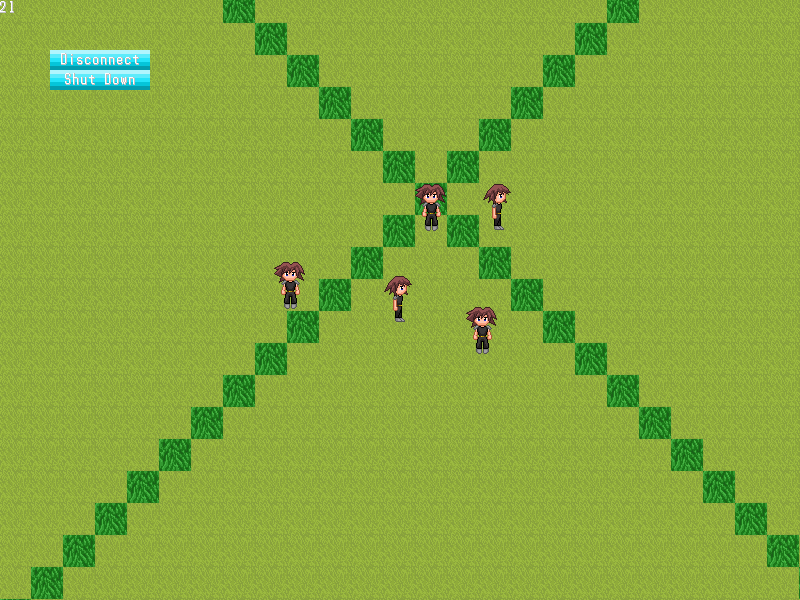
Tortuga Game Design Document

By Kayne Ruse, KR Game Studios



# Table of Contents

Table of Contents 2

Introduction 3

Elevator Pitch 3

Monetization 3

My Greenlight Checklist 4

Description 4

Required Upkeep 4

Gameplay Overview 5

The Game Map 6

The Overworld 6

Towns and Safe Zones 6

Roads 6

Dungeons 6

Ruins 6

Towers 7

Forests 7

Caves 7

Legendary Dungeons 7

Combat Mechanics (TODO) 9

Enemy Types (TODO) 10

Item Types (TODO) 11

Server Management (TODO) 12

Lua Scripting API (TODO) 13

Modding Support (TODO) 14

To Do List 15

Possibilities 15

Additions to the GDD 15

References 15

Random Thoughts 15

Devving: Creepy Ass Mother Fucker 15

# Introduction

This is the game design 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.

I have a long road ahead of me, and design docs are never really finished until the project is. So, if you see any italic text, you can consider that to be an incomplete or removed section of text, or an inline comment.

## Elevator Pitch

Tortuga is a 2D multiplayer JRPG featuring permadeath (deletion of a character upon death). The emphasis of this game is on multiplayer cooperation, exploration and customization. The game runs on customizable server software that can support up to 150 simultaneous players or more.

This game is inspired by classic 2D RPGs, as well as more modern sandbox MMOs. This project is currently independently created and funded, with the goal of creating a game that will engage user's imagination and inspire a large modding community.

## Monetization

I’m currently planning to submit this game to the Steam Greenlight process, and offering the following pricing system.

* Free Until Alpha
* $5 During Alpha
* $10 During Beta
* $15 On Release

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.

I will probably start using a modified version of the Uplink Developer’s License if and when this game becomes available on Steam.

# My Greenlight Checklist

## Description

* Early Access
* Multiplayer / Private Server MMO
* Stat Heavy JRPG
* Procedurally Generated Open World
* Dungeon Diving
* Permadeath
* Highly Moddable
* Source Code And Resources Available On Purchase

## Required Upkeep

* Change the license (probably the Uplink license).
* Keep the documentation up to date (The GDD)
* Promotional video (first impressions)
* Developer video (more in depth view of the game)
* Screenshot mock-ups (clearly marked as such)

# Gameplay Overview

The game takes place in a procedurally generated world, consisting of a single “overworld”, and a series of “dungeons”. Different areas of the overworld have different difficulty levels, which scale the types and strengths of the enemies that are found there, and subsequently the rewards for defeating them. These difficulty levels also extend to the dungeons, whose entrances are found scattered around the overworld.

