li> <li>• Items to enhance your character.</li> <li>•</li> </ul> |
|---|--|

## **Research**

From these notes I have devised that my "Creative Sandbox" game will:

- Be a side-scrolling 2D game, to reduce computational load on a system.
- Use Keyboard and Mouse/Trackpad controls.
- Feature pixel art graphics (8p8px).
- Have free WAN multiplayer capability.
- Include a uniquely generating map and cave system.
- Include crafting/progression to enhance a player's character.
- Include game saving where the world can be saved and loaded from the server.

My solution will require two applications, a server and client. My client will run the server application and then, they can connect using client applications.

## **Outline of Solution**

### **Random Map Generation**

To meet the clients brief, I will be making use of random map generation, meaning that a uniquely generated map can be made for each new game. In order to do this, I will need to find a way to randomly generate a map.

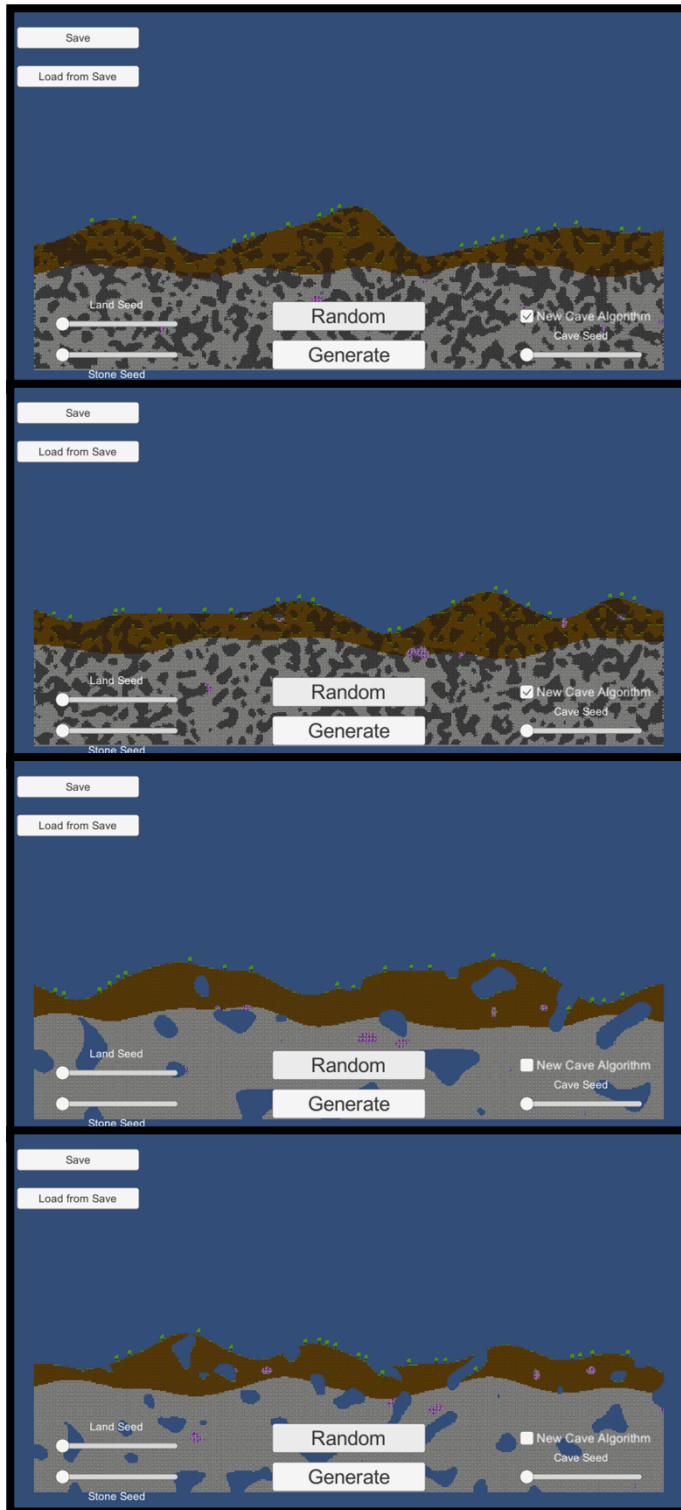
Although, Minecraft is a 3D game its' early beta methods for world generation are great to analyse. As it is made for 3D, an altered 2D version of its' system producing a single slice would be highly efficient. For my research of Procedural generation therefore, I will have researched the solution used by beta versions of Minecraft. Minecraft uses a Perlin noise function to give the height at each given coordinate, in addition to many other functions to add trees and caves when creating a map. As my game will be 2D I will have to research other methods of generation of cave systems and other world features. For cave generation, I will produce a prototype to explore different generation methods; giving me a chance to analyse and evaluate each solution. In this I will explore both a Perlin noise approach and an iterative cellular automata approach.

