



Design Document for:

ENEMIES

The Ultimate Sci-Fi Retro Game

“Back to the Roots, but in 3D!”™



All work Copyright © 2010 by Gamefactor.eu

Written by Robert Judycki

Version # 1.0

Wednesday January 20th 2010

Table of Contents

NAME OF GAME	1
DESIGN HISTORY	4
VERSION 1.00	4
GAME OVERVIEW	5
PHILOSOPHY	5
• <i>Philosophical point #1</i>	5
• <i>Philosophical point #2</i>	5
• <i>Philosophical point #3</i>	5
COMMON QUESTIONS	5
• <i>What is the game?</i>	5
• <i>Why create this game?</i>	5
• <i>Where does the game take place?</i>	5
• <i>What do I control?</i>	6
• <i>How many characters do I control?</i>	6
• <i>What is the main focus?</i>	6
• <i>What's different?</i>	6
FEATURE SET	7
GENERAL FEATURES	7
GAME PLAY	7
THE GAME WORLD	8
OVERVIEW	8
PUZZLES	8
THE PHYSICAL WORLD	8
• <i>Overview</i>	8
• <i>Key Locations</i>	8
• <i>Travel</i>	8
• <i>Scale</i>	8
• <i>Objects</i>	8
• <i>Day and Night</i>	8
• <i>Time</i>	8
RENDERING SYSTEM	9
• <i>Overview</i>	9
• <i>3D Rendering</i>	9
CAMERA	9
• <i>Overview</i>	9
GAME ENGINE	9
• <i>Overview</i>	9
• <i>Game Engine Detail</i>	9

• <i>Water</i>	9
• <i>Collision Detection</i>	9
THE WORLD LAYOUT	10
OVERVIEW	10
GAME CHARACTERS	10
OVERVIEW	10
ENEMIES	10
USER INTERFACE	11
OVERVIEW	11
CHARACTER RENDERING	12
OVERVIEW	12
OBJECTS APPENDIX	13
GRAPHICAL USER INTERFACE APPENDIX	14

Design History

This document is a living document, expect many changes in the future versions as the game development for the ENEMIES continues.

Version 1.0

This is the first version of the design document. Everything added in the document is new.

Game Overview

Philosophy

Philosophical point #1

This game is trying to create a Terrain level with random structures, like Procedural buildings and textures creation that never repeated at the beginning of the game level. Fundamentally I am trying to achieve something that has never been achieved before. The world will be shocked at how we are using an existing engine with new art.

Philosophical point #2

Our game only runs on Windows XP 32-bit platforms or newer Windows Systems. The reason for this is such that the engine can't compiling another systems. Development resources, such as time and lack of ownership of other platforms led me to this decision.

Philosophical point #3

My game design goals is to use (may be) all functions from the engine to this project, like applying a Predefined Surface Shaders, Predefined Postprocessing Shaders or combining with another external shader from the community.

Common Questions

What is the game?

In this game you control the Vehicle character Player. In the center of level sector are player life as energy in capsules with shield barriers where the player must protect them, or you lose.

Why create this game?

I am creating this game because I've always random structures. Since there aren't many games currently on the market that are logical randomized created and I feel this may be an interesting game to create.

Where does the game take place?

This game takes place in terrain level sector with random structures, like Procedural buildings and textures creation etc.

What do I control?

The player will control the character vehicle model who is named iBot.

What is the main focus?

The player's goal of the game is to escape the terrain level sector and activate an Teleporter for teleporting player-life to the next stage also destroy any buildings that collide with player-life platform when the platform is moving, also eliminate enemies before an unknown enemy destroys the barriers and player-life.

What's different?

This game is a 50% top down design game, but can be changed in the Options.

