

SandBox Wars 3D

Game Concept

SandBox Wars 3D Game Concept

SandBox Wars 3D Game Concept	2
About SandBox Wars 3D	3
Floating items rendering solution	4
Game Entities	5
The environment	6

About SandBox Wars 3D

SandBox Wars 3D is a MMORPSG, this is the acronym for Massive Multiplayer Online Role Playing Sandbox Game.

The game is actually more than just this, since it contains elements from other game classes, such as strategy and adventure.

SandBox Wars 3D is totally online, all the players play in the same network, this means that (provided that they have the in-game equipment to do it), two players could always meet, without switching server.

~~“Massive” because a huge number of players should be able to play together at the same time.~~

Every player can only control one character, they may sacrifice it to create a new one, but **absolutely** no player should control two characters at the same time.

The server is populated by communities, a player can decide if he wants to join a community, or to play alone.

Every player may decide to go on adventure alone or with friends, he may decide to explore planets colonized by other players or NPCs, some NPCs may provide quests for players.

Every player can decide what kind of character he wants, while playing, it will train on specific tasks, and grow stronger. Depending on the personality of a player, his character will evolve differently and acquire a specific role in the community.

The player may also decide to ask help from advanced civilizations, and mutate their character from a specimen to another. The game also includes the possibility to build machines able to directly modify creatures genetically, thus allowing to deeply customize player's avatar.

The map is not static, any player can (with the correct tools and technologies) interact with the environment and modify it, anyway, both NPCs and players can adopt defensive measures, such as shield generators and hiding underground.

The map will be explorable both on the surface and underground.

On the first log-in, a small chunk of map is assigned to every player, they will find basic hardware and shelter here, plus a space where to familiarize with game mechanics.

When ready to join the community, they will just have to pass a portal to spawn in one of the stations built by server administrators and other players (selected randomly depending on character genes, players of the same race will spawn in the same station).

Players can craft items from materials, and build machines with crafted tools, they may also combine materials to obtain alloys.

The server will be composed by a network of planets, accessible through portals (one per planet, suitable for big communities), or space ships (suitable for beginners, but not so affordable).

Every planet will have a huge space (initially desolated) with a unique biome.

Anyone can colonize a planet (or just a land) and build up a base, but every planet will provide different resources, players will have to conquer other planets or trade with other colonies to build an efficient settlement.

Multiple starting classes: Human, Android, (others)

Characters level up on particular skills during gameplay

Map composed by Organic Voxels

Floating items rendering solution

A floating item is composed by a light cone and a floating model.

Depending on what the floating item stores, both the model and the light cone will vary.

Every storable object will have an associated colour, light cone will assume the average colour of all contained objects.

Since floating items are affected by gravity, if the object is falling, the light cone should not exist, when the object reaches the ground, it should slowly dampen the fall and go back to a floating state, while the light cone should slowly brighten.

Items should sink in any kind of fluid or liquid, with a varying speed depending on the fluid viscosity (and not the total mass of stored objects).

The upper model will keep swinging and rotating gently as if kept floating by the light cone.

Depending on the amount of objects stored in the floating item, the floating model will be different, it will be composed with models of the three most common objects stored in the floating item, or less if there are less than three object stored.

Examples:

- There is 1 yellow box, the floating model is a yellow box.
- There is 1 green box and 1 purple one, the floating model is composed by both a green and a purple box.
- There are 2 blue and 1 red box, the model is composed by two blue and a red box.
- There are 100 grey and 1 red box, the model is composed by two grey and a red box.
- There are 10 pink, 1 blue, 5 red and 3 grey boxes, the model is composed by a pink, a red, and a grey box.
- There are 1 blue, 1 grey, 1 red and 1 white box, the model is composed by a blue, a grey and a red box.

Since the list of stored objects is ordered, if there are many objects with the same quantity, the first will be used as components for the floating model.

