

**Game Design Document for:**

# **GUMBO**

A Mission-critical Action Game



by Andreas Mastichis

Written by Andreas Mastichis  
All art designed by Andreas Mastichis.  
Copyright © 2007.

Version # 1.1

# Table of Contents

<b>GUMBO GAME</b>	<b>1</b>
<b>DESIGN HISTORY</b>	<b>4</b>
VERSION 1.0	4
VERSION 1.1	4
<b>GAME OVERVIEW</b>	<b>5</b>
GAME CONCEPTS	5
<i>Game Concept</i>	5
<i>Game Goals</i>	5
<i>References</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>	5
<i>How many characters do I control?</i>	5
<i>What is the main focus?</i>	5
<i>What's different?</i>	5
<b>FEATURE SET</b>	<b>6</b>
GENERAL FEATURES	6
GAMEPLAY	6
<b>THE GAME WORLD</b>	<b>7</b>
OVERVIEW	7
GAME FLOW	7
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>Time</i>	8
RENDERING SYSTEM	9
<i>Overview</i>	9
<i>2D/3D Rendering</i>	9
GAME ENGINE	9
<i>Overview</i>	9
<i>Game Engine Details</i>	9
<i>Collision Detection</i>	9
<b>THE WORLD LAYOUT</b>	<b>10</b>
OVERVIEW	10
STORY	10
<i>Overview</i>	10
<i>Your Mission</i>	10
<i>The Facility</i>	11
LEVEL 1	11
LEVEL 2	11

LEVEL 3	12
<b>GAME CHARACTERS</b>	<b>13</b>
OVERVIEW	13
MAIN CHARACTER	13
ENEMIES	13
<b>USER INTERFACE</b>	<b>14</b>
OVERVIEW	14
OPTIONS	14
MAIN GAME	14
OTHER SCREENS	14
<b>WEAPONS</b>	<b>15</b>
OVERVIEW	15
WEAPON 1 DETAILS	15
WEAPON 2 DETAILS	15
<b>MUSICAL SCORES AND SOUND EFFECTS</b>	<b>16</b>
OVERVIEW	16
SOUND DESIGN	16
<b>POTENTIAL PROBLEMS</b>	<b>16</b>
OVERVIEW	16
AI TECHNIQUES	16
<b>MATHEMATICAL PRINCIPLES</b>	<b>17</b>
OVERVIEW	17
2D TILE ARRAY	17
<b>TESTING</b>	<b>18</b>
OVERVIEW	18
TESTING STRATEGY	18
RESULTS	19
<b>“USER INTERFACE APPENDIX”</b>	<b>20</b>

# Design History

This is for keeping a log of the game design and document.

The design history is intended to keep track of both the game design and document changes. For every change made in either, this section will be updated accordingly.

## Version 1.0

Version 1.0 is the initial version of this document. It is based on the game design document template by Chris Taylor, designer of Dungeon Siege.

## Version 1.1

Version 1.1 includes some additions I did after writing the initial document.

1. Added the level layout documentation.
2. Added mathematical details
3. Added potential problems

Game design: Have added and edited some graphics, mainly action screen -level 1.

# Game Overview

## Game Concepts

### Introduction

Gumbo is a 2D game for the PC that sets a player in a comical combat against enemy forces to reach a goal and be the hero.

### Genre

Gumbo is a 2D arcade action in the style of classic top-down scrolling game.

### Platform

Gumbo is being developed to work mainly on the Windows platform PCs. It requires hardware that support DirectX version 9x.

### References

- Rambo III, the game
- Wolfenstein
- Far Cry

## Common Questions

### What is the game ?

Gumbo is a simple 2D top-down arcade style game featuring original graphics and levels.

### Why create this game ?

This game was created as a Games Development Project for developing a game with a provided game engine as fast as possible.

### Where does the game take place ?

The game takes place on a deserted island somewhere in the south Pacific Ocean in the year 2046.

### What do I control ?

You control the character named GUMBO, a veteran soldier fit for any combat.

