Tortuga Technical Document

By Kayne Ruse

Introduction 2

Languages and APIs 2

Gameplay Mechanics 2

Movement 2

Combat Portals 2

Combat 2

Permadeath 3

Player Character 3

Player Character Statistics 3

Items 4

Equipment 4

Mini Games 4

Server Mechanics 4

Server Structure 4

Data Storage 4

User Accounts 4

Scripting 4

Modding Support 4

Communication Protocols 5

Client Mechanics 5

Client Structure 5

Platforms 5

Game Controls 5

Map System 5

TODO List 5

# Introduction

This is the technical document for Tortuga. This is intended to plan out required tasks for the game, as well as serve as an instructional text for modders and others interested in Tortuga’s development cycle.

At this stage, I’m filling in the sections as I go; any empty sections aren’t a concern yet. To see some planned and expected features, refer to the design document. As with the design doc, if you see any italic text, you can consider that to be an incomplete or removed section of text or an inline comment.

# Languages and APIs

The languages of choice for creating this game are C++11 and lua, for their large user bases and wide feature sets. Third party libraries I’m using include SDL (Simple DirectMedia Layer), SDL\_net and SQLite3, for much the same reasons.

# Gameplay Mechanics

*TODO*

*Walk around the world, combat*

# Movement

*TODO*

*The players can move around the world using pixel precision.*

# Combat Portals

*TODO*

*Multiplayer*

*server control*

*map design and generation*

*exploration*

*roguelike dungeons*

*control of server mechanics and scripts*

*travel between regions (world gates)*

*RPG mechanics like items, equipment, stats, etc.*

*Player Interactions*

*TODO*

# Combat

*TODO: Move this section to the design doc*

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).

*TODO*

*Player vs monsters*

*equations*

# Permadeath

*One of Tortuga’s most influential game mechanics is permadeath i.e. the deletion of a character when the player runs out of life.*

*MORE*

# Player Character

*The player characters (PCs) will be created and customized by users. The PCs will gain levels and stat increases as the players progress with that character. When a character’s health value reaches zero, that character will die and is deleted from the server (see permadeath).*

# Player Character Statistics

*Each PC has their own unique set of statistics (stats). Possible PC stats include:*

*Health - Life Remaining*

*Mana - Magic Remaining*

*Level - Skill Level*

*Attack - Offensive Ability*

*Defence - Defensive Ability*

*Strength*

*Speed*

*Luck*

*Magic Channelling - Magic Regen?*

*TODO*

*stats can be increased by methods other than levels and equipment*

*stats increased by items and levels*

*level progression*

# Items

*TODO*

*There will be many items in the server, whether they’re consumable items, equipment or other types.*

# Equipment

*TODO*

# Mini Games

*Although a series of mini games would be a nice addition to the game, they’re unnecessary, and a waste of resources while there are other more efficient uses of development time. Despite that, it should be possible to implement mini games further down the line.*

# Server Mechanics

*TODO*

*What can a server do, and how does it do it?*

# Server Structure

*TODO*

# Data Storage

*TODO*

# User Accounts

*Each person who accesses a server must have their own user account. This allows players to keep track of their PCs, items, and other settings. This will also allow a server owner to whitelist or blacklist certain players, as well as other server specific options.*

*Each user account will have a certain number of PC slots. The items, etc. that a character collects stays with that character when a user logs out.*

*The accounts will be stored in a database.*

# Scripting

*Servers can run custom scripts on the clients, but there needs to be a limit to this.*

# Modding Support

*TODO*

# Communication Protocols

*TODO*

*TCP vs UDP?*

# Client Mechanics

*TODO*

*Available options, how to connect to a server.*

# Client Structure

*TODO*

# Platforms

*At this stage, due to a limited scope and budget, this game will only be available on PC.*

# Game Controls

*This game will have both keyboard & mouse support, as well as generic controller support.*

*TODO*

*navigate through menus, move, select, etc.*

# Map System

*TODO*

# TODO List

*Clean up this document*

*Page breaks*

*Separate the deprecated map doc*

*Add more*