ONE PIECE GAME

5 march 2025

Software Development

Table of Contents

[Introduction 3](#_Toc192458500)

[Game Concept and Story 4](#_Toc192458501)

[Concept Summary: 4](#_Toc192458502)

[Story/Setting: 4](#_Toc192458503)

[Gameplay Mechanics 5](#_Toc192458504)

[Core Gameplay 5](#_Toc192458505)

[Player Actions 6](#_Toc192458506)

[Character Mechanics 6](#_Toc192458507)

[Level Design 9](#_Toc192458508)

[Art, Animation, and Audio 9](#_Toc192458509)

[Technical Architecture and Tools 10](#_Toc192458510)

[Multiplayer and Online Components 11](#_Toc192458511)

[Testing and TDD 12](#_Toc192458512)

Introduction

One Piece Pixel Showdown is an engaging 2D pixel art multiplayer game that brings the iconic world of pirates and marines to life. Players choose a side Pirates or Marines and initially select one representative character per team. Each character features a unique set of basic attacks, special moves, and energy boosts that capture their signature style.

The game is structured into three levels: on Levels 1 and 2, each side faces distinct challenges and enemy encounters that hone their abilities and set the stage for the final confrontation. In Level 3, players connect online to battle head-to-head in a real-time duel, testing their mastery of timing, strategy, and character skills.

Built in Java with frameworks like libGDX for graphics and KryoNet for networking, the game offers smooth animations, responsive controls, and scalable multiplayer functionality. This modular design not only supports a robust initial experience but also lays the groundwork for future expansions with additional characters and refined mechanics.

Game Concept and Story

## Concept Summary:

Players take on the role of either a Pirate or a Marine. As a Pirate (Luffy), you start on level 1 facing waves of enemy pirates before encountering powerful bosses. As a Marine (Akainu), you battle against pirates with a different set of challenges and bosses. The story leads to an online duel on level 3 where both sides face off directly.

## Story/Setting:

**Pirate Side (Luffy):**

**Level 1:** You spawn at the left side (x=0, y=0) in a tutorial area with overhead blocks and a ground. A brief tutorial introduces basic moves and button mappings. Once complete, enemy pirates spawn in structured formations (e.g., two on an upper block, two on a nearby block, followed by three that attack sequentially).

**Mechanic Highlight:**

You can grab a pillar using the Gomu Grab move, launching yourself (Gomu Rocket) so that a headbutt counts as three punches. Pirates require four punches each to be defeated.

**Boss Encounter:**

After the pirate waves, you face the first boss, Arlong. Arlong uses a water shot special move that can freeze you briefly; you must land ten hits to defeat him. When your HP falls to 20%, you trigger Gear 2, boosting your speed and doubling your punch strength.

**Level 2:** You progress to a new map where you encounter the second boss, Kuma. Kuma attacks with lasers, explosive radius moves, and engages in a brief scripted conversation with Luffy before the fight. Both you and Kuma can use a counter move (Armament Haki) while idle to reduce damage.

**Additional Move:** After Level 1, you unlock a grab move to catch your opponent and perform a Gomu Rocket. Note that after using this move, movement is temporarily disabled and it has a cooldown.

**Level 3 (Online Final Duel):** After defeating Kuma, you transition to Level 3, where you meet an online opponent. This final duel pits the refined skills from earlier levels against another player.

**Marine Side (Akainu):**

Akainu’s progression mirrors Luffy’s with tailored enemies and bosses.

**Level 1:** You face pirate enemies and your first boss is Crocodile, a devilfruit user that dissolves into sand and can quickly reposition. Being caught in his sand inflicts drowning and additional damage.

**Level 2:** Akainu faces Ace, a fast moving opponent who can fly, shoot fire, and generate tornado like effects.

**Level 3:** You meet Luffy online. Though both are equally matched, subtle differences—Akainu’s raw strength versus Luffy’s agility—determine the flow of battle.

**General Rule:**

If a player dies during a level, they respawn on that same level. However, dying on Level 3 results in a loss.

