

Université de Bordeaux

L1 MIASHS



---

## Projet ASD Donjon & Dragon

---

Projet bâti par : Brieux CAQUELIN - Julien LETANG

Année universitaire 2016-2017



## Sommaire

1	Structures de nos données . . . . .	3
2	Principaux algorithmes . . . . .	5
2.1	Fonction attaque perso . . . . .	5
2.2	Fonction attaque . . . . .	6
2.3	Fonction soin . . . . .	6
2.4	Fonction gestion équipe . . . . .	6
2.5	Action loop . . . . .	7
2.6	Mode Aventure . . . . .	7
	Loot . . . . .	7
	Scenario . . . . .	7
3	Bilan . . . . .	8
3.1	Ce qui fonctionne : . . . . .	8
3.2	Ce qui ne marche pas : . . . . .	8
4	Critique du travail réalisés . . . . .	9
5	Sources . . . . .	10

<b>Références</b>	<b>10</b>
-------------------	-----------

## 1 Structures de nos données

Nous avons utilisé des class pour regrouper toutes les statistiques des personnages. Cette méthode nous a permis de les appeler plus vite lors de leur utilisation dans d'autres fonctions (*fonction attaque, soin et initiative*).

Ci-dessous nous avons la "class Personnage(object) :" qui nous permet d'extraire les informations contenues dans les fichiers .csv, des différents personnages ou ennemis.

```
class Personnage(object): #création de la classe personnage et attribution
                           des valeurs
    #begin
    def __init__(self,nom,famille,force, point_de_vie,
                  classe_d_armure,bonus_initiative,BBA,
                  type_arme,nombre_des,type_des,zone_critique,facteur_critique,
                  bonus_magique,nb_soin_1,nb_soin_2, nb_soin_3,initiative):
        self.nom=nom
        self.famille = famille
        self.force=force
        self.point_de_vie=point_de_vie
        self.classe_d_armure=classe_d_armure
        self.bonus_initiative=bonus_initiative
        self.BBA=BBA
        self.type_arme=type_arme
        self.nombre_des=nombre_des
        self.type_des=type_des
        self.zone_critique=zone_critique
        self.facteur_critique=facteur_critique
        self.bonus_magique=bonus_magique
        self.nb_soin_1=nb_soin_1
        self.nb_soin_2=nb_soin_2
        self.nb_soin_3=nb_soin_3
        self.initiative=initiative

        #calcul des valeurs
        #bonus d'attaque lié au perso
        self.bonus_attaque = mod(self.force)+self.BBA+self.bonus_magique
        #bonus de dégât lié au perso
        self.bonus_degat = modificateur_arme(self.type_arme) * mod(self.force)
                           + self.bonus_magique
```

Nous utilisons également des list. Les listes nous sont utiles pour employer à nouveau les personnages issus des données class un à un dans les équipes. Chaque début de liste est composée par un élément 0 qui définit si l'équipe est jouée par l'ordinateur ou le joueur.

## 1 STRUCTURES DE NOS DONNÉES

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Nom	Famille	Force	Points de vie	Classe d'armure	Bonus initiative	Base attaque	Type arme	Nombre des	Type des	Zone critique	Facteur critique	Bonus magique	Nb soin 1	Nb soin 2	Nb soin 3	Initiative
2	Gromel	Le Nain	18	16	12	2	1	2	1	12	20	3	0	1	0	0	0
3	String	Le Voleur	11	6	14	3	4	1	1	4	19	2	0	1	0	0	0
4	Tytharas	Le Druide	8	16	12	1	2	1	1	4	19	2	2	1	0	0	0
5	Orvis	Le Prestre	15	13	17	0	0	1	2	8	20	2	1	1	0	0	0
6	Rani	L'Elfe	10	9	14	3	0	2	1	6	18	2	0	0	1	3	0
7	Yrien	La Nymphe	11	15	15	5	2	1	1	4	19	2	0	2	0	0	0
8	Judge	Le Paladin	16	28	19	1	3	1	1	8	19	3	2	1	0	0	0
9	Grubba	Le Démon	17	21	17	3	1	1	1	8	20	2	0	1	1	0	0
10	Hey	Le Barde	18	84	22	5	7	1	1	8	20	1	1	0	0	0	0
11	Hodor	Le Géant	15	45	17	-2	3	1	1	8	20	2	0	1	0	0	0
12	Bolt	L'Humain	12	13	14	2	2	2	2	4	19	2	2	1	2	0	0
13	Waterblue	L'Esprit	17	21	14	2	1	2	2	6	20	2	2	1	2	1	0
14	Gaia	La Déesse	16	13	13	4	2	1	1	10	19	3	1	0	0	2	0
15	Dararen	Le Demi-Dragon	14	25	19	1	1	1	1	12	20	2	1	1	0	0	0
16	Leeroy	Le Gnome	25	3	19	7	4	2	2	12	18	3	2	0	0	0	0
17	LanceGoo	L'Orc	18	17	9	1	1	2	1	14	19	2	2	0	1	0	0
18	Aythea	La Chasseuse	12	29	19	0	1	2	2	4	18	2	1	1	0	0	0

FIGURE 1 – perso.csv

Enfin nous avons utilisé des fichiers .csv, pour créer et gérer les personnages, les ennemis, ainsi que les salles des donjons. Toutes les informations issues de ces fichiers peuvent être appelées dans les différents algorithmes.