### **Multiplayer**

For multiplayer I will be using Riptide Networking through Unity, I chose this as it offers a high degree of control of network entities through code, with open-source code. For my client, I will produce two applications, a server and client build. With clients many clients connecting to one server. Clients will have to send the position of their player to the server as player movement will be managed client side. Ordinarily this is a bad idea for a multiplayer game, as this makes cheating much easier; a cheater can simply change the data in messages sent to the server, however Elliot has said that this won't be a problem as only a few trusted friends will have access to the game. As well as this, managing

movement client side will spread the processing load across multiple systems in a thick-client type solution, reducing the load placed on the computer running the server application, Elliot notes that he is likely to run both the client and server applications simultaneously on his laptop and therefore optimisation of the networking side is key.

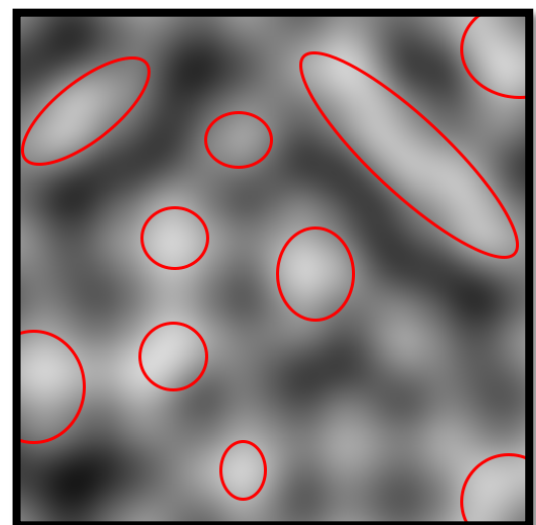
## World Generation Prototype



In this prototype, terrain generation is implemented using Perlin noise, by giving the function the x-coordinate and a seed value instead of a y-coordinate, for each value of x. This produces a smooth and varied terrain, before using Perlin noise I had experimented with using a  $\text{Sin}()$  based function to generate terrain, however this led to a far too periodic looking terrain with repeating hills and few areas of flat land. Elliot says it's important for varied terrain to keep worlds looking interesting, however having a few flatter spaces is ideal for building.

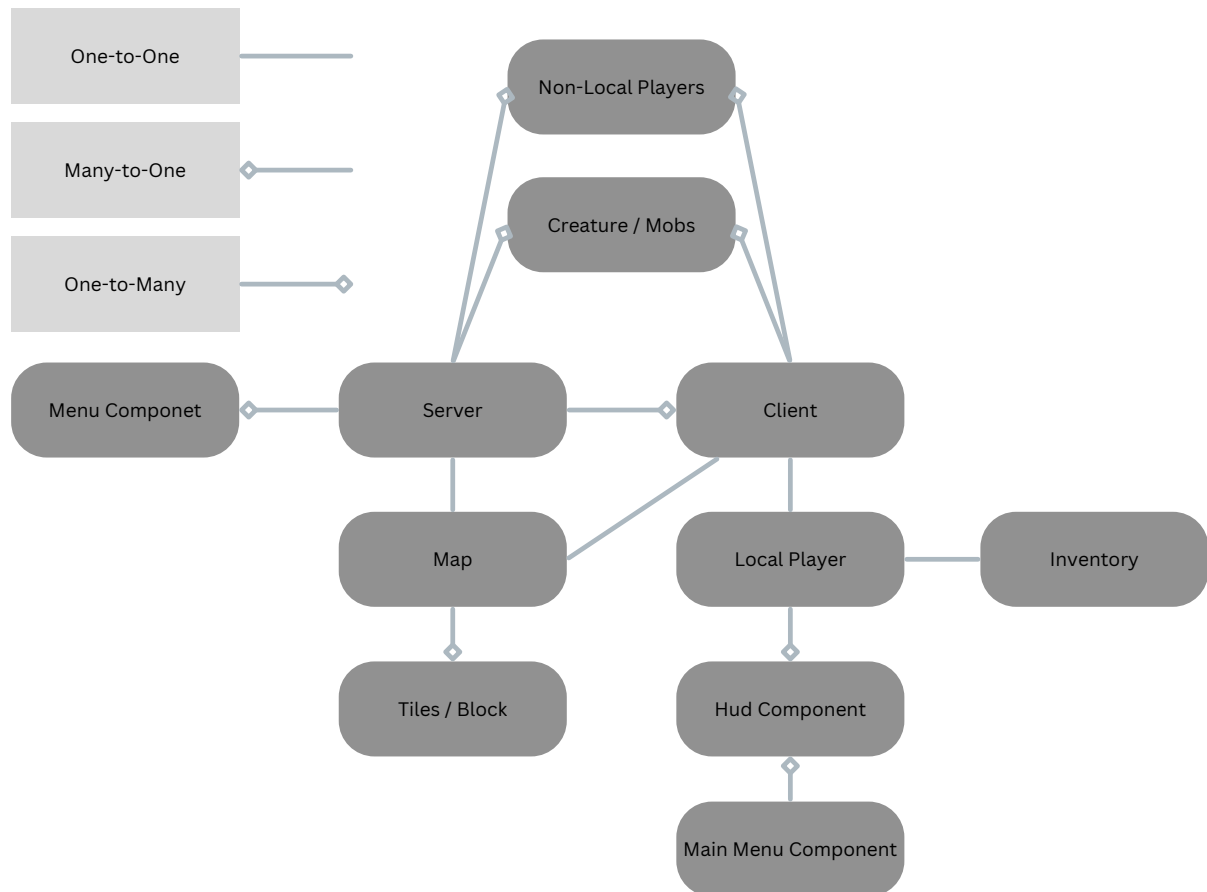
Trees are added randomly, using a random integer value for the x-position, then checked for collisions to prevent trees from spawning onto of each other.