Gameplay Mechanics

## Core Gameplay

**Tutorial Phase:** Learn basic moves (movement, jumping, attacking) in a dedicated area.

**Combat Encounters:** Face waves of enemies that gradually increase in difficulty.

**Boss Battles:** Encounter challenging bosses whose special moves require strategic counters.

***Online Duel:*** *After completing Levels 1 and 2, face an online opponent in Level 3.*

## Player Actions

**Movement:**  
Walk forward, backward, and jump to navigate through levels.

**Combat Actions:**  
Execute basic attacks, special moves, and counter moves based on timing and combos.

**Advanced Maneuvers:**  
Utilize character-specific mechanics such as Luffy’s Gomu Grab to manipulate the environment or reposition during combat.

**Power-Ups:**  
Activate temporary energy boosts to enhance combat performance.

**Interaction:**  
Grab environmental objects or enemies, integrating environmental strategy into combat

## Character Mechanics

#### ****Akainu (Marine Side)****

**Basic Attack:**  
**Lava Fist** – A strong, direct punch infused with magma.

**Special Move:**  
**Magma Explosion** – A charged attack that releases a burst of lava over a small area, dealing heavy damage.

**Counter Move:**  
**Armament haki** – A defensive stance that minimizes the damage from incoming special moves.

**Additional Mechanic:**  
**Ground Melt** – His attacks briefly alter the terrain by creating hot zones that slow down opponents.

**Energy Boost:**  
**Magma comet** – throws a few huge comets on opponent.

**Idle Behavior:**  
When inactive, he exclaims “Justice burns eternal!”.

#### ****Luffy (Pirate Side)****

**Basic Attack:**  
**Gum-Gum Punch** – A stretchy, powerful punch that can hit from a distance.

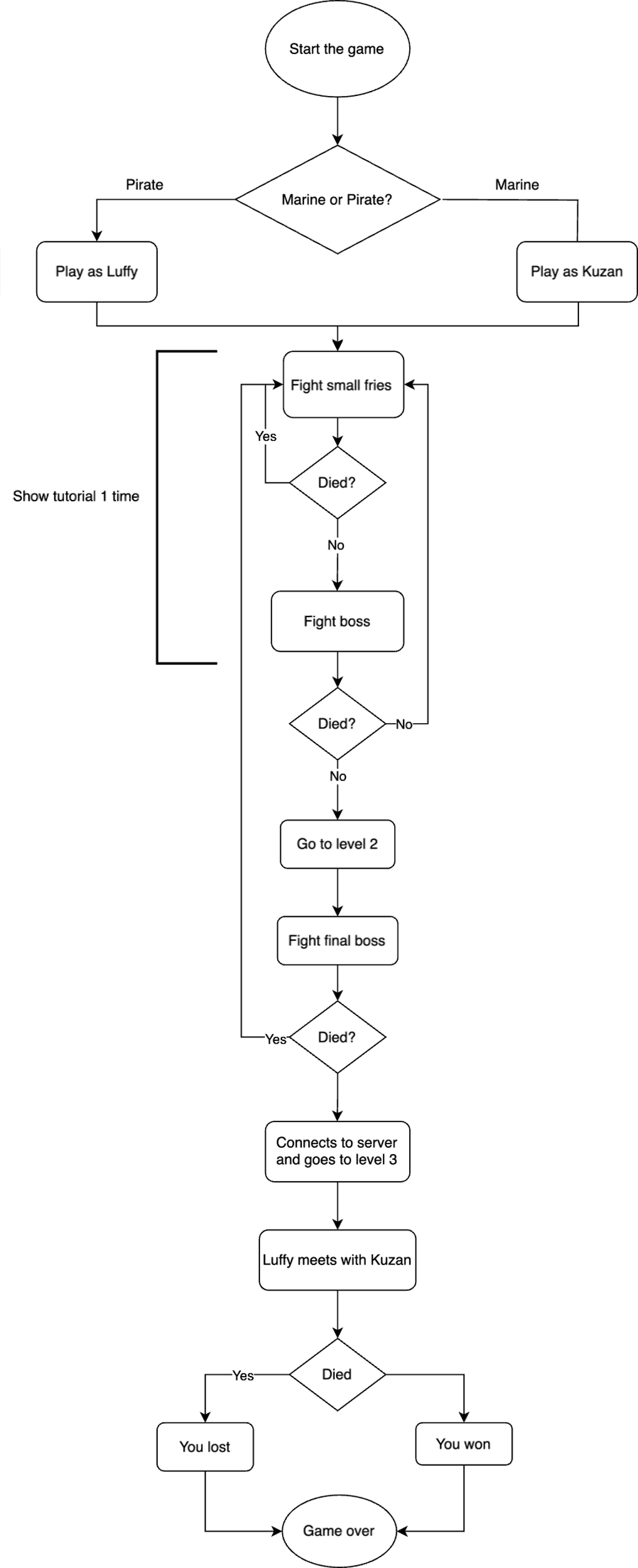
**Special Move:**  
**Gomu Gomu no Pistol** – After multiple button presses, he unleashes a long-range, high-impact punch.