Each player can create a set of player characters (PCs) that are connected to their account on the server; however they can only play as one at a time. 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). Each PC’s stats can be increased either by levelling up, or by using certain items.

Items can be found throughout the world, either dropped by enemies or hidden in chests. Item types include consumables (food that increases HP, boosts, stats, etc.) equipment (for boosting stats), legendary (unique and hard to find) and mundane (vendor trash, etc). Equipment items 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.

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).

A combat portal represents a fight in progress, located in the overworld or a dungeon. Random portals can be generated near players, sucking them and their parties in against their will. Static portals can be generated by the dungeons, as a type of boss or mini-boss encounter. Portals have visual cues, signifying their content (such as difficulty, current participants, etc.). A player can enter an existing battle, or flee one if the situation turns dire.

Enemies are procedurally generated when a combat portal is first created, based on the environment (i.e. plant-based monsters would only live in a forest) of the portal and the difficulty dictated by the location. Defeating enemies earns players rewards, but not all enemies in a portal need to be defeated to receive that reward. Landing a killing blow against an enemy will grant that player a bonus reward, while killing the last enemy on a portal will grant a significantly larger prize.

The combat system allows several people to fight side by side as a group. Each player will be able to choose their own actions, including attacking, using spells and items, or fleeing the battle. To prevent a player from acting too fast, that player must wait for their ATB gauge to fill completely before acting, after which it resets to zero.

The overall tone of the game is geared towards player cooperation in order to progress, but this is not explicitly required. Players can progress perfectly well alone, and they can even “grief” other players; some features in the future may actually support and reward these opposing styles of play.

# The Game Map

*Please note that what is listed here are hoped-for goals for the game, rather than what actually exists at the moment. I’ll rewrite this at a later date, hopefully after some of this is implemented. The schematics mentioned here refer to areas of the maps hand-crafted in an external map editor.*

*The game’s world is for the most part procedurally generated, consisting of biomes, towns, safe zones, roads, dungeon doors, etc. In addition to this, the world is broken up into difficulty regions. The difficulty at any point can be determined from an onscreen display, and is often a significant factor in what is found nearby.*

## The Overworld

*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. Dungeon entrances are also found throughout the overworld.*

## Towns and Safe Zones

*Towns and safe zones are created in the world to provide places for players to heal their characters and restock their inventories, but this may not be available at all locations. The most prominent town is called “Root Town”, where the newly created characters begin the game. This town is for the most part quite large, having all available facilities available, as well as a large town square and a teleportation device.*

*Root Town is identical in every server, since it is loaded from a schematic. Other towns are much smaller and often generated.*

## Roads

*Throughout the overworld, roads criss-cross the various biomes allowing fast and relatively safe travel between important areas like towns and legendary dungeons, etc. The further away from a town (especially Root Town) you travel, the fewer roads are available, forcing players to travel though dangerous areas.*

## Dungeons

*Scattered around the overworld are several types of “dungeons”. Uniquely themed monsters and items can be found inside, and they usually have one final “boss” 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, where in the world the dungeon is located, etc. 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, and may be entirely loaded from schematics.*

## Ruins

*The ruins are the remains of an ancient civilization. Their structures have multiple underground levels, with sprawling corridors and halls honeycombing through the ground. There are also symbols written in certain locations, hinting at an ancient language.*

*The ruins are the most Rogue-like inspired dungeons, taking inspiration from traditional dungeon generators. The separate floors of a ruin may be accomplished by teleporting between great distances in a single room object.*

## Towers

*Towers are gauntlet-style challenges that get progressively harder the higher you go. Each level in a tower is a single room, but it has its own enemies and challenges. Towers are fairly rare, and are designed as a challenge for higher-level characters. Death is entirely possible here, and exiting the tower is not possible from all floors.*

*The tower dungeons may not need a complex map generator. Instead they can simply be a series of combat instances strung together, or consist of simple round rooms filled with combat portals that get progressively harder the higher you get.*