### How many characters do I control ?

Only one, GUMBO.

### What is the main focus ?

The focus of the game is to visualize a combat field, action paced and being a hero in the end.

# Feature Set

## General Features

- Small world
- Single Player
- 2D graphics
- 32-bit color
- Sound effects

## Gameplay

Playing Gumbo is a hero's goal to enter the domain of a massive enemy force and use his speed and ability to maneuver around enemy forces to get further into the domain to complete a mission. In combats, characters get killed by shooting each other. The game uses simple animation and effects suitable to ages 8 and up.

# The Game World

## Overview

The game is spanned across 3 levels, each with its own graphics and story. Characters will change only for boss-enemies.

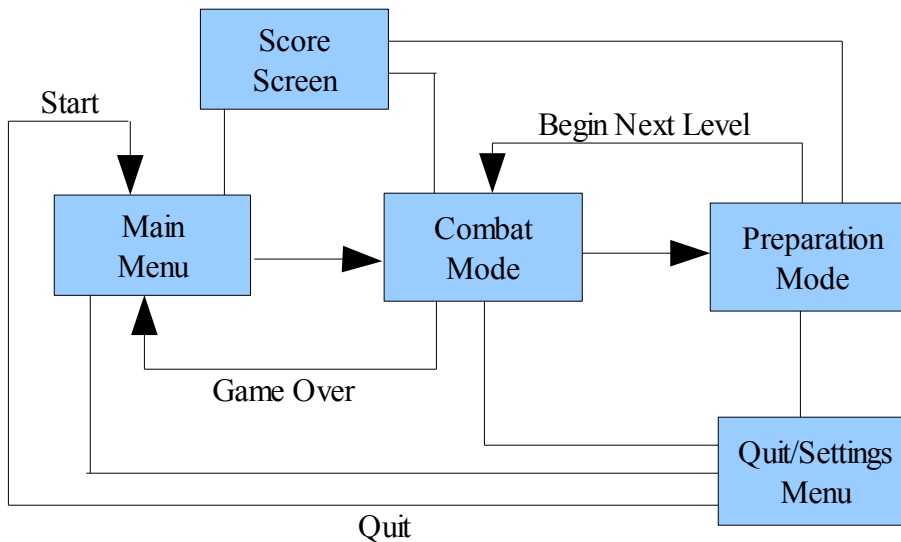
## Gameplay Layout

The main screen is populated with the main character, the enemies he must fight with, a health bar showing how many lives are left for the player and a score bar to show what the player has accomplished counting the kills of the enemy.



Drawing 1: Gameplay Layout

## Game Flow



## The Physical World

### Overview

The world is populated with grass terrain, palm trees, sandy beaches and surrounded by water and the facility being a metal-based building, wide open space rooms and smaller rooms and passages between. These object also depend on the game level.

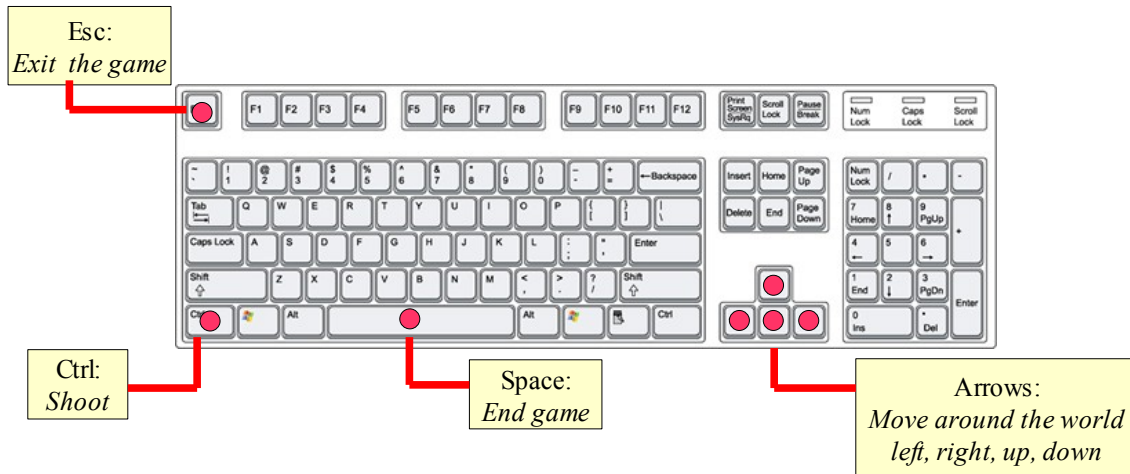
The following describes the key components of the physical world.

