

**Design Document for:**

# Singularity

**One Liner, i.e. The Ultimate Racing Game**

“Something funny here!”™

All work Copyright ©2017 by Group Name

Written by Group Name

Version # 1.00

Monday, October 10, 2022

Table of Contents

Name of Game 1

Game Overview 4

Common Questions 4

What is the game? 4

Why create this game? 4

Where does the game take place? 4

What do I control? 4

How many characters do I control? 4

What is the main focus? 4

What’s different? 4

Feature Set 5

General Features 5

Gameplay 5

The Game World 6

Overview 6

World Feature #1 6

World Feature #2 6

The Physical World 6

Overview 6

Key Locations 6

Travel 6

Scale 6

Objects 6

Weather 6

Day and Night 6

Time 7

Camera 7

Overview 7

Camera Detail #1 7

Camera Detail #2 7

Game Engine 7

Overview 7

Game Engine Detail #1 7

Water 7

Collision Detection 7

The World Layout 8

Overview 8

World Layout Detail #1 8

World Layout Detail #2 8

Game Characters 9

Overview 9

Creating a Character 9

Enemies and Monsters 9

User Interface 10

Overview 10

User Interface Detail #1 10

User Interface Detail #2 10

Weapons 11

Overview 11

Weapons Details #1 11

Weapons Details #2 11

Musical Scores and Sound Effects 12

Overview 12

Red Book Audio 12

3D Sound 12

Sound Design 12

Single-Player Game 13

Overview 13

Single Player Game Detail #1 13

Single Player Game Detail #2 13

Story 13

Hours of Gameplay 13

Victory Conditions 13

# Game Overview

## Common Questions

### What is the game?

Describe the game in a paragraph. This is the answer to the most common question that you will be asked. What are you working on?

This game is primarily a first-person shooter. The player controls a robot and uses their weapons to defeat several human enemy non-player characters (NPCs) who are out to defeat the player. These enemy characters will use different algorithms to assist them in their objective and will have a different arsenal of weapons. In addition to a first-person shooter, the game will employ an aggressive movement engine, allowing for high-speed gameplay.

### What is the target audience?

Who is going to play your game? What age bracket? Tell us details about your audience.

This game is designed for players who enjoy shooter games with an emphasis on high-speed combat and fluid movement. The expected age bracket for this game is 15 – 22. This is because the game revolves around gun violence, so it is expected that mature teens and adults will play the game.

### Why create this game?

Why are you creating this game? Do you love 3D shooters? Do you think there is a hole in the market for Jell-O tossing midgets?

Our team is creating this game because we enjoy the feel of satisfying movement combined with the skill-based aspect of a shooter game. The fast-paced nature of the game keeps the game-play interesting. We also believe that players may enjoy playing a man versus machine game, where the player takes the side of the machine.

### Where does the game take place?

Describe the world that your game takes place in. Simple as that. Help frame it in the reader’s mind by spending a few sentences on it here. You can go into lengthy detail later in a section solely dedicated to describing the world. Remember that we want to keep this part of the design document light and readable.a

### What do I control?

Describe what the player will control. You will be in charge of a band of rabid mutant fiddle players. If you want you can switch on the AI and turn it into a fish bowl simulation.

The player controls a robot.

### How many characters do I control?

If this applies talk a little more about the control choices. Remember to add answers to questions that you think the reader will ask. This is totally dependent on your design.

The player controls a single robot.

### What is the main focus?

Now that we know where the game takes place and what the player controls. What are they supposed to achieve in this world? Angry fiddle players take over the U.N. building. Be careful not to add a bunch of salesmanship here. Your design document wants to stay light and informative.

In this game the player controls a robot character. The objective of this robot player is to wipe out the human enemy NPC’s so that the robots can achieve the singularity – the point in time where technological growth becomes faster and uncontrollable.

### What’s different?

Tell them what is different from the games that are attempting this in the market right now. This question comes up a lot.

This game is different in the way that is uses aggressive movement inside a single-player shooter game. Most single-player games in the market focus heavily on the story, such as Cyberpunk 2077 and Half-Life Alyx. On the other hand, most popular multiplayer shooters focus heavily on slow-precise gameplay such as Valorant or Escape from Tarkov. By focusing on having an aggressive movement engine, the player gets to enjoy high energy fun.