Model coordinates:

1: (0, 0, 0)

2: (0.3, 0.3, 0.3) (-0.3, -0.3, -0.3)

3: (0, 0.6, 0.6) (-0.6, 0, 0) (0.6, -0.6, -0.6)

Game Entities

The game engine keeps track of the status of the world with a few different types of entities:

Creatures: they have a location, a single-axed rotation, and a motion speed, they are affected by gravity and collisions.

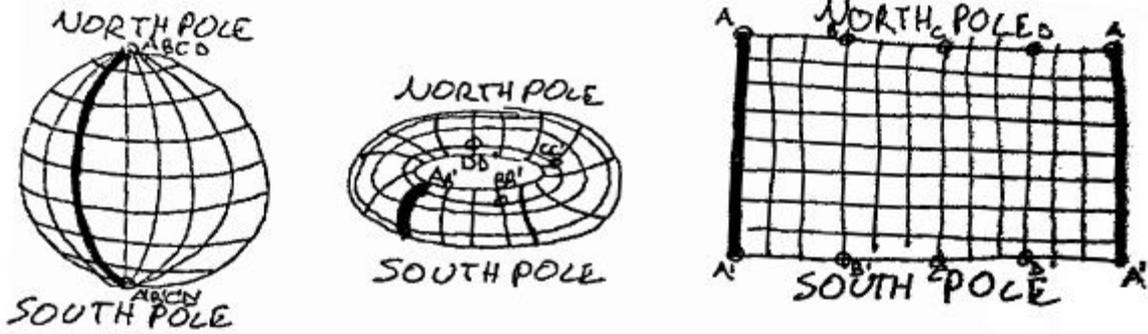
Items: they have a location and they are affected by gravity, they can be picked up by creatures.

Models: they have a location and a fixed rotation (0, 90, 180, 270), they act as blocks, but are rendered as meshes, they are not affected by gravity.

The environment

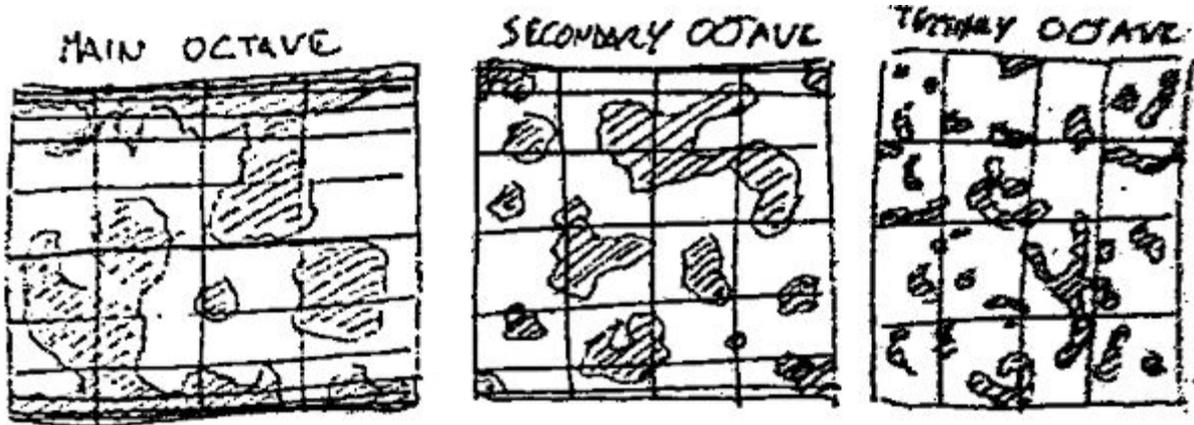
The server will be composed by a network of planets; every planet will have a huge space with one or more unique biomes; every biome will provide different resources.

Each planet will be a



seamless 2D matrix mapped to a sphere. The 2D matrix will just represent the surface of a 3D matrix of voxels.

The height surface will be



generated with a perlin noise-like function stretched on the poles at longer octaves, and plain at shorter.