# Introduction

For this project the goal was to create an adventure game where the player can explore a vast randomly generated world. Within the world the player would be able to encounter NPCs to receive quests and trade, find dungeons (which would also be randomly generated) and defeat the enemies within, and loot chests they encounter on their journey. The hope was that the game would develop into more of an RPG as features were added, however this was an ambitious task given the timeframe, so the focus was placed on the adventure game aspects.

# Design

## Project Description

This game design document describes the design for a tile-based top-down adventure targeted at casual gamers, featuring several mechanics and some characters.

The game plays like most point-and-click adventure games with an emphasis on exploration and combat.

## Characters

There will be a few characters present in the game. The first of which is the player character. The player character can traverse the world and explore, engage in combat with enemies, and collect loot to sell/use.

There will also be bandits/monsters that can be found in dungeons or out in the world who will attack the player character and reward the player with experience points (XP) or currency upon their defeat in combat.

Additionally, there will be a trader character and other NPCs who will be non-hostile to the player and allow for additional interactions such as quest-giving and trading.

## Story

There will be very little story elements as the game primarily focuses on exploration and combat, however there will be some radiant/fetch quests that the player can receive to further incentivise the exploration of more hostile areas.

## Gameplay

### Goals

Overall, the player’s goal is to complete quests and fully explore the world. Short-term goals will typically be defeating enemies, clearing dungeons, and searching for loot.

### User Skills

The player will be expected to:

* click on the screen to interact with the world, move around, engage in combat etc.
* manage resources such as health, magic, stamina, etc.
* strategize to defeat enemies.

### Game Mechanics

* Procedurally generated dungeons will allow the player to explore a more challenging area than the outside world. The use of randomly generated dungeons allows for each dungeon to be unique with minimal time spent designing each one individually and will also increase the replay value of the game. The dungeons will contain several enemies of varying difficulty top provide a challenge to the player and loot can be found throughout to encourage and reward the player.
* A turn-based combat system will allow the player to defeat the enemies they encounter (i.e. in dungeons etc.). The combat itself will involve the player choosing from a few options to determine their next move (such as attack, heal, etc.) after which the enemy will react accordingly. The player will also be responsible for managing their resources throughout as each action will consume either some magic, stamina, or an item from their inventory, while incoming attacks from the enemy will decrement their health.
* An inventory system will allow the player to manage the items they loot/purchase from the trader. This will include equipping any weapons/armour they own, consuming any potions etc., and managing their owned currency. A carrying capacity will also be present to encourage players to prioritise only the most valuable/useful items when looting.
* A trader will be present to allow the player to sell unwanted gear or valuables they have acquired in exchange for some currency that they can use to purchase more useful items. The trader will be located at the town present in the world which will act as a safe space for the player.
* Other NPCs will be present in the town and can offer the player quests to award the player additional rewards for exploring/clearing dungeons. This will be accomplished through a simplistic dialogue system.

### Items

The player will be able to acquire additional weapons and armour through looting/trading and will be able to equip these in their inventory for a boost to their damage output/defence. Potions will also be gained through looting/trading which can be consumed to restore some of the players core stats (health, magic, and stamina). \*Quest items may also be found when looting which can be returned to the quest giver for a reward.

### Progression & Challenge

As the player completes quests/clears dungeons etc. they will gain XP and level up which will result in enemies becoming tougher to defeat and greater in numbers to pose an increased threat to the player. To mitigate against this, the player will have to improve their gear and tactics and remain well-stocked in supplies when engaging in combat. \*As they level up, more dungeons/areas could become available to increase the challenge to the player further while offering an improved reward.

### Losing

If the player loses all their health, they will be revived at the town which acts as a safe space and will perhaps lose their progress towards the next level.

## Art Style

This is a top-down 3D tile-based game with many 3D assets. They will be in a low poly style to reduce the artistic workload and will be following the colour scheme of many other medieval/fantasy games.

\*\*Assets may be sourced rather than created where necessary

## Music & Sounds

The music will have a fantasy style like many older RPGs. Sound effects will be used to praise and encourage the player when they are succeeding (e.g. defeating an enemy) and alert/warn the player when they are struggling (e.g. low on health).

\*\*Assets may be sourced rather than created where necessary

## Technical Description

The game will be created using Unity 2019.2.5 using an existing codebase containing an initial tile-based world generator and player movement/pathfinding. It will be targeted for PC using a mouse for most controls.

GitHub will be used for source control and the project will be pushed after the completion of each individual mechanic with an aim to keep the main version of the project uncorrupted with broken or unfinished code.

All code will be thoroughly commented to ensure it is readable and can be updated easily where applicable.

## Other Ideas

* Build Mechanic
* Player housing/home storage
* Collectibles
* Achievements
* Main story/more NPCs
* More enemy types/dungeon variation

# 3rd Party Assets/Tools

The font used throughout the game can be found at <https://www.dafont.com/enchanted-land.font>.

All additional art assets present in the project were made myself using either Paint.net, FireAlpaca, or Blender. (Excluding the assets provided in the starter project).

# Evaluation

Overall, the final product meets most of the criteria I initially set out to achieve, however some features had to be abandoned due to time constraints so a better balance in time dedicated to each feature and a better anticipation of the problems that were encountered would have improved the outcome.

. The initial design was somewhat ambitious to begin with so perhaps a more conservative design could have resulted in a more polished end product. However, the project in its current state serves as a good foundation from which a solid game could be created through further development.

The majority or problems encountered throughout development involved the world generation, particularly with the dungeons/structures. Problems such as the interior for the dungeons not spawning the correct models the first time the player entered due to the corridor generation occurring after the tiles were instantiated and a temporary solution was devised to reload the map a second time upon entering a dungeon. While this solution worked as intended and solved the problem, it’s not ideal as larger sized maps could begin to cause some more serious performance issues, however due to the current size of the maps used, it appears to sufficient and so development time was spent elsewhere on Instead.

The inventory and trading system caused several problems primarily caused when attempting to handle button clicks through code. This was necessary in order to allow the button click to trigger a function call that needed data not available in the editor, and so a non-persistent event was needed. The benefit to handling the button click events this way was that if more inventory slots were added, it’d be scalable, however unfortunately due to time constraints a solution had to be devised involving manually entering the data in the editor meaning it is no longer scalable but works mostly as intended. The result of the inventory system works for using/dropping items as well as transferring items to/from containers. The only remaining error with this involves transferring stacked items and given more time to debug would have been solved.

Despite these problems, most features such as the world generation and combat system work as intended and with a little more polish, this project could be made into a solid game.