**Brave New Galaxy**

**The Design Document**

Creator: Bill DeHaven

Version #: 1.0.0

Date: October 11, 2014

Contents

[Version History 4](#_Toc400824405)

[Game Overview 5](#_Toc400824406)

[Design Language 6](#_Toc400824407)

[Design: Xenogrunge 6](#_Toc400824408)

[Graphical Style: Illustrative Retro-Noir 6](#_Toc400824409)

[Mechanics 7](#_Toc400824410)

[Procedural Generation 7](#_Toc400824411)

[The Galaxy 7](#_Toc400824412)

[Planets 7](#_Toc400824413)

[Voxel-Based Terrain 7](#_Toc400824414)

[1st/3rd Person Control 7](#_Toc400824415)

[Interplanetary and Interstellar Travel 7](#_Toc400824416)

[Resource Collection and Crafting 7](#_Toc400824417)

[Real-Time, Server-Based World Persistence 7](#_Toc400824418)

[PvE and PvP 7](#_Toc400824419)

[Community Controlled Social Systems and Economy 7](#_Toc400824420)

[RPG Plotlines and Quests 7](#_Toc400824421)

[Game World 8](#_Toc400824422)

[The Galaxy 8](#_Toc400824423)

[Star Systems 8](#_Toc400824424)

[Planets 8](#_Toc400824425)

[Ecosystems 8](#_Toc400824426)

[Races 8](#_Toc400824427)

[History 8](#_Toc400824428)

[Technology 8](#_Toc400824429)

[Travel 8](#_Toc400824430)

[Extras 9](#_Toc400824431)

[Resources 10](#_Toc400824432)

[Future Features 11](#_Toc400824433)

# Version History

**1.0.0** – Design Document first published.

# Game Overview

Project: Brave New Galaxy

Possible titles: Brave New Galaxy, Brave New Universe, Dark Matter

Media Type: Java Desktop Game

Target Audience: Males, 12- to 30-year-olds, MineCraft players, sci-fi fans, rpg players, MMO players.

Genre(s): MMORPG, voxel

Screen Projection: 3D

Programming Language/Environment: Java w/OpenGL and Slick2D

# Design Language

According to Wikipedia:

"A design language or design vocabulary is an overarching scheme or style that guides the design of a complement of products or architectural settings. Designers wishing to give their suite of products a unique but consistent [look and feel](http://en.wikipedia.org/wiki/Look_and_feel) define a design language for it, which can describe choices for design aspects such as materials, colour schemes, shapes, patterns, textures, or layouts. They then follow the scheme in the design of each object in the suite" (http://en.wikipedia.org/wiki/Design\_language, accessed October 11, 2014).

## Design: Xeno-Grunge

“Grunge” – a style deriving influence from an emotionally dark, hard rock music genre. Centered in Seattle, this music inspired art in a variety of northwest locations including coffee shops and pubs. It searches for beauty in the decay and disorder of reality, which will be one of the overarching themes of the game.

“Xeno” – pertaining to aliens, from the greek for *guest*, strange. “Strangeness” will be another overarching theme of the game, as each of the races within the game, alien or humanoid, live in a very different way than we do today. Even human beings in the game will be living in a time several thousand centuries in the future, and their worldview, though with similarities, will effectively be alien to our own.

## Graphical Style: Illustrative Retro-Noir

Illustrative Retro-Noir is both moody and epic, semi-realistic but pixeled -- somewhat decayed. Similar to Quake 2 and the 3D games of that era, it should allude to the sophisticated but dilapidated technology systems featured in the game.

The main GUI elements should replicate the technology of the player's race, whether that be digital, biological, chemical, nuclear, mechanical, or mental. GUI elements should also make intuitive use of transparency and allow the player to align the various menus and control their attributes in a layout that makes sense to them.

Each 3D model and mesh should use enough polygons to create a simple detail within the game, but only that many. The game's graphical style does not require realistic graphics, but will strive to shy away from 'block' graphics.

Each race will also utilize a style, or styles, of architecture that make sense with their history and values. Each race will also have its own color palette, though items used by or manufactured by that race do not necessarily always need to use just those colors. Above all else, the graphical style for the world and the objects within it must seem natural – not a realistic natural, but a kind of used, worn down, actually-existing-within-a-world-of-entropy kind of natural.