## Forests

*Forests take up actual space in the overworld, possibly forming natural barriers, but their internals do exist in their own instance. They can be bigger on the inside than the outside, they may have multiple exits, and their internal doors don’t need to precisely correspond to their external ones.*

*The Forests are single levels, with a mazelike structure containing plant-based and poisonous enemy types. There could be a variation to forests with a swamp theme.*

## 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.*

*Caves are single room tunnel systems, and can branch apart in several locations. They may prove to be the most difficult to program, since each entrance for each cave needs to be present in the overworld’s generation algorithm, and each entrance can possibly trigger the internal generators. Forests have a similar problem, but cave doors are not connected by any overworld object, and may in fact be separated by several kilometres.*

## Legendary Dungeons

*Legendary dungeons are special cases in the overall generation algorithm. These unique locations are guaranteed to spawn with certain features in every server, and are often loaded partially or wholly from schematics. These legendary dungeons are connected to the game’s story or lore.*

*The first legendary dungeon is “Beginner’s Hall”, a ruins-type dungeon whose entrance is located at the outskirts of Root Town. This dungeon is schematic based, and provides new characters with basic equipment and items at predefined points.*

*Another possible Legendary Dungeon could be “The Ringed Fortress”, which is surrounded by impassable cliffs. To access The Ringed Fortress, you must first pass through a cave dungeon that exits inside the ring. The algorithm for creating this scenario would be particularly difficult, even if it’s partially schematic-based; nonetheless it would be a fantastic endgame- goal, and well worth the effort.*

# Combat Mechanics (TODO)

*• Level*

*• Experience*

*• Maximum Health*

*• Health*

*• Maximum Mana*

*• Mana*

*• Attack*

*• Defence*

*• Intelligence*

*• Resistance*

*• Accuracy*

*• Evasion*

*• Luck*

*Combat Mechanics*

*Tortuga’s combat consists of a unique drop-in/drop-out multiplayer active time battle system, inspired by earlier Final Fantasy instalments. Battles that a player can participate in will be visible in the game world as “combat portals”. See the next section for more information.*

*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 and items, or fleeing the battle, etc. To prevent a player from taking too many actions too fast, that player must wait for their ATB gauge to fill completely before acting, after which it resets to zero. The speed at which a character’s ATB gauge refills is affected by that character’s speed stat; each player needs 100 “ticks” in their ATB gauge to attack. They can select their specific attack from the menus while their meter is loading, but actions won’t be taken until the meter is full.*

*For a speed of 5, it would take 20 seconds to take an action, while for a speed of 50, only 2 seconds are taken, etc. These extreme time delays may seem strange, but I have a hunch that it may be necessary to allow players to search their menus. I’ll make tweaks further down the road.*

*TODO: Stats and equations*

*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. Instead, when all monsters inside a combat portal have been defeated, the portal disappears and the remaining 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).*

# Enemy Types (TODO)

*TODO*

*So I'm thinking about Boss Battles for the game. The game will have a drop in-drop out battle system, where if at least one person is still fighting the battle continues, but the people who drop out usually give something up in return (i.e. no exp, lose exp or gold, etc.) This will make permadeath easier on the players.*

*Sidenote: If a player does defeat a minion, or a part of a monster horde, they should receive at least part of the reward.*

*Anyway, for Boss battles, I want to encourage several people to take on a boss at once. one way to do this is to make the boss super strong, but I don't really want to rely on that. Another idea was to have several "minion" creatures that the boss uses.*

*i.e. You're fighting the Frog King, who summons four Frog Knights at the beginning of the battle. Every time a Frog Knight dies, a new one is summoned, so you're never fighting less than five creatures at once. The battle ends when the Frog King is dead, but the king never attacks you directly, instead summoning new Frog Knights, as well as healing and buffing the knights already on the field.*

*To defeat the Frog King, you can't just focus on him, since letting the Frog Knights attack you will almost always result in your death. So to defeat the knights and the king, the best strategy is to stun, incapacitate or draw the attention of the knights to other players, while one person attacks the king directly.*

*Now, although this would take a lot of mid level players to bring down, I could scale the number of knights being summoned in addition to the stats of the creatures for when there's only one or two players in the fight. With the drop in-drop out mechanic, I can scale the combat to match the number of players as well.*

