FF project Implementation

## --Version 0.1

## Program elements :

**modules needed :**

random

time

### Class Character:

**constants :**

RANDOM\_MALE\_NAMES : {‘Albert’, ‘Kronos’, ‘Alec’, ‘Bran’, ‘Urtuk’, ‘Shun’}

RANDOM\_FEMALE\_NAMES : {‘Lydia’, ‘Tulie’, ‘Aimee’, ‘Selene’, ‘Kira’, ‘Ksyncix’}

RANDOM\_OTHER\_NAMES :{‘Al’, ‘Effen’, ‘Juno’, ‘Lerrewen’, ‘Gorgo’, ‘Jviz’}

GENDERS : {‘Male’, ‘Female’, ‘Other’}

RACES : {‘Githzerai’, ‘Rakshasa’, ‘Illithid’, ‘Tieflin’, ‘Banshee’}

ABILITIES : {‘Githzerai’ : (name ‘Vicious Swash’, damage min -10, damage max 10), ‘Rakshasa’ : (’Subjugate’, -10, 10), ‘Illithid’ : (’Mind pump’, -10 ,10), ‘Tiefflin’ : (‘Sting whip’, -10, 10), ‘Banshee’ : (‘Scream’, -10, 10)}

RACE\_LIFE\_POINTS : {‘Githzerai’ : 30, ‘Rakshasa’ : 38, ‘Illithid’ : 25, ‘Tieflin’ : 35, ‘Banshee’ : 28}

RACE\_INTELLIGENCE\_POINTS : {‘Githzerai’ : 15, ‘Rakshasa’ : 10, ‘Illithid’ : 25, ‘Tieflin’ : 8, ‘Banshee’ : 20}

RACE\_STRENGTH\_POINTS : {‘Githzerai’ : 12, ‘Rakshasa’ : 20, ‘Illithid’ : 10, ‘Tieflin’ : 18, ‘Banshee’ : 8}

**attributes :**

name : a random selected name from a set (male/female/other sets)

gender : a random selected male/female/other

fantasy race : a random choice from set (Githzerai, Rakshasa, Banshee, Tiefflin, Illithid)

special ability : for each race, a special power that deals damage

strength : int = race strength points + random range of 5 to 10

intelligence : = race intelligence points + random range of 5 to 10

level = 1 by default (at the beginning of the program)

life : = race life points + random range of 10 to 20

armour : a random choice from a set

spell : a random choice from a set

weapon : a random choice from a set, including fist

**methods :**

\_\_init\_\_

\_\_repr\_\_ : print name, gender, race

hit :

* Use weapon attack or spell or special ability to make damage or fail.
* If no weapon and no spell, use default fist. Fail means 0 or a negative damage.
* Damage is dealt with weapon power or spell power, strength (if weapon), intelligence (if spell).
* Damage is reduced by armour points.
* This function modifies in place the enemy.life with the damage dealt
* This function prints a desc\_hit + desc\_damage or desc\_heal

heal : if special healing ability doesn’t fail, gives the player back some random points of life (range ?)

desc\_hit : prints a description of each blow : “{player.name} uses {weapon/spell/special\_ability} !”

desc\_damage : prints a description of received damage : “Miss !” or “{weapon/spell/special\_ability} deals {damage} damage on {player.name or character.name}”

desc\_heal : prints a description of received healing : “{player.name} restores {heal} points of life !”

### Class Player(Character) :

**constants :**

XP\_LEVELS = {‘2’ : 500, ‘3’ : 1000, ‘4’ : 2000 , ‘5’ : 3500, ‘6’ : 5000, ‘7’ : 7000, ‘8’ : 10 000, ‘9’ : 15000, ‘10’ : 22000}