# Mechanics

The mechanics of the game utilizes the following systems:

1. Procedural Generation
2. Voxel-Based Terrain
3. 1st/3rd Person Control
4. Interplanetary and Interstellar Travel
5. Resource Collection and Crafting
6. Real-Time, Server-Based World Persistence
7. PvE and PvP
8. Community-Controlled Social Systems and Economy
9. Non-Intrusive RPG Plotlines and Quests

## Procedural Generation

The galaxy and all planets will utilize Procedural Generation, a process of using consistently applied seeds and algorithms to generate, and later reconstruct, structures and features.

### Galactus Prime

**Galactus Prime will be the test-bed galaxy for all testing of procedural generation. Eventually, multiple galaxies may exist across one or more servers, each representing a shard on which players can play *Brave New Galaxy*.**

**Each galaxy will have its own Primary Key, serving as its unique identifier. In addition, it will also have its own Seed, allowing it to be procedurally generated every time the generation algorithms are run. Galactus Prime will have the following properties:**

* **Primary Key: 0**
* **Name: Galactus Prime**
* **Seed: 20151963**
* **Size: 1024**
* **Type: Elliptical**

When first constructed, the galactic generation algorithm will be used to create a finite number of System Level objects, which are the primary components of the galaxy. Each system will be recorded and maintained in a database table representing the galaxy – this is the only part of the game which will not be dynamically generated at run-time. Each System Level object will have the following properties assigned to it: Primary Key, Seed, System Type, X-Coord., Y-Coord. The procedure for generating these objects, and the participants, will be as follows:

1. Within the 'WORLD\_GENERATION' game state, initialize the GalaxyGenerator director object with an EllipticalGalaxyBuilder builder object assigned to it. The GalaxyGenerator will request the EllipticalGalaxyBuilder to accomplish the following steps:
   1. Initialize GalaxyRoot as the root object for the galactic structure. This object is a composite, as are its children, and will be returned to the game by the GalaxyGenerator once generation is complete.
   2. Set the number of arms, by using a noise function with the galaxy's seed fed into it. The number should be between 1 and 7 inclusive, and should gravitate toward numbers closer to 4. The field armCount will be set with this number. Even though the arms are not necessary for some galaxy types (Elliptical, for instance, or irregular), this value will still be preserved – however, it may be ignored if necessary.
   3. The system's seed and the galaxy's seed will then be used to determine the system's type (Star System, Nebula, Dust Cloud, Black Hole, etc.).
   4. Starting at 0, use another noise function to initialize the characteristics of each system. The noise function should initially be fed the galaxy's seed and the system's primary key to find the system's seed, then the system's seed and an incrementing counter should be used in a noise function to generate random X and Y coordinates, as well as a dice with 100 sides. Each set of coordinates will be compared to the nearest arm to determine the probability of it existing there – if the probability is lower than the dice roll, those coordinates become that system's location; otherwise, the counter is incremented and the next set of coordinates and dice roll are generated.

### Planets

## Voxel-Based Terrain

## 1st/3rd Person Control

## Interplanetary and Interstellar Travel

## Resource Collection and Crafting

## Real-Time, Server-Based World Persistence

## PvE and PvP

## Community Controlled Social Systems and Economy

## RPG Plotlines and Quests

# Game World

## The Galaxy

A procedurally generated galaxy akin to our own Milky Way Galaxy.

## Star Systems

Star Systems can include any number of stars, though higher numbers become increasingly less likely. The following is the current list of objects which can be found within a star system:

1. Star
2. Asteroid Belt
3. Comet
4. Nebula
5. Planet
6. Black Hole

Stars can range from white to red and from super giant to dwarf. A few other star types exist too, including wolf-rayet, neutron, and quasar.

## Planets

The following planets will exist in the game:

1. Terran (Rocky, Earth-like)
2. Gas (Jovan)
3. Icy
4. Ocean
5. Volcanic
6. Carbon

## Ecosystems

Bears

## Races

Humans

Androids

Plants, Gaia

Rock-Based (Basalt?)

Jellyfish

Crustacean-like, Methane breathers

Cuttlefish? (Not playable)

## History

## Technology

## Travel

# Extras

# Resources

# Future Features