*Anyway, this is ages away. I just wish development could go faster than it is right now. Making a video game is hard work, doubly so when you're doing it solo.*

*MOVED: Monsters*

*Monsters are script driven opponents generated inside combat portals, which also drop random rewards when they are killed. At this stage, there are no concrete plans for monsters outside of this.*

*TODO: Types of monsters, and their possible algorithms.*

# Item Types (TODO)

*TODO*

*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), legendary (unique and hard to find) and mundane (vendor trash, etc).*

*TODO: legendary items*

*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.

*TODO: Expand these few sections when I’ve reached that part of development.*

# Server Management (TODO)

*Server Mechanics*

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

*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 these 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 by altering the source code, but this would require obtaining the new programs. For security reasons, the vanilla client will not be allowed to connect to servers with modified source code, and the vanilla server will reject modified clients.*

*Data Storage*

*TODO*

*User Accounts*

*TODO*

*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.*

*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.*

*Lua Scripting*

*TODO: Information on the packaged scripts and the APIs.*

*SQL Scripting*

*TODO*

# Lua Scripting API (TODO)

*Map API*

*TODO: Not documented*

*Combat API*

*TODO: Not finished yet*

*Enemy API*

*TODO: Not finished yet*

# Modding Support (TODO)

*TODO*

* *An unmodified client must be able to connect to any server, be it vanilla, custom or modded.*
* *A server must allow the client to operate correctly.*
* *If any information causes issues, it must be resolved.*
* *If the source of the issue is the vanilla software, than that is the developer's responsibility to bring the software into line with the expected features.*
* *If however, the problem is caused by modified software or assets, than it is the responsibility of the server operator to resolve the issue.*
* *The source is available to anyone who buys the game*

# To Do List

This section is mostly just rubbish dumped here, that will eventually be moved to a previous part of the document, or outright deleted.

## Possibilities

* NPC Characters
* Factional Warfare
* Controllable Mana Nodes, captured by FW, similar to Meridian 59
* Chat system
* Money to acquire items from NPC shops
* Chaos Gates for long distance teleportation
* Gates could be automatic, but only activate if you’ve reached that gate

## Additions to the GDD

* Add the combat and statistic equations (already written, just need the section)
* Forest enemies tend to be poisonous
* Tower enemies are usually difficult, and humanoid

## References

* [Fewer Options, More Meaningful Choices](•%09http:/www.gamasutra.com/blogs/EvanJones/20130701/195361/Fewer_Options_More_Meaningful_Choices.php) - Gamasutra

## Random Thoughts

I read in an article about magic being affected by the environment. What if in the ATB system, magic was affected by what type of terrain you were fighting on? Like, if you were in a volcano, fire magic was more powerful, but ice magic was weaker? Unless you had an ice crystal to draw magic from, which makes up for the lack of ice magic sources, but the crystals only had limited amount of uses?

When a combat instance is created, a list of enemies is generated based on a series of parameters, like the biome/terrain type/dungeon type/etc. and the level (base and derived) of the area, combat instance, opponents, etc. I kinda… yeah, I’ll have fixed levelling, so your character can be outgunned by an OP boss monster, so I won’t need to adjust the difficulty based on the human characters in the fight (thank god, that would be complex has hell).

## Devving: Creepy Ass Mother Fucker

I designed one of the strongest, if not the strongest boss creature for my game today. It'll probably be the optional end goal, along the same lines as Minecraft's Ender Dragon; something mythical to strive for, but really out of reach for regular players. It also happens to be a Lovecraftian Horror, BTW.

[Depicting Cthulhu is hard](http://youtu.be/7DyRxlvM9VM), and games themselves are the worst possible medium to do it with; however, I think I have an idea. I'll get the the physical depiction in a second, but I just wanted to touch on the lore of the game.

I don't have much in the way of story line, outside of a Mutant Frog King for the first dungeon (first besides "Beginner's Keep", or even Beginner's Keep itself). To remedy this, I'm going to lay down some basic elements of the world. There may be libraries and books in the game (if I have the time), I may convey the lore via out-of-game texts (like a wiki), or I might just leave the lore ambiguous, who knows.