Feature Set

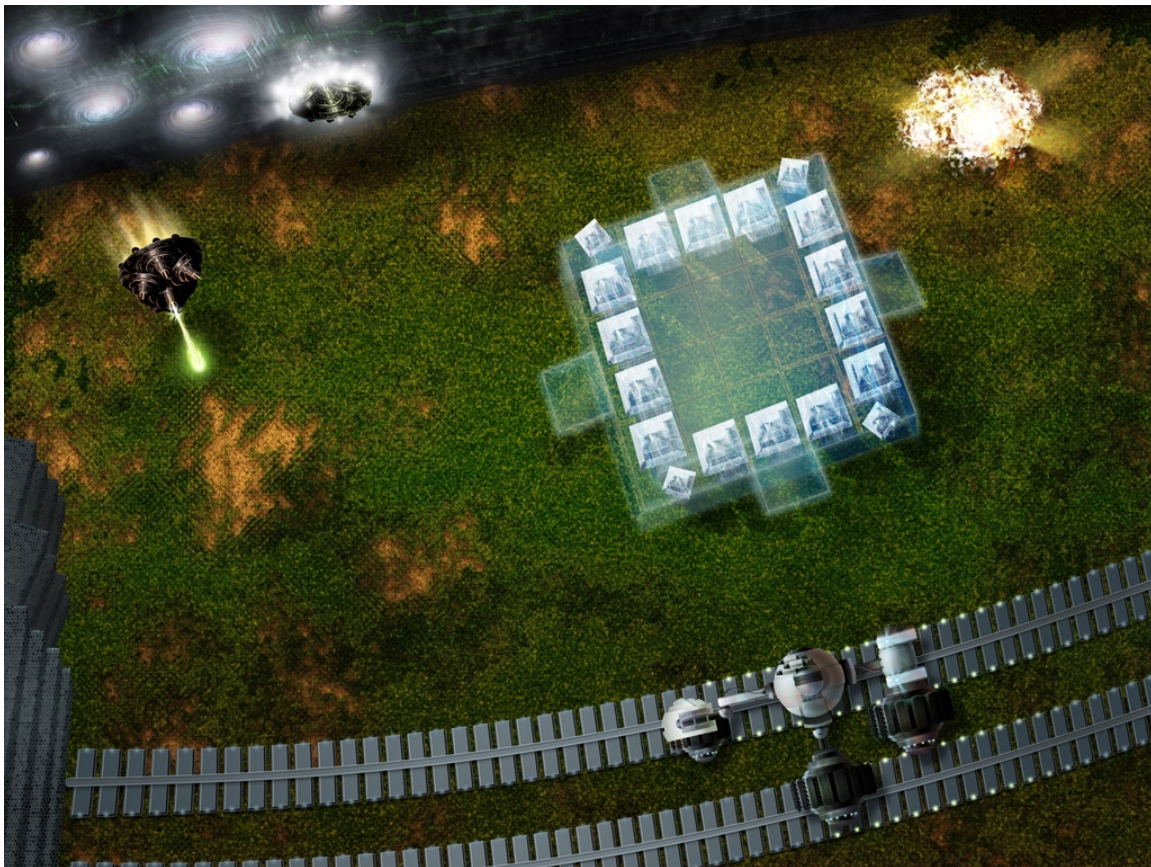
General Features

Huge world
Sci-Fi vehicle model Player
Many various enemies
3D graphics
32-bit color

Gameplay

For some reason you have 3 of your ships placed in the middle of the playing field and Enemies all around them.

Luckily, they're protected by 1 rows of barriers and you're on the bottom, doing your job to protect them before the timer limit expires and the platform of player-life is moved to teleporting place.



The Game World

Overview

The game world is a series of level sectors.

Puzzles

The player will need to control the platform-life. Other sectors will require player to get various elements of the level into specific areas to unlock certain teleporter.

The Physical World

Overview

The physical world is a series of buildings that are located procedurally.

Key Locations

The location is randomly where the player must activate the teleporter.

Travel

Player on a free platform as Sci-Fi vehicle, but with distance moving limit.

Scale

2 cm = 1 Quant

Objects

See the “Objects Appendix” for a list of all the objects found in the world.

Day and Night

Enemies AI attacks should be changed between Day and Night modus.

Time

Before the limit timer expires, teleportation to the next Sector going when the platform of player-life is moved to the teleportation place, otherwise you get lost.

Rendering System

Overview

The rendering speed depends on how many entities, polygons, and pixels are actually drawn on the screen. There are several scene management methods. One of the oldest scene managers was the **Octree** system. Most high-end engines today use a **BSP** tree scene management with a **PVS** visibility set.

3D Rendering

BSP/PVS rendering process will be used.

Camera

Overview

The camera in this game is focused on the side of iBot player.
The camera will move around the iBot player.

Game Engine

Overview

The game will be created using Acknex A7.xx engine in conjunction with lite-c code.

Game Engine Detail

The engines of the Acknex A7 contain functions for 2D and 3D graphics, collision detection, sound, multiplayer, physics, and user interface, together with a scripting language that offers easy access to those functions.

Water

There will be water in the world that looks awesome and our game engine will handle it beautifully.

Collision Detection

The game engine handles collision detection really well. It uses many proven professional techniques for effective collision handling.

The World Layout

Overview

The world will be laid out in a grid based tile system with graphics rendered over the tiles.

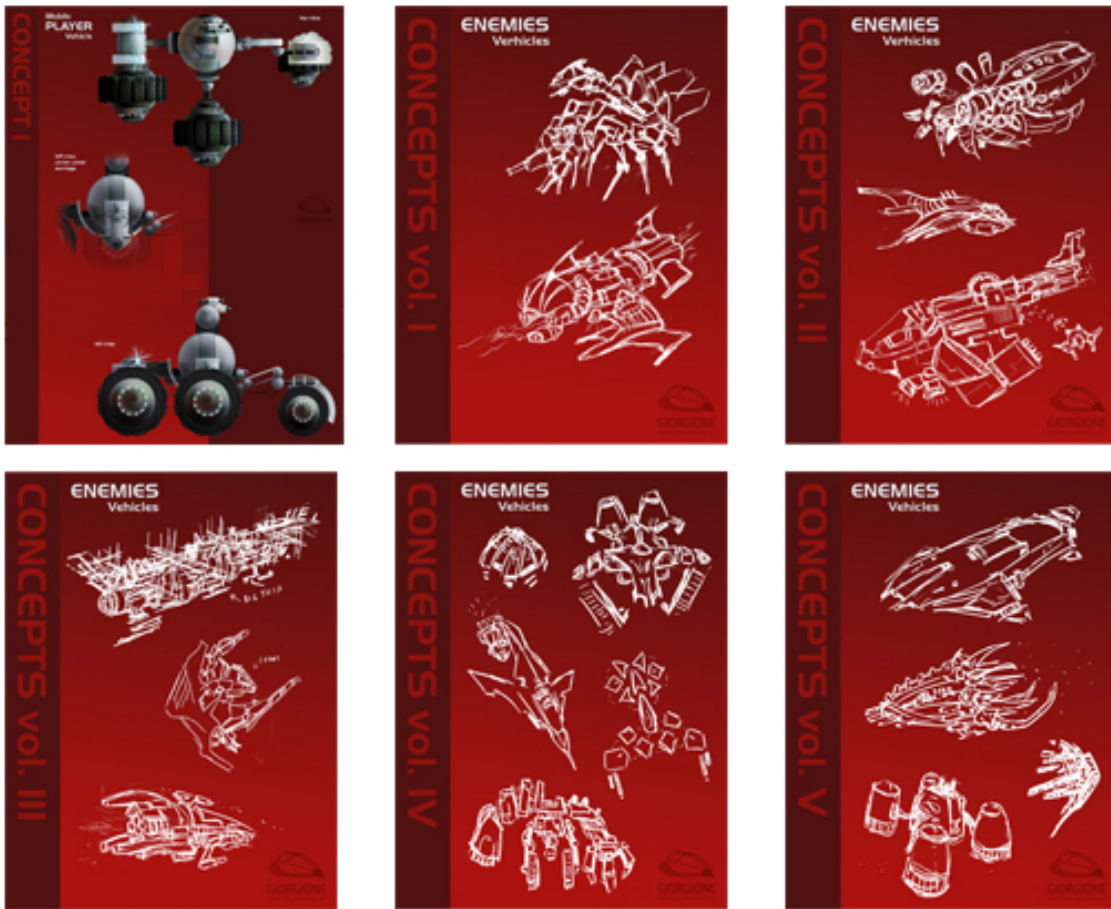
Game Characters

Overview

The main character of this game will be iBot player.

Enemies

The enemies in this game will be robots, ships, spiders or similar with AI, sometimes larger vehicles such as from the "Mars Attack" movie.



User Interface

Overview

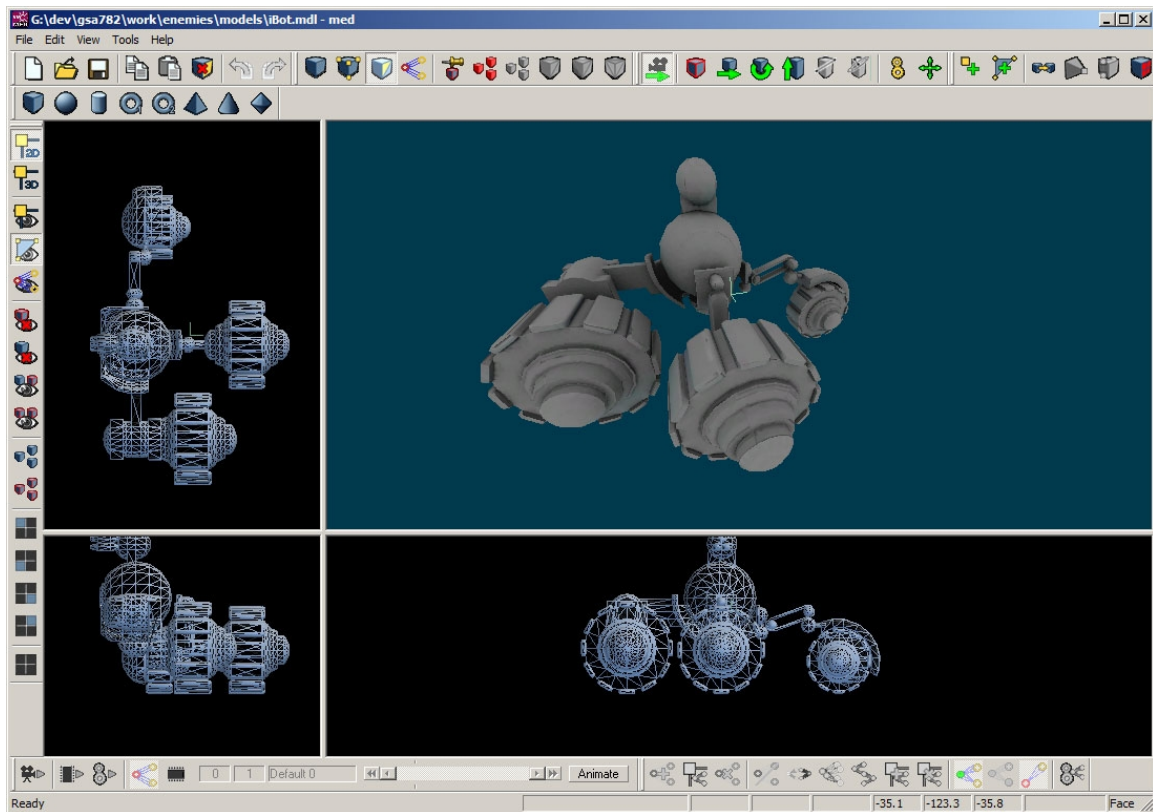
The user interface of the game will be relatively basic. The user can use the mouse or keyboard when traversing the welcome screen to start a game or set game options. All keyboard mapped inputs may be changed by the player in the options screen on the main menu.



Character Rendering

Overview

The character will be rendered in the same way everything else is rendered. It will use DirectX9.0c libraries to enable the use of the video card for quick image rendering.



Objects Appendix



Graphical User Interface Appendix