# Feature Set

## General Features

Huge world

Mutant fiddle players

3D graphics

32-bit color

Huge Map

First Person Shooter

3D Graphics

Full-Color

Main Menu

Settings

Sound effects

Music

Robot playable character

Human enemy NPCs

Pause menu

Character selection ??

## Gameplay

List stuff here that is key to the gameplay experience

List a lot of stuff here

Hey, if you got nothing here, is this game worth doing?

Shooting

Several different weapons

Sliding

Double jumping

Wall running

Wall jumping

Interactable environment

HUD showing health and ammo

Reloading

Slow down time ability ???

## Flowcharts

Add Flowcharts here

## Storyboards

Add Storyboards here

## Controls

What are the controls

W – Move Forward

A – Move Left

S – Move Back

D – Move Right

CTRL – Crouch/Slide

ESC – Pause menu

E – Interact with environment / Pickup weapon

G – Drop weapon

SPACE – Jump

# Single-Player Game

## Overview

Describe the single-player game experience in a few sentences.

Here is a breakdown of the key components of the single player game.

## Single Player Game Detail #1

## Single Player Game Detail #2

## Story

Describe your story idea here and then refer them to an appendix or separate document which provides all the details on the story if it is really big.

The story for this game is that the player is a robot trying to bring forth the technological singularity. It is essentially robots conquering over humans. The robot’s task is to wipe out the human civilization so that robots can take over.

## Hours of Gameplay

Talk about how long the single-player game experience is supposed to last or what your thoughts are at this point.

The single-player game experience is fast. With the minimal focus on storyline, the game shouldn’t take more than 30 minutes depending on number of levels implemented. Players are dropped into the game and have all the controls from the get-go.

## Victory Conditions

How does the player win the single-player game?

The player wins by defeating all the NPC’s and reaching the end goal.

# Artificial Intelligence

## AI Algorithms

The two different AI algorithms that are used in the game are: **Finite State Machine (FSM),** and **Navmesh** **Pathfinding.**

Finite State Machine

Finite state machine represents an NPC behavior framework that defines what the NPC does at any given time. These behaviors are classed into different ‘states’ and are used whenever a set condition is met. All the enemy NPCs in singularity employ this framework.

An example of how FSM is used in singularity follows.

|  |  |  |
| --- | --- | --- |
| State: | **Entry Condition** | **NPC Behavior** |
| **None** | Default – enemy spawns in this state | None, stands on the spot |
| **Chase** | Player is within chase range | NPC destination position is set to players current position |
| **Attack** | Player is within attack range | Deal damage at 5hp/second to player |
| **Frenzy** | Health is less than 50% | 5 second bullet wave (invulnerable during this time) followed by 5 seconds chase. Repeated until player is dead or helath is less than 0. |
| **Dead** | Health is less than 0 | Enemy object destroyed. |

Navmesh Pathfinding

Navmesh pathfinding is an AI algorithm used to determine a path between two positions that is walkable and avoids obstacles. The game engine examines the terrain, props, and other environment features of the game level, and draws a mesh of convex polygons that represents areas that are agent-traversable, meaning that it is suitable and realistic for an agent to walk/jump in that area. This is used by all non-stationary NPCs in singularity.

## AI Diagrams

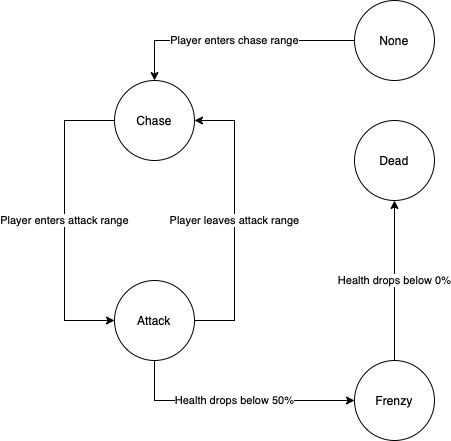


Figure : FSM diagram representing the behavioural states of Boss NPC

# The Game World

## Overview

Provide an overview to the game world.

The map is medium sized post-apocalyptic urban city environment which is set in the near future. The fog in the city washes out the color of what the city once was and further encompassing the desertion and isolation of the city. The player can easily navigate through the city by being guided by landmarks and obstacles, the player will go and discover buildings, as they travel to the end point.