### Key Locations

- Seneri island, part of the Tourtouri islands is where you will be dropped off to start the game.
- Your next destination will be the island of Kula where your critical mission will take place on the island and inside the the facility.

### Travel

The player can move the main character using the main arrow keys and is able to shoot enemies using the left Ctrl key. See the illustration below for all keyboard operations:



### Scale

The scale of the game does not change at all levels.

### Objects

Ammo: An important object found throughout the world. Only one type of ammo is available but is used for both the first level weapon and the upgrade one.

Health: Health can be found in areas along the islands and in some cases internally inside rooms of the facility.

### Time

The game has no timely events.

## **Rendering System**

### **Overview**

The game will be rendered using 2D and DirectX will be used to move images and animations.

### **2D/3D Rendering**

The game uses the DirectX3D graphics engine to render all 2D images/graphics and animation. There are no 3D objects in this version of the game.

## **Game Engine**

### **Overview**

The game uses the EGG Framework as the game engine written in C++.

### **Game Engine Detail**

The game engine configures animations, collisions and the graphics rendering that take place in the game world.

### **Collision Detection**

The game engine handles collision detection using pointers. A target would point to the relative coordinates X and Y of a moving object detecting if a collision has occurred or if not.

# The World Layout

## Overview

The game takes place on a far-away island with GUMBO being controlled by the player. Backed-up by a story, the character has a mission spread over 3 levels of gameplay, each in different layouts and enemies to confront. The character must kill enemy opponents to get to his final goal.

## Story

The year is 2046, on a deserted island not well known to the rest of the world in the South Pacific Ocean, the science facility known as Humaxon is where you are to be sent. The facility is run by a private businessman, well-known in the tobacco field. Built underground on one of the tropical South Tourtouri islands (Seneri, Kula and Pali islands), the facility is set on the island of Kula. The facility also hosts a military base to protect the illegal science labs held there. The military base is a force of hundreds of military soldiers armed with guns, explosives and missile weapons for ground and air destruction. The facility have captured 3 scientists to give the intelligence and work in the science labs developing bionic humans. The labs have already constructed a number of bio-humans which are scattered throughout the island. The island itself is a beautiful tropical paradise filled with tall palm trees and sandy beaches surrounded by the waters of the Pacific Ocean. The wildlife on the island has not been harmed in any way, since the facility is an underground establishment. A building also exists above on ground level, built on top of the science facility to keep the it unnoticed. The facility also has smaller research labs spread over the other islands within the 10 to 15 mile distance in reach. The World Research & Discovery center need a well-trained expert to find the scientists and return them to the center!



*Drawing 2: Kula Island*

## Your mission

You are sent by the CWRD(Center of World Research & Discovery) to find the missing scientists that were kidnapped a year ago on the 15<sup>th</sup> of March 2045. Scientists, Prof. Hans Wittgenstein - professor of the discovery sector and microbiologist of the CWRD, Dr. John Miguel - research sector doctor and Dr. Benjamin McDonald – computer science expert and DNA analyzer, who are also the head of CWRD. Without them the research center is not much good and even worse, the world is in great danger if these few genius men are in the hands of the so feared multi-billion man named Zorx Gastroni also known as the “Black Widow” for his dirty business in the tobacco field. The CWRD security agents have informed the center that the scientists are being held on one of the South Tourtouri islands in the South Pacific. Your mission is set clearly to save the three scientists from the hands of the “Black Widow”, and evade them to the CWRD. It is also known but not accurate that the science facility held there on one of the islands, might have already developed clone soldiers to patrol the facility. Be prepared and armed for war, since it is known they armed and dangerous. You will be taken by a boat-plane to the nearest island, Seneri (15 miles from Kula) hopefully undetected by the facility forces, where you need to find your way to the facility and continue your mission to free the scientists and destroy the facility.



