Software Requirement Specification

For

Dungeon Run

Version 1.0.0

Prepared By MF-JAP
Adam, Fredrik, Jamel, Md, Palash

Date Created: 2019-12-18

Table Of Contents:

1. Introduction

1.1 Purpose

This is the project work under the course Agile and Project methodologies. Group members implemented a game called Dungeon Run using and following the agile project methodologies.

1.2 Intended Audience

As this project is done under a project work in a course, the main purpose is to complete the game in workable states for the presentation and the course instructor.

1.3 Intended Use

Though is is done for a course project work group members try to make a total playable game. So that the game can be develop and update later on also.

1.4 Project Scope

This project is done under three weeks time and develop programming and documentation according to agile project methodologies. The project version is 1.0.0. The project can be implemented with graphical interfaces. This is the biggest scope project can be upgraded later versions.

2. Overall Description

2.1 User Needs

This is a general dungeon run game. Every can play this game. This include help documents which can help to direct how to play.

2.2 Assumptions And Dependencies

This project comes with .exe file, so, anyone can just use .exe file to play the game. This also has have .py. Readme.md file.

3. System Features And Requirements

3.1 System Features and Requirements

It is a dungeon game being written in Python 3.7 with curses module. It is in a very early version, but the main features will be:

- Mostly first person experience, along with top-down view on the map.

- Dungeons, treasures, chests, monsters, skeletons, etc.

- Basic kind of map/locations

- Some kind of AI

- single multiplayer

- character generation

- Everything else that is awesome

Progam features: Classes and modules are describe below

Dungeon.py has the following functions---

```
def game_meny(inpt):  starting manu for the game

def create_map(map_Size):  building maps and map sizes

def accepted_Hero_Input(question): hero choose by user

def place_Monster(grid_map):  place monster on the map

def spawn_Rate():  will spawn random monsters

def create_Monster(monster_Type):  Argument monster type, as a string. Will return a monster object.

def hero_Attack(hero_Object, monster_Object):  Argument is Hero object and Monster object. Will
return 0, 1 or 2 if the Theif creats.

def monster_Attack(monster_Object, hero_Object):  Argument is Monster object and Hero object. Will
return 1 or 0.

def start_Battle(start_Type, hero_Object, monster_Object): Start battle when meets monsters.

def flee(flexibility): Argument is Heros Flexibility, Will return 1 if you fleed else 0.
```

Creating hero_class.py which includes:

```
class Hero: Superclass for heroes, arguments are initiative, immunity, attack, flexibility. This
class got the flee func.

def __init__(self,initiative, immunity, attack, flexibility): defining the heroes behavior.

And three heroes classes are:

class Knight(Hero):
class Wizard(Hero):
class Thief(Hero):
```

monster_superclass.py

```
class Monster: creat a common monster class with different behavior.

def __init__(self,initiative,immunity,attack,flexibility,rate,name):

and different monsters classes are:

classSpider(Monster):

classSkeleton(Monster):

classOrc(Monster):

class Troll(Monster):
```

Treasers.py includes the following class and functions:

```
class Treasure:

class Coin(Treasure):

class Money_Pouch(Treasure):

class Gold(Treasure):

class Gemstone(Treasure):

class Small_Chest(Treasure):

and the treasure drop rate calculate with the function:
def drop_rate():
```