XP\_GAINS = {‘1’ : 125, ‘2’ : 175, ‘3’ : 200 , ‘4’ : 275, ‘5’ : 375, ‘6’ : 400, ‘7’ : 475, ‘8’ : 500, ‘9’ : 750}

**attributes :**

name : an user input

gender : an user choice from male/female/other

fantasy race : an user choice (Githzerai, Rakshasa, Banshee, Tiefflin, Illithid)

gold : int, randomly generated (10 to 200)

weapon : the equiped weapon from player inventory (default : ‘Fist’ damage min 0 damage max 3)

armour : the equiped armour from player inventory (default : ‘Underwear’)

spell : the equiped spell from player inventory (default : None)

inventory : dict containing all weapons, spells and armours in possession of character

experience = 0 by default (at the beginning of the program)

wins = 0 by default

**methods :**

\_\_init\_\_ : same as character + other or modified attributes listed upper

\_\_str\_\_ : display player name, race, gender, strength, life, intelligence

gain\_xp : player gains experience, depending of enemy’s level

loot : loot random gold (0-100)

level\_up : increase player’s stats (including special abilities) by 1 for 1 to 5 level, by 2 for 5 to 10

print\_achievements : print last level reached and number of enemies defeated

equip(item) :

* If another object is already equipped in related slot, put it back in inventory
* add item to player’s weapon or spell or armour

hit(enemy) :

* Choose weapon attack or spell or special ability to make damage or fail (verify input)
* then same as Character.hit()

display\_inventory : display weapons, spells, armours, gold.

### Class Shop :

**attributes :**

stock\_armour : dict with name and properties of armours

{‘Corset’:(price 100, prot 5), ‘Rags’ : (10, 1), ‘Leathersuit’ :(250, 10), ‘Platemail’ : (700, 25), ‘Mithril jacket’ :(1500, 40)}

stock\_spells : dict with name and properties of spells

{‘Blizzard’ : (price 200, damage min 8, damage max 16), ‘Scorch’ : (100, 4, 10), ‘Venom gaze’ : (150, 15, 25), ‘Wasp stings’ :(50, 1, 4), ‘Lightning’ : (400, 25, 35)}

stock\_weapons : dict with name and properties of weapons

{‘Scythe’ : (price 400, damage min 25, damage max 35), ‘Scissors’ :(50, 1, 4), ‘Halbert’ : (200, 8, 16), ‘Club’ : (100, 4, 10), ‘Dagger’ : (150, 15, 25)}

**methods :**

\_\_init\_\_

\_\_str\_\_ : display the selected stock, (objects stats and prices)

buy : checks if the player has enough gold and puts item in their inventory

sell: buys an equipment the player chooses in their inventory for half its value and removes it from player’s inventory

### Class Armour() :

**attributes :**

name : a string

price : int

protection : int

### Class Weapon():

attributes :

name : a string

price : int

damage min : int

damage max : int

### Class Spell (Spell):

**attributes :**

name : a string

price : int

damage min : int

damage max : int

## Functions:

autogen() : generates random stats for player opponents. Must also generate a random armour, spell or weapon of low level.

fight(player, \*enemies) :

* as long as all the enemies and the player are still alive, alternate player.hit() and enemy.hit()
* There may be multiple enemies
* choose the enemy to hit if needs be (verify the input)
* hit(chosen enemy) or hit(player)
* the fight is lost when the player dies before all enemies, print “You loose”, wait 5 secs and call game\_over function
* otherwise fight is won, print “You win”

game\_over() : print “Game over\n Thank you for playing\n Credits : conception : Aphios”, wait 15 secs then close the program

verify\_inputs(input, available\_terms) : to be used in equip method and character creation and questions asked to player in game. Verifies that the input corresponds to one of the available terms (list of strings). If yes it returns the input.

## Main :

create user character

create random opponent matching player level

if player is level 3 or more, he may have two opponents of 1 level lesser, if player is level 6 or more, he may have three opponents of 2 levels lesser (see if needs adjustments)

go to shop to buy some equipment

equip character (player chooses which equipment if he has more than one in inventory) and work out combat stats

fight (loop)

get xp and level up if need be or die and get game over message

if player wishes to continue :

eventually go to shop again and buy/sell until the players says to quit shop

generate new adversaries matching player’s level

fight again

otherwise endgame