**Counter Move:**  
**Armament haki** – A defensive stance that minimizes the damage from incoming special moves.

**Additional Mechanic:**  
**Gomu Grab** – Can grab enemies or objects; if the grabbed object is immobile, he can launch himself forward using his Gomu Rocket.

**Energy Boost:**  
Gear 3: Triggers when HP is low, making big fist and landing stronger punch.

**Idle Behavior:**  
When idle, he shouts “I'm hungry!” after a shoperiod of inactivity.



# Level Design

**Level 1:**

**Environment:** A tutorial area with a ground and overhead blocks.

**Enemy Placement:** Tutorial followed by strategically placed pirates.

**Key Object:** A pillar that can be grabbed to execute the Gomu Grab move.

**Boss Encounters:**

Pirate side: Boss Arlong appears with water shooting and freezing special moves.

Marine side: Boss appear with unique mechanics (e.g., dissolving sand and rapid movement).

**Level 2:**

**Environment:** A new map with increased challenges and dynamic obstacles.

**Boss Encounters:**

Pirate side: Boss Kuma appears with laser shooting and explosions as special moves.

Marine side: Boss Ace appear with unique mechanics like flying, fire shooting and tornado.

**Level 3:**

**Online Arena:** A final battle area designed for real-time multiplayer duels.

**Rule:** Death in this level results in an immediate loss.

Art, Animation, and Audio

* Visual Style:

2D pixel art with vibrant, animated characters and detailed environments.

Use sprite sheets to create smooth animations for all actions (idle, move, attack, special moves).

* Animation Techniques:

Implement animations via libGDX’s Animation class.

Ensure each action has a dedicated animation sequence synchronized with game logic.

* Audio Design:

Music: Background tracks that match the thematic setting of each level.

Sound Effects: Custom sounds for attacks, special moves, boss abilities, and environmental effects.

Technical Architecture and Tools

**Development Environment:**

**Language:** Java

**IDE:** Eclipse

**Build Tools:** Maven or Gradle

**Frameworks & Libraries:**

**libGDX:** For game development, rendering, and animation.

**KryoNet:** For multiplayer networking.

**JUnit:** For test-driven development (TDD).

**Data Persistence:**

Save game progress (e.g., current level, player HP) using lightweight file storage (JSON or similar).

**Version Control:**

Use Git for managing code, with a clear branching strategy to support team collaboration.

Multiplayer and Online Components

**Networking Architecture:**

**Server-Client Model:**

Use a dedicated server application (via KryoNet) to manage connections, synchronize game state, and relay actions between players.

**Matchmaking:**

Implement a lobby where players wait until both sides are connected before entering Level 3.

**State Synchronization:**

The server maintains the authoritative game state (player positions, moves, HP) and regularly broadcasts updates.

**Latency Handling:**

Use interpolation and buffering techniques to ensure smooth gameplay despite network lag.

Testing and TDD

**Testing Approach:**

* Use JUnit to write tests for each module (game logic, networking, UI components).
* Follow TDD principles: write tests first and then implement functionality.
* Maintain clear commit logs and document each team member’s contributions.