## Roads

The road is a t-section which was made by a number of prefabricated pieces. E.g. straight roads, corners and junctions.



## Barbed Fences

These fences are made

## The Physical World

### Overview

Describe an overview of the physical world. Then start talking about the components of the physical world below in each paragraph.

The following describes the key components of the physical world.

### Key Locations

Describe the key locations in the world here.

### Travel

Describe how the player moves characters around in the world.

### Scale

Describe the scale that you will use to represent the world. Scale is important!

### Objects

Describe the different objects that can be found in the world.

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

### Weather

Describe what sort of weather will be found in the world, if any. Otherwise omit this section. Add sections that apply to your game design.

## Camera

### Overview

Describe the way the camera will work and then go into details if the camera is very complicated in sub sections.

The camera is attached to the player character. As the player moves their mouse cursor around, the camera will move, but the cursor will stay in the middle of the screen, acting as a crosshair.

### Camera Detail #1

The camera will move around like this and that.

### Camera Detail #2

The camera will sometimes move like this in this special circumstance.

## Game Engine

### Overview

Describe the game engine in general.

### Game Engine Detail #1

The game engine will keep track of everything in the world like such and such.

### Water

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

### Collision Detection

Our game engine handles collision detection really well. It uses the such and such technique and will be quite excellent. Can you see I am having a hard time making up stupid placeholder text here?

# The World Layout

## Overview

Provide an overview here.

## World Layout Detail #1

## World Layout Detail #2

# Game Characters

## Overview

Over of what your characters are.

## Creating a Character

How you create or personalize your character.

Use character selection at start of game ??

## Enemies and Monsters

Describe enemies or monsters in the world or whomever the player is trying to defeat. Naturally this depends heavily on your game idea but generally games are about trying to kill something.

There will be three types of enemies: Ranged, melee and boss.

Ranged enemies: using FSM, they patrol an area until player enters shooting range, they shoot while standing still until player either leaves the shooting range (back to patrolling), or they get killed/kill the player.

melee enemies: Constantly chasing player even when out of view. They use A\* to find the shortest path to the player. If they get within melee range, they attack and the game is lost.

Boss: The boss enemy uses FSM like the ranged enemies, the difference being that it has more health, does more damage and attacks faster.

# User Interface

## Overview

Provide some sort of an overview to your interface and same as all the previous sections, break down the components of the UI below.

## Main Menu

The Main Menu uses simple, easy to read buttons that tell the user where they lead. While the buttons aren’t immediately obvious, if the user hovers the mouse over the text, it displays a highlighted color which illuminates that the text is interactable.

The user can use this main menu to go to the game, settings or quit. If the player clicks the game button it loads the game scene. If the player clicks the quit button, the application will close. If in the editor, the console will print the words “QUIT”.

If the user clicks the settings button, the settings menu becomes enabled and the main menu becomes disabled.

Within the Main menu is the Settings UI. This Menu has a back button and a volume slider. The back button will re-enable the main menu and disable the settings menu. The volume slider is used to adjust the volume of the game music and effects.

## Pause Menu

The Pause menu uses similar buttons to the Main menu, contributing to the look and feel of the game. When the player presses the “esc” key, the pause menu is enabled. This plays an animation that fades in a grey see-through filter over the game and pauses the time scale so that the game doesn’t keep playing. The pause menu has a “Resume”, “Menu” and “Quit” button. Resume turns the pause menu off so that the game keeps running. The quit button closes the application but if done in the editor, prints “QUIT”. The menu button loads the scene of the Main Menu.

# Weapons

## Overview

Overview of weapons used in game.

## Weapons Details #1

Pistol – Single shot weapon. Semi auto. 8 shots. 10 damage.

Rifle – Fully automatic weapon. 30 clip magazines. 15 damage.

Shotgun – Burst shot weapon not completely accurate. 2 Shots. 30 damage.

## Weapons Details #2

# Musical Scores and Sound Effects

## Overview

This should probably be broken down into two sections but I think you get the point.

## Red Book Audio

If you are using Red Book then describe what your plan is here. If not, what are you using?

## 3D Sound

Talk about what sort of sound APIs you are going to use or not use as the case may be.

## Sound Design

Take a shot at what you are going to do for sound design at this early stage. Hey, good to let your reader know what you are thinking.