# Dungeon Delve Specification

## Title/Author

Dungeon Delve, Nicholas Wyatt

## Overview

A 2D RPG video game in which the player moves through a dungeon and maybe a town, gathering gear and supplies to gain power, while slaying monsters, conquering bosses, and evading death.

The player would progress through multiple dungeon floors, each floor with it’s own boss in order to gain treasure, items, and to level up their character in order to be able to do higher difficulty dungeon floors (the dungeon levels scale in difficulty as the player progresses). Each dungeon instance of a floor would take around 10-15 minutes to complete.

Unlike many games of this type, I will do my best to not make the game turn based unless I absolutely have to for either time or design reasons. My reasons for this is that I prefer live-action games more, and I will have more of an ability to be creative with my ideas.

## Target Audience/Customer

People who enjoy playing Video Games, specifically those who enjoy an RPG style of game. I plan to show my game along it’s different stages to members of my family, who have varying ages, and see what they think about it. This will also help with feedback on what needs to be added or changed.

## Scenarios

Scenario 1

* Jack wants to play a game for a short period of time. Jack decides to play my game and starts a new dungeon instance. Because the dungeon instances are randomized in structure, Jack has never seen this iteration before. Jack completes the dungeon, and decides he is done for the day, saves his character and closes the game. Jack’s character is now ready for another game session, and his progress is saved.

Scenario 2

* Emily wants to play my game for a long time, around 2-3 hours she thinks. She sits down and begins to play, running through many dungeon instances in the span of time she is playing. Because of the variation in treasure, monsters to fight, and dungeon layout, she is able to continue playing without becoming bored. Again, Emily is able to save her character and pick up playing another time.

## Technologies Used/Why

* Visual Studio
  + Main coding environment downloaded along with Unity. I have used this before, and it is my favorite editor to use.
* Unity
  + Used as a framework with which my game will run on. In addition, it makes the import of assets easier.

## Design/Why

Notes

* From a visual perspective, I want the game to be fairly pixelated if possible, giving an 8-bit feel. Not so much like an old arcade game but looking something along the lines of Stardew Valley or Terraria. The perspective of the game will be more like Stardew Valley than Terraria, as it is a top-down game or 3rd person game, rather than a side-scrolling one. The levels and art hopefully might look a bit like something from Children of Morta (second picture).
* Example of perspective and what I want the game to look like (taken from google, it is a picture of Stardew Valley):
* 
* 

Player Experience

* Players would be incentivized to continue playing the game through rare items being given upon a boss being killed
* Players will have to deal with multiple new challenges as they level their character up, but the new content will be metered so as to not overload the player

Game Mechanics

* Multiple dungeon floors, each with new enemies and bosses
* New Items with different stats and abilities than the ones before
* Death will function as a way to punish the player by making their dungeon instance close, losing all of their progress, but they will not lose anything past that, because having too many consequences makes the experience less fun, and more frustrating
* Multiple types of monsters, that all have unique drops and abilities
  + Monsters would drop gear, powerups, and coins
    - Coins would buy powerups and other basic gear from NPCs
      * Examples include ranged weapon ammunition, potions that give temporary boosts
* A skill tree that the player would put points into to give themselves stat boosts, and other abilities. This skill tree would be low priority compared to other important game mechanics such as coding monsters, basic items, and just making the game work. I will do my best to implement this later when I have made other stuff work.
  + Players will be able to reset the skill tree at the end or beginning of any dungeon run
  + The currency for the skill tree will be gained from multiple sources, while the points to unlock new “boxes” on the skill tree illustrated below can be gained from completing achievements
    - Skill points
      * Acquired from completing a new dungeon floor for the first time
      * Acquired from completing certain other achievements
    - Skill tree currency (name undecided as of yet)
      * Acquired from killing monsters
      * Acquired from opening reward chests upon completion of a floor
    - Cost of unlocking new levels of “boxes” on skill tree:
      * Level 1 is unlocked by default
      * Level 1 -> 2 = 2 currency
      * Gaining levels afterwards doubles the price
    - Players can only unlock the “boxes” on a higher tier on the skill tree when they have unlocked a minimum of 1 “box” on the previous tier

