Tortuga Game Design Document

By Kayne Ruse

Introduction 2

Monetization 2

Gameplay 3

Movement 3

Combat Instances 3

Combat Mechanics 3

Permadeath 3

Monsters 4

Items 4

Equipment 4

The Game Map 5

Dungeons 5

Ruins 5

Towers 5

Forests 5

Caves 5

Legendary Dungeons 5

Server Management 6

Player Accounts 6

Player Characters 6

Todo List 7

# Introduction

Tortuga is an open source 2D multiplayer rogue like role playing game; the game runs on customizable player run servers. This project is currently independently created and funded, with the goal of releasing an early version for alpha funding and incorporating community feedback.

I have a long road ahead of me, but one of the first things I need to do is to complete a draft of this document. So, if you see any italic text, you can consider that to be an incomplete or removed section of text or an inline comment.

I’ll also need to review this for grammar, etc.

# Monetization

I’d like to develop this game as an open source project until I reach alpha, at which point I’ll run a kickstarter and begin taking preorders. The preorder system will work just like Minecraft’s preorders: players can buy the game during alpha for 50% of the initial price, or 75% during beta. The kickstarter will be used partially to raise awareness of this project during the beginning of the alpha.

Anybody who has purchased this game has access to all of the development files, as well as a license to modify and redistribute the game to anybody else who has also purchased the “vanilla” (unmodified) version. This is to encourage mods and customisation of the game.

# Gameplay

## Movement

The game is a top down RPG, so the players will walk around the 2D world, while being unable to pass through solid objects (including walls, water, etc). The player can enter/exit areas by colliding with door objects. These different areas include buildings, dungeons and combat instances. There will also be the possibility of teleportation at some point.

## Combat Instances

Combat takes place in specialized instanced rooms, known as combat portals. The combat mechanics are reminiscent of early Final Fantasy instalments, with two sided combat using an active time battle system.

Each combat instance is created in the overworld or a dungeon at random, and is located where the players encountered it until that instance is complete. Players can also create combat instances purposefully by attacking another player; this pits these two players against each other.

Players are capable of entering pre-existing combat instances from the game world (via combat portals), and individual players already in combat can exit that combat (by selecting the run command).

## Combat Mechanics

Tortuga’s combat consists of a unique drop-in/drop-out multiplayer active time battle system.

Battles that a player can participate in will be visible in the game world as combat portals. The combat system will allow several people to fight side by side as a group. Each player will be able to choose their own actions, including attacking with their equipped weapons, using spells or items, or fleeing the battle. To prevent a player from taking too many actions too fast, that player must wait for their ATB gauge to fill completely before taking an action, after which it resets to zero. The speed at which a character’s ATB gauge refills is affected by that character’s statistics.

The biggest innovation of this design is the drop-in/drop-out system. Since permadeath is such a massive aspect of the game, a player must have the ability to flee a battle at any time. Fleeing a battle causes a player to take a penalty such as losing money, experience points, or more. Leaving a battle, if there are still players fighting that battle, does not end the battle itself. When all of the monsters inside a combat portal have been defeated, the portal disappears and the players are rewarded. However, if there are no players currently fighting a particular battle and the monsters were not defeated, the battle resets. The monsters regain their full health and are cured of any status ailments.

The complement of the ability to flee a battle is the ability to join a battle in progress. If there are players currently inside a combat portal, a player within sight of the portal will be able to distinguish this. In addition, that player can join the battle to assist the other players, or to possibly cause those players problems (like stealing the loot).

## Permadeath

One of the most notable features of the game is the presence of permadeath: deletion of a character upon death. The purpose of this feature is twofold. First, to add an extra layer of challenge to the game, and secondly, to prevent any one character from becoming too powerful without taking risks. Any legendary items that the character owns at the time of death are returned to the world.

## Monsters

The monsters are, at this stage, just generic programming placeholders. However, they do have rudimentary AI, as well as full combat statistics for basic PvE play. When a natural combat portal is spawned in the overworld, monsters based on that specific location are selected, and created inside that instance.

## Items

Items are randomly found throughout the world, or dropped by defeated monsters and players. Item types include consumables (food that increases HP, etc.) equipment (for boosting stats) and mundane (vendor trash, etc).