The first two screenshots show a cellular automata approach to cave generation. This code will be explained in Documented Design.



Elliot says that he greatly prefers the Cellular Automata based cave generation algorithm, as the caves produced by the algorithm are far more complex and varied.

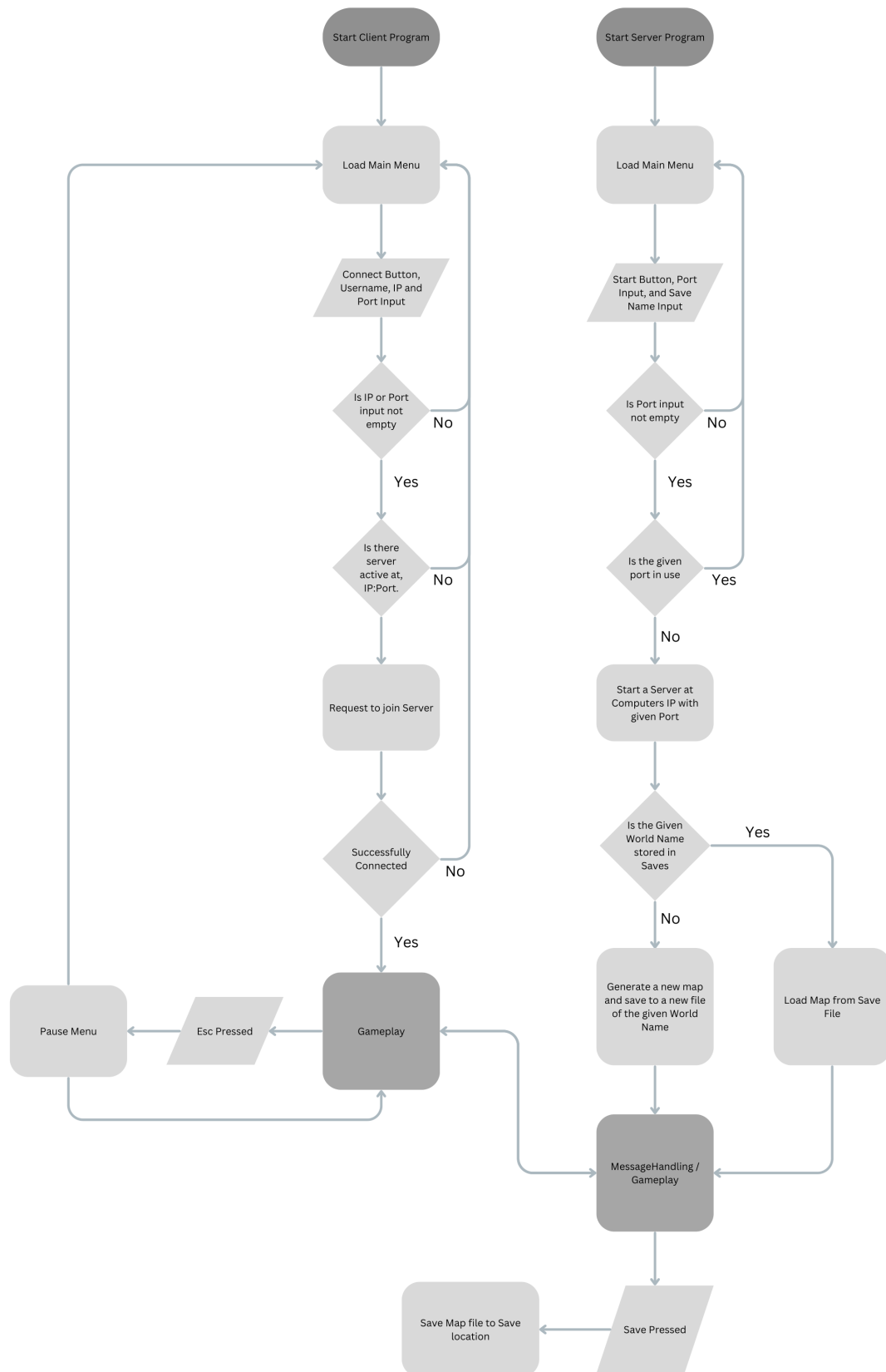
### Entity Relationship & Scene Diagram



Within Unity entities, known as “game objects”, can be added to scenes, using the Instantiate() function. As shown above, on the client side a local player will be added to a scene in addition to non-local players. A local player contains all the control and interaction scripts which takes in the users’ inputs. A non-local player is a projection of another clients’ local player, whose position is sent to all other clients through the server to update the position of its’ corresponding non-local player.

Map, Mobs and Non-Local players, will be shared between both Server and all Clients connected. With. Mobs being controlled, spawned and respawned by the server.

## System Flowchart



## Required Runtime Data

Player GameObject		
Data Name/ Access Name	Data Type	Represents
transform.position	Vector3	Position of the player in the scene
transform.scale	Vector3	The overall scale of the player in the scene
transform.localScale	Vector3	The scale of the player relative to its current scale
playerHealth	integer	Current player health
inventory	Integer array	An array of the current block held in the inventory
hasAxe	bool	Whether the player has a Pickaxe
hasSword	bool	Whether the player has a Sword
itemHolding	integer	The ordinal number for the inventory array
ScreenToWorldPoint(Input.mousePosition)	Vector3	Returns the position of the mouse in relation to its position in the physical scene
velocity	integer	The velocity of the player in the scene