Goals

* Learn how to make a game with Unity and Visual Studio
* Most importantly to me is the Unity part of making my game
* Save player information to be used later, allowing for save states, and continued progress

Game Structure

* Players begin with a small tutorial level in which they learn how to use the controls, and how the game works as a whole
  + The tutorial level will be pre-built, and not randomized like the rest of the dungeon is

Skill Tree Example:

Start of Skill Tree

Skill Level: 1/50

Increase most base stats by +5% every level

Current stat boost: +5%

Skill Level: 1/50

Increase Armor stats by +5% every level

Current stat boost: +5%

Skill Level: 1/50

Increase Accessory stats by +5% every level

Current stat boost: +5%

Skill Level: 1/50

Increase Weapon stats by +5% every level

Current stat boost: +5%

Skill Level: 1/50

Gives Additional Health every Level

Health Boost: +2

Skill Level: 1/50

Increase base damage every level

Damage boost: +2

Skill Level: 1/50

Gives Additional Defense every level

Defense boost: +2

Skill Level: 1/50

Increase strength every level

Strength boost: +2

Skill Level: 1/50

Gives Additional Coin drops from monsters

Coin Increase: +5%

Skill Level: 1/50

Gives a chance at better boss drops

Drop Chance Boost: +1%

Skill Level: 1/50

Gives Additional Skill Point Currency

Currency Increase: +2%

Skill Level: 1/10

Increase most stats by +10% every level

Stat boost: +10%

Dungeon Layout Example:

Entrance Room

Start Here

Boss Room

(or portal to boss room, if the boss needs space)

1x1 Standard Room

3x1 Standard Room

Trap Room

(Filled with traps, but treasure at end)

Puzzle Room

1x1 Standard Room

Mini-boss Room

2x1 Standard Room

1x1 Standard Room

## Project Goals

* Start by creating the base of my project in Unity, and using some tutorials to understand how to make a game using it (the tutorials will just be what all the buttons do and how the system works)
* Make a game scene (whatever that means)
  + Throughout this project I will definitely be using a lot of googling to find out how stuff works and what I need to do to make things work
* Make a map that the player can run on (a very small and simple one), and display it in the Unity game viewer window
  + Make sure here that I am making it easy for myself to make more rooms later on, I don’t want to create something quick and dirty and then have to rewrite everything later when it breaks
* Make a player that can move around on the map, and make tweaks to the player and the map to suit my preferences to what the game should be like
  + Part of this step will be making sure that the player does not cross through walls and such, so I will create objects that the player should not be able to pass through and test to make sure that the player cannot
* Create a weapon that can be used by the player and test it
  + This will be the first thing that I will get a sprite animation for (I will probably use sprites because my game will be 2D, and they seem to be the easiest option), because I need to see it visually, whereas the player at this point can just be a square
  + Side note: I want to create an arrow on the ground by the player that rotates based on the way that they are facing, so that the user will be able to tell visually which way he will swing a sword or shoot an arrow. I hope that I will be able to make this independent from the movement.
* Create an enemy that can be spawned that will path-find to the player, but not attack until I have made sure that the player can hit it and do damage to it
  + Again, make sure that I make the enemy adaptable, so it is easy for me to do things with down the road.
* Up until this point, I will be spending as little time as possible on graphics, but I will begin to start making this look how they should around now
  + This includes attack animations for enemies, maybe not so much for the player as it isn’t really necessary.
  + The map will also need to look nicer, but that isn’t too important either, or at least, not as important as the enemies will be. I plan on creating terrain features later, when I get to start making the full tutorial dungeon
* I will now create the full tutorial dungeon, and populate it with monsters, run through it, and see if I can break anything
* I will now focus on making the dungeon generate randomly
* After this, I will spend my time on creating an inventory/weapon system with items being able to be moved around inside of it, and during this, I will also code up the skill tree that I talked about earlier
  + At this point, I will also create a way to save the characters, to make people able to save and load their progress
* Now that the core of the game is done, I will try to make it run in a separate game window, instead
* I will now add new items, new monsters, new rooms, and new bosses to the game
* While the game is somewhat completed now, I want to be able to distribute the game somehow, so I will begin to work on that. This is my lowest priority step
* Please let me know if you think I missed anything crucial