*Drawing 3:  
Player  
character*

## The Facility

The facility is hidden and so, very little information is known. What is known is, it is as an underground establishment on one of the islands of Tourtouri (possibly Kula). Some agents have notified that it was seen as a metal-base construction and there are two main entrances the size of a 3 ton vehicle. There are guards all over the facility, so it is hard to enter without being detected unless some secret entry can be found. Nothing else about the facility is known so far.



*Drawing 4: Science Facility Lab*

## Level 1

The first level takes place on the island of Seneri where you will account a layout of an island type terrain. Running along the cost you will fight with the first level of soldiers called “The Clones”. Their resistance is fairly easy since they are only armed with M-16 machine guns similar to what you have at this point. They are not easy targets as they are trained well for combat and they know the island very well. Forwarding no major change should occur until you proceed to the next level which is fantasied on the island of Kula.



*Drawing 5: Game Level 1*

## Level 2

Level 2 is placed on the island of Kula, 15 miles from Seneri island. You will fight against more army forces until you reach the facility itself. You will also play against the first boss-enemy (called “B.E.H.C.” - Bionic Experiment of Human Cell) of the game before you enter the facility building at level

3. This would be a more toughened soldier than what you have already fought with to this point. Armed with a 7mm double chain-gun and armored with a 15mm steel vest, makes this enemy a really hard-to-kill target. A lot of firepower is needed to blast him out and he can not be avoided! Once B.E.H.C has been defeated by your intelligence and strong heart, you will advance to the last and final level 3.

### **Level 3**

The final level is placed again on the island of Kula but inside the facility. You must make your way to where the scientists are held and fight many forces throughout the facility until you also encounter Zorx for the final battle. This level features new objects and gameplay not found in the other 2 levels.

# Game Characters

## Overview

The main character (GUMBO). There are three types of enemies in the game, the Clones, B.E.H.C and Zorx. Each of them feature different powers and abilities.

## Main Character (GUMBO)

GUMBO was designed as a simple 2D pixel graphed image. The character was also designed to be viewed from a 45° angle from screen level.



*Drawing 6:  
GUMBO*

## Enemies

### Overview

Enemies were all designed as simple 2D pixel shaped images and all to be viewed from the same angle as the main character.

### The Clones

These enemies are the most widely found across the entirety of the game world. They are fairly easy to kill when in little forces and have just about the same level of power and armor as the player.



*Drawing 7:  
Clones*

### B.E.H.C

This is the first “boss” of the game. He has most powers of all others, a double chain gun, a heavy armor but has limited speed and is not very smart.



*Drawing 8:  
B.E.H.C.*

### Zorx

This is the final confront the player has to fight with. Zorx has no special features or weapons, but uses all of his forces(Clones & B.E.H.C) to destroy you!

# User Interface

## Overview

The user interface of GUMBO is available in the gameplay and options screens.

## Options

The first interaction with the game will be the Options screen. There are three main selections for the game all used via the keyboard arrow buttons (See: User Interface Appendix):

- Options – shows the options you can change in the game
- Start – will start a New Game from Level 1
- Exit – will Exit the game

## Main Game

The main game screen is where all the action happens and the only things noticeable other than the action are static displays of Health and the current Score. To stop playing or exit the action while playing a player can either hit the ESC key on the keyboard or the SPACE key.



