Game Design Document (GDD)

Game Title: 404 - Survival Not Found

Genre: 2D Survival Adventure

Platform: PC

Libraries: Java (Swing and AWT)

1. Game Overview

Game Description:
"404 - Survival Not Found" is a 2D Java-based survival game where players control a hero struggling to survive in a dynamically generated map. The game blends exploration, strategic movement, and fast-paced combat, offering an engaging pixel-art experience.

Key Features:

- **Dynamic Survival**: Increasingly difficult waves of monsters as you progress.
- **Environmental Hazards**: Avoid trees, spikes, and other obstacles that impede your survival.
- **Immersive Atmosphere**: Retro-inspired pixel art graphics and an intuitive interface.
- **Progression**: Level up your hero's health and speed to face stronger challenges.

2. Core Gameplay

Objective

Survive as long as possible by defeating waves of monsters while navigating obstacles.

Player Abilities

- **Movement**: Controlled via arrow keys for precise movement.
- **Combat**: Utilize weapons like speed rifles and assault rifles to fend off enemies.
- **Level Progression**: Gain power-ups that increase speed and health regeneration over time.

Game Loop

1. Navigate through a randomly generated map populated with enemies

and obstacles.

2. Engage in combat with progressively difficult monsters.
3. Avoid or use environmental elements strategically to survive.
4. Advance to higher levels with stronger enemies and hazards.

3. Characters and Entities

Hero

- **Description**: A lone survivor equipped with weapons and determined to fight against the odds.
- **Attributes**: Health, speed, and combat ability.

Monsters

- **Types**:
- **Melee Zombies**: Close-range attackers with high health.
- **Ranged Zombies**: Fire projectiles from a distance.
- **Behavior**:
  - Chase and attack the hero.
  - Spawn in waves, becoming stronger over time.

Environment

- **Obstacles**:
  - Trees and bushes block movement.
  - Spikes cause damage when stepped on.
- **Aesthetic Elements**: Rocks, decorative items for immersion.

4. Art Style and Visual Design

- **Graphics**
  - Retro pixel art style for characters, monsters, and environments.
  - Distinct visual cues for obstacles and power-ups.
- **User Interface**:
  - Simple and unobtrusive, showing essential stats like health and remaining enemies.

5. Game World

- **Map**:
  - Dynamic and procedurally generated to ensure unique play throughs.
  - Balances open areas for combat and tight spaces for strategic movement.
- **Themes**:

- Post-apocalyptic aesthetic with ruined terrain and eerie environments.

6. Controls

- **Arrow Keys**: Move the hero.
- **Space**: Attack or shoot weapon.
- **Shift**: Sprint (if implemented).

7. Technology Stack

- **Language**: Java
- **Libraries**: Swing and AWT for graphical and UI elements.
- **IDE**: IntelliJ IDEA
- **Version Control**: Git

8. Development and Implementation

Key Modules

- **Player Module**: Handles hero actions, animations, and stats.
- **Enemy Module**: Manages different monster types and behaviors.
- **Environment Module**: Generates obstacles and procedural maps.
- **Combat System**: Implements weapon mechanics and damage calculations.

Project Structure

- src/: All source code.
  - item/: Decorative elements like trees and rocks.
  - main/: Core game logic (e.g., Main.java, GamePanel).
  - player/: Hero and monster classes.
  - weapon/: Classes for various weapons.
- res/: Graphics resources (e.g., images for the hero, monsters, and environment).