## 2 Principaux algorithmes

### 2.1 Fonction attaque perso

```
def attaque_perso(att,cible,type_att):
    """ Définition de la fonction attaque perso qui gère la différence
    entre attaque d'opportunité et attaque normale. """
    assert isinstance(att, object), "Type att incorrect, il faut un object"
    assert isinstance(cible, object), "Type cible incorrect, il faut un object"
    assert isinstance(type_att, bool), "Type type_att incorrect, il faut un bool"

    #var
    nbr_attaque = 0 #int

    #begin
    if type_att == True :
        #begin
        #gère le nombre d'attaques en fonction de la BBA
        att.nbr_attaque = nombre_attaque(att.BBA)
        while att.nbr_attaque != 0:
            att.attaque(cible)
            att.nbr_attaque -= 1
        else :
            #vérifie si l'équipe ennemie n'est pas morte
            if type(att) == bool:
                pass
            else:
                #dans le cas d'une attaque d'opportunité il n'y a qu'un seul coup porté
                att.attaque(cible)
            #end

    print("Les points de vie de",cible.nom,"sont à :",cible.point_de_vie)
    # Gère la mort d'un joueur et le retire de son équipe
    if cible.point_de_vie <= 0:
        equipe = verif_equipe(cible,equipe1,equipe2)
        equipe.remove(cible)
        print("")
        print(" † R.I.P.",cible.nom,cible.famille,"est mort : il est retiré de
            l'équipe")
    #end
```

Cette fonction permet de gérer la différence entre l'attaque normale et l'attaque d'opportunité.

Elle affiche aussi le nom, et la famille du personnage lorsqu'il est mort, en indiquant aussi qu'il est retiré de l'équipe.

## 2.2 Fonction attaque

La fonction gère l'action attaque pour le personnage qui la choisit. Elle vérifie aussi que l'adversaire peut encore être attaqué. Elle permet de le sélectionner et de lancer l'attaque en fonction du jet, pour enfin infliger les dégâts à l'adversaire sélectionné.

## 2.3 Fonction soin

Une fois activée elle vérifie le nombre de potions du personnage et lui propose laquelle il veut boire. Elle permet aussi de gérer l'attaque d'opportunité que le personnage peut recevoir pendant l'utilisation de la fonction soin, de manière aléatoire. Elle vérifie également l'état des autres personnages.

## 2.4 Fonction gestion équipe

La fonction suivante permet de créer les équipes, de manière aléatoire pour le cas d'une équipe contrôlée par l'ordinateur, et manuellement si l'équipe est composée par le joueur.

Elle affiche aussi le numéro permettant de sélectionner les personnages, et rafraîchit la liste de numéros après le choix de chaque personnage en le retirant des choix possibles.

Elle liste les personnages quelle récupère dans la classe, et affiche à l'écran le nom de chaque personnage avec sa famille, son nombre de PV et sa force.

```

*** CHOIX DES HÉROS ***

EQUIPE 1

Tapez 0 pour Gromel ----- ' Le Nain ' ----- 16 PV 18 Force
Tapez 1 pour Stings ----- ' Le Voleur ' ----- 6 PV 11 Force
Tapez 2 pour Tytharas ----- ' Le Druide ' ----- 16 PV 8 Force
Tapez 3 pour Orvis ----- ' Le Pretre ' ----- 13 PV 15 Force
Tapez 4 pour Rani ----- ' L'Elfe ' ----- 9 PV 10 Force
Tapez 5 pour Yrlen ----- ' La Nymphe ' ----- 15 PV 11 Force
Tapez 6 pour Judge ----- ' Le Paladin ' ----- 28 PV 16 Force
Tapez 7 pour Grubba ----- ' Le Demon ' ----- 21 PV 17 Force
Tapez 8 pour Hev ----- ' Le Barde ' ----- 84 PV 18 Force
Tapez 9 pour Hodor ----- ' Le Geant ' ----- 45 PV 15 Force
Tapez 10 pour Bolt ----- ' L'Humain ' ----- 13 PV 12