OnGround	bool	A physics overlap at the players feet to determine if they are on the ground
Id	ushort	The players Id in the scene
IsLocal	bool	Whether the player is the Local player

Zombie GameObject		
Data Name/ Access Name	Data Type	Represents
transform.position	Vector3	Position of the entity in the scene
transform.scale	Vector3	The overall scale of the entity in the scene
transform.localScale	Vector3	The scale of the entity relative to its current scale
playerInViewRange	bool	Whether a player is in sight range



PlayerInAttackRange	bool	Whether there's a playing the attacking range
closestPlayer	Vector3	The closest player entity to the zombie
Id	ushort	The Id of the zombie in the scene

Input field Component		
Data Name/ Access Name	Data Type	Represents
text	string	The text held in the interactable area
position	Vector2	Position on the UI canvas

Button Component		
Data Name/ Access Name	Data Type	Represents
DisplayText	string	The text held in the text area
position	Vector2	Position on the UI canvas
colour	Color	The colour applied to the texture of the button

Tile Component		
Data Name/ Access Name	Data Type	Represents
TileMap	TileMap	The tilemap the tile belongs to.
Tile	RuleTile	The current tile being rendered
position	Vector2	The position of the tile in the TileMap's grid
tilemapCollider	Collider	The physics collider of the tile

Hud Component		
Data Name/ Access Name	Data Type	Represents
text	string	The text held in the text area
position	Vector2	Position on the UI canvas
texture	Texture	Current rendered texture

## **Asset Requirements**

Text fonts used will be acquired from [www.dafont.com](http://www.dafont.com), specifically using fonts that are completely royalty free, all fonts used will be credited in the source tracker.

All textures are of my own making, made using Adobe Photoshop.

## **Design**