## Equipment

*Equipment items are items that can be worn by players to boost certain stats. Each piece of equipment has a certain “type” which dictates what slot it can be used in.*

# The Game Map

The game’s maps are for the most part procedurally generated. The main world map that connects most of the game world together is called the overworld. The centre of the overworld map (spawning around (0, 0)) is the root town, where players first enter the server. Safe towns, outposts, etc. like this root town are also generated elsewhere, depending on the overworld’s generation algorithm.

The overworld consists of different biomes including mountains, oceans, plains, etc. Different sections are also given difficulty ratings, from zero (for towns, safe zones, etc.) upwards. The difficulty ratings are fixed, meaning that you can be easily outgunned in a dangerous area, so it’s a good idea to keep track of the ratings as you explore.

## Dungeons

Scattered around the overworld are several types of “dungeons”. Unique monsters and items can be found inside, and they usually have one final monster that drops the biggest reward. Each dungeon has a specific difficulty rating, based partially on its location.

Each area inside a dungeon is procedurally generated, based on various factors including the dungeon’s type, difficulty rating, how deep the player is, and where in the world the dungeon is located. The locations of these dungeons are generated by the overworld algorithm. There are also unique “legendary” dungeons that are guaranteed to spawn with certain parameters.

### Ruins

The ruins are the remains of an ancient civilization. Their structures have multiple levels, with sprawling corridors and halls honeycombing through the ground.

### Towers

Towers are gauntlet-style challenges that get progressively harder the higher you go. Each level in a tower is a simple room, but it has its own monsters and challenges.

### Forests

Forests take up actual space in the overworld, but they do exist in their own instance. They’re often bigger on the inside than the outside, and often have multiple exits. The Forests are single levels, with a mazelike structure.

### Caves

Caves have several exits, often extremely far apart from each other with the cave acting as a fast transit system. However, like other dungeons, caves also have dangerous obstacles, and must be explored to reach the other end. Their structures are segmented into separate chambers, each with unique features.

## Legendary Dungeons

Legendary dungeons are special cases in the generation algorithm. These unique locations are guaranteed to spawn with certain features in every server, and are often related to the plot or the game as a whole.

# Server Management

Players can setup their own game server, while other players can connect via the internet or over a LAN. The first time a player connects to a server, they have to create a new character for that specific server.

Server moderators can whitelist or blacklist players, as well as install mods. Mods are packaged in zipped archives, and dropped into the server’s “mods” folder (simplicity here is what I’m aiming for). Actually creating mods requires knowledge of the scripting API, as well as familiarity with the lua programming language.

At first, only the world generation code can be modified, but eventually new graphics, etc. will be available. More in depth modding is possible if the server’s code is modified, rather than just the scripts.

## Player Accounts

The player accounts can be handled using Amazon servers down the line, but for now players have to input their username into the configuration file. This username is used by the server, so if two people connect using the same username, there could be problems.

## Player Characters

Each player can create a set of player characters that are connected to their account on the server. Each character has standard metadata, including name, sprite, etc. that make that character unique. In addition to this, each character also has a set of statistics (used for combat purposes and progression) and an inventory (their personal items).

If a player’s character runs out of health, and they have no way to save themselves from death, then that character is deleted; this is called permadeath. Any items that the character is holding at the time are also lost (with the possible exception of legendary items).

When a player first logs onto a certain server, they must create a new character. Characters generally cannot be transferred from one server to another, preventing players from importing overpowered characters into a world that isn’t ready for them.

# Todo List

*non-player characters? If I include a storyline further down the road, NPCs would be important.*

*factional warefare? PvP is probably the best focus for this game. Creating guilds, parties, etc. would be a good way to encourage communities on the same server. This might be a better fit for a single shard game.*

*at least a basic chat system is needed at some point*

*money & economy? again, something like this probably wouldn’t fit with a game aiming for 150 people per server.*

*mana nodes? If magic becomes a gameplay mechanic, mana nodes that increase that magic would be useful. Also, faction controlled mana nodes would be pretty cool. It seemed to work for Maridian 59.*

*world gates? Long distance teleportation. For root towns, it could be automatic, but maybe you need to have found other gate to be able to teleport there.*

*This needs some serious editing for consistency*