Project Spectre Environment Design

Table of Contents

[Overview 1](#_Toc409807636)

[Task Breakdown 1](#_Toc409807637)

[Environment Representation 2](#_Toc409807638)

[Map Generation 2](#_Toc409807639)

[Generating Cells 2](#_Toc409807640)

[Generating Tiles 3](#_Toc409807641)

[Placing Players 3](#_Toc409807642)

[Placing the Boss 3](#_Toc409807643)

[Generating Mini-Bosses 3](#_Toc409807644)

[Generating Enemies 4](#_Toc409807645)

[Generating Miscellaneous Elements 4](#_Toc409807646)

[Map Manipulation 4](#_Toc409807647)

[Map Events 4](#_Toc409807648)

# Overview

This document contains technical and design details pertaining to the technical representation of the environment. It also deals with environment generation, population and manipulation. The environment in the game is defined as the game arena and all components within it that can be interacted with, excluding ‘living’ entities.

Not every aspect of this design is solid or confirmed, but all sections attempt to make logical assumptions and interpretations of information supplied so far, both by designers and by other teams.

# Task Breakdown

This section contains a list of tasks expected to be required for the environment to be developed to a point of completeness. This section is not assumed to be comprehensive, but instead aims to expose as many areas of work as possible at this stage of design.

* Map class
  + Location data structure
  + Updating the map
  + Clearing the map
* Map generation
  + Random cell selection
  + Player spawning
  + Enemy spawning
  + Item spawning
  + Tile generation
* Environment hazards
* Map events
  + Closing ring of fire

# Environment Representation

The game map will likely be composed of a data structure that represents each possible location for an in-game character to occupy. This structure will update each frame to be sure that all locations are accurate.

This structure may be used for collision detection, movement processing, and rendering.

The map class may also require functionality for generating tiles, generating cells, and placing various entities in specific locations.

# Map Generation

Map generation occurs at the beginning of each new round. Maps will be generated using a combination of static and random elements. The steps associated with generation are listed in order below:

1. Fill all placeholder cells with content
2. Generate tiles
3. Place players
4. Place boss and mini-bosses
5. Generate enemies
6. Generate miscellaneous elements

## Generating Cells

Each placeholder cell in the game world should be filled with content at the start of a round. This content should be selected from a list of options that are pre-defined by the developers.

Option selection should be pseudo-random, with each option corresponding to a function call that handles the generation of the cell.

## Generating Tiles

Tile placement should be done after all cells have been placed, so that new structures and thematic locations receive the same tiling pass as the rest of the map. Tiling should be done in such a way that map edges, divisions and quadrants are automatically identified and tiled correctly. No tiling should need to be done by hand, except perhaps the base game map.

This should be completed by looping through each tile space on the game world, identifying where that space is in relation to the surrounding tiles and the overall game world, and placing the correct tile.

In cases where multiple tiles can be placed, a random selection should be made. This random selection should be more likely to choose nondescript tiles and less likely to pick unique tiles. An example of this is a grassy field where flower tiles are occasionally placed.

## Placing Players

At the start of a round, players should be placed randomly in each of the four corners of the map. This selection is done completely randomly, starting with the player that was added to the player list first. No starting location should be more optimal than the others, and therefore it does not matter where players are placed.

## Placing the Boss

The boss should be placed at the centre of the map, and may or may not have an effect on the surrounding environment. In the event that the boss does not affect the environment, they should simply be placed in the correct location.

If the boss does affect the environment, the surrounding cell should be revised accordingly.

## Generating Mini-Bosses

Mini-bosses should be placed in a spread-out fashion across the game world. The idea is that they should be equidistant to all four players, although this may not always occur. Mini-bosses should be placed on a cell-basis, and should select randomly from a small area.

## Generating Enemies

Enemies should be generated semi-randomly throughout the game world at the start of each round. Enemies are generated at the beginning of the round and not again, except in cases of deity abilities.

Generation should be done cell-by-cell, taking into consideration what currently exists in that cell (eg: mini-bosses, the boss, hazards). Enemies should be generated in groups of varying size depending on the enemy type.

## Generating Miscellaneous Elements

Any final objects placed in the game world, whether they can be interacted with or not, should be placed on a cell-by-cell basis and should be generated based on what type of item they are.

This process could be done via a sort of generation class, or a function within each object type that determines their generation process.

# Map Manipulation

A number of deity abilities may deal with the direct manipulation of the environment. This does not include dynamic shifting of the ground, but instead deals with concepts such as generating environmental hazards, creating structures, removing miscellaneous objects and spawning enemies.

Specific details pertaining to hazards and other environmental manipulation abilities have not been supplied, but it is assumed that they will indeed exist.

# Map Events

Certain events, planned or random, can happen in the game world. An example of a planned event is the ring of fire that forces players slowly towards the centre of the game world.

These events exist independently of the game world, but can directly affect elements of the game world.