*Drawing 9: Main Game Screen*

## Other Screens

The game also features a GAME OVER screen and a HIGH-SCORE screen to fulfill the completeness of a standard game. (See: User Interface Appendix)

# Weapons

## Overview

There are mainly two types of weapons used in the game. Below is a list.

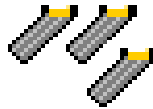
Name	Description	Filename
M-16 Automatic	Automatic machine gun used by both, player's character and Clone enemies	-
7mm Double Chain-gun	Machine gun used by the B.E.H.C. enemy	-

### Weapon 1 Details: M-16 (player, clones)

The M-16 is a fast automatic and light gun to travel with. It is solid and effective making a one shot kill just less than a screen height. Players may find ammo around the game world to refill the weapon and continue their fights against enemies.



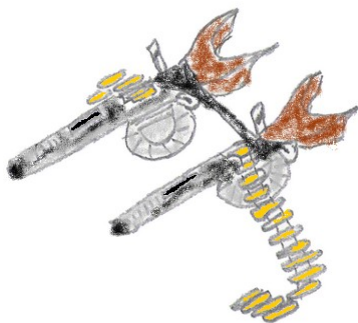
*Drawing 11: Weapon 1  
1 - M-16*



*Drawing 10: Weapon 1  
ammo*

### Weapon 2 Details: 7mm Double Chain-gun (B.E.H.C. only)

This mean weapon is used by the B.E.H.C. enemy and has fast bullet rounds nothing compared with any other weapon faced in the game world. A real killing machine. Players will get killed in an instant if found in it's path.



*Drawing 12: Weapon 2 - Chain-  
gun*

# Music and Sound Effects

## Overview

Sounds in the game are in two types. The background sounds or music and the sound effects.

## Sound Design

Sounds and music background are played everywhere in the game world. Almost every incident has attached sound to create a more realistic environment. The list below is what sounds have been included in the game up to this stage of design.

Name	Description	Filename
Intro	Music played when entering the game	intro.mid
Background	Music played as background while in play	background.wav
Kill	Sound played when either of the character is killed	killed.wav
Shoot	Sound played when either of the character shoots	shot.wav

# Potential Problems

## Overview

Problems occurring in the game at the moment.

## Gameplay

There is possibility that the gameplay intended here may result in problems with character movements when there will be more than just a few. This may be due to wrong use of collisions between them.

## AI techniques

AI of the enemy also will face a problem. When colliding with each other and/or with the main character, they get locked in one position until the game resets or restarts. It may also cause the game to lock up the system.

# Mathematical Principles

## Overview

Use of tiles for to design large screen layouts in a mathematical way.

## What to consider

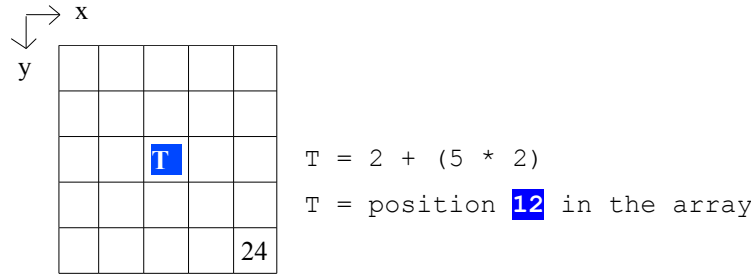
The construction of a level needs to be tiled. This means that every piece of the world we need to show must be designed in little rectangular blocks that build up the world as a whole, like a puzzle. Another thing to consider is at what degree of view will be used. This means if we were to look from above, at what angle the object on the screen should be to be viewed correctly.

## How it can be done

To display tiles on screen, a big array is needed. The following mathematical formula will be needed to calculate the position of a tile:

$$T = X + ( Y * W )$$

This formula would calculate the position at any point within an array. In the following grid, to find the position of a tile placed at the position T, we need to move along the X coordinate 2 tiles, then we add the Y position multiplied by the width of our grid which would be 5 \* 2.