Anyway, the final boss is an Eldritch Horror who exists in a much higher plane of existence than the players. As a result, what the players are actually attacking, the entity they see, is little more than the Eldritch Horror's "Shadow". A creature in it's own right, but nothing more than an impression of the Horror, lurking above the world of Tortuga. Destroying the Shadow doesn't actually hurt the Horror, any more than ramming a ship into Cthulhu does.

Still, destroying the Shadow is a reward in it's own right.

Anyone who follows me probably knows that players can set up their own game servers, which itself I'm planning on working into the mythos a bit. Basically, the Horror is so large, it creates Shadows across every world on every server. It would also explain why the Shadow respawns, if it even does.

Since I tend to be way to descriptive, I'm putting the physical description under the cut.

https://secure.assets.tumblr.com/assets/scripts/vendor/tiny_mce_3_5_10/themes/advanced/img/trans.gif

OK, the physical description. Please note that the Shadow doesn't represent the Horror's "shape" in an way, any more than your shadow does when you stand under a light. I wish I could post the sketch, but you'll have to survive with text.

The top half of the Shadow's body resembles an emaciated human, with unnaturally long arms and fingers, a rib cage showing through the skin, and a bald head without a face, only a month. It's skin is mottled grey (I considered black, but it's too bland), possibly with cancerous growths (or not, I just added that idea).

The lower half starts just below it's rib cage (it lacks any kind of abdomen), where a large number of tendrils sprout. These tendrils move and flow through the air softly, much like long hair floating in water (Idea: what if each tendril represents somebody it's eaten?). The tendrils are long, giving the Shadow an overall appearance of wearing the lower half of a strange, unnatural ball gown.

Height wise, it's not large, only about 180cm from top to bottom. It floats through the air, about 15cm off the ground, but it can travel as high or low as it needs to, often moving between levels of it's habitat (I'm not sure where it lives yet) with flawless ease. It travels at roughly normal human walking speed, but it can move much faster by tilting it's upper body forward, and trailing it's tendrils behind itself. Overall, it's movements are graceful and ethereal, almost beautiful.

That's it for the main body, onto the "eyes". This, I believe, may be one of this creatures strangest attributes, and also the most Lovecraftian. Its eyes are disconnected from it's body, at least from the player's perspective. The eyes themselves can be destroyed, which only serves to piss the Shadow off.

To the average viewer, they appear crystalline, often coming in several different shapes. The first shape is of a blue, transparent octahedron, about 50-60cm wide, and a little longer from top to bottom. When designing it, I was heavily inspired by the [5th angel Ramiel](http://wiki.evageeks.org/Ramiel), sans shape shifting abilities. The second resembles several geological crystals fused together side by side, with major vertical stones clearly being "dominant" among them. The third resembles [Carbink](http://bulbapedia.bulbagarden.net/wiki/Carbink_(Pok%C3%A9mon))'s or [Diancie](http://bulbapedia.bulbagarden.net/wiki/Diancie_(Pok%C3%A9mon))'s lower halves, with a central rock-like structure and several small crystals either visible or partially visible. I think of the third form as a sort of immature eye.

The main body of the Shadow has no eyes of it's own, so to see the world, it creates these crystal eyes which can be found floating around it's habitat, and can even be found by brave adventurers who've strayed too far from safety. The Shadow can sense any and every bit of light that touches an eye; that must be a hell of a way to see the world.

The eyes themselves don't do much, but they may be able to attack if they're on their own; that's to be determined by the gameplay. When you're fighting the Shadow itself, they're probably going to act as walls.

Anyway, That's about as much as I have. I'm going to drop this post into my GDD, for posterity. I hope you liked it, and if you're interested in my game, follow or contact me!

Edit: Nearly forgot, my brother recommended making the Shadow invisible at first, until getting a silph scope style item. I think that's an OK idea, at least for pushing this into the end game. Personally, I want it to be... IDK, more attached to the lore? So I might make the player's sacrifice an eye or something in a big ritual to reveal it. What do you think?