This formula would then be placed inside a function which would serve to display the tiles on screen. The function would be using the x and y calculated from the array and used in a second function to calculate the visual position of every tile. The visual position of a tile would be calculate with the following formula:

$$\begin{aligned} X\text{coord} &= T\text{-xPosition} * T\text{-Width} && \text{-to find the X pixel of tile's position} \\ Y\text{coord} &= T\text{-yPosition} * T\text{-Height} && \text{-to find the Y pixel of tile's position} \end{aligned}$$

## Future Thoughts

An even more complicated method can be used to design an Isometric Tile Display which makes the game layout to look as a more 3D realistic theme. The difference between the above 2D method is that the tiled map is rotated 45 degrees. This makes the x and y axes to be more complexed to work with but the result is what counts.

# Testing

## Overview

Game testing has been carried out to point important errors or bugs in the game and try to fix them in further development.

## Testing Strategy

A risk-based testing strategy will be used in the development life-cycle of this game. Priority will be given to the most important and functional parts in the game that can be released first and leave lesser problems to be solved and released with another version of the game.

Testing will be performed on the highest level of view, i.e. important sections of the game producing crashes will be taken into account first. The areas that can be sensed as 'higher priority' testing can be separated from the whole by specifying which are the most game-critical functions and which are not (an example here can be the main action going on in the game, a critical aspect). Also testing sections that are more vulnerable and may develop more defects than others (i.e. AI and collisions) should be given a priority. The point in using this method of testing is not to avoid any testing but more precisely to build a concrete solution by picking out the most important problems and concentrate on them until completely solved. In this manner the game will reach the deadline to release and all major errors or defects will be eliminated leaving the smaller ones for the next release.

A few points that should be considered a criteria on this testing scheme:

- Overall system testing has been performed
- Code review has been done and defects been fixed
- Unit testing has been performed and meets a standard
- Units have been tested and passed at runtime
- Dry-runs performed on specific detail required cases
- Documentation is complete and of some quality








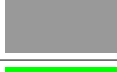





Below is an example table of priority testing and how it will be calculated for the game:

Area to Test	Criticality	Visibility	Complexity	Change frequency	RISK
Weight	3	10	3	3	
Collisions	5	4	5	3	79
AI	3	3	5	1	57
Graphics	3	5	3	1	69
Animations	3	5	5	1	77




In the above table, to calculate the risk analysis of every factor (grey cells) we just set the weight of every factor and then multiply each area (purple cells) of testing by every factor's weight (yellow cells) to get the priority risk factor. The highest number is the most important. As we can see here, collisions is the priority for testing since it has the highest risk factor (79). This of course depends of what weight we will give for every factor.

I believe for rapid application development, this method is fast and efficient creating a more solid development and relieving stress from requiring a full testing scheme which may never reach its completeness.

## Results

Type	Description	Result	Expected	Test
Game	Game starts, combat begins	Started, combat active	Start & combat	
Game	Reset Game after player killed	Game not reset	Game resets	
Game	Compatibility Test	Only tested on WinXP	Test on various type of platforms	
Graphics	Load graphics on screen	Loading	Load	
Bullet	Bullets collide with characters	Colliding	Collide	
Bullet	Bullet animates	Animating	Animate	
Main Character	Characters move within limited screen	Move within limited screen	Move within limited screen	
Enemy Characters	Characters move within limited screen	Move within limited screen/ can move outwards	Move within limited screen	
Characters	Characters collide with each other	Colliding	Collide	
Characters	Characters make sounds when killed	Making sounds when killed	Make a sound when killed	
Characters	Characters make sounds when shooting	Making sounds when shooting	Make a sound when shooting	
Characters	Characters animate	Animating	Animate	
AI	AI of enemies move in specific order	Move in wrong order/ lock up	Move in specific order	

### Legend:

	Successful
	Not fully tested
	Unsuccessful

## “User Interface Appendix”

The user interface design screens:



*Drawing 13: Options Screen*



*Drawing 14: High Scores Screen*



*Drawing 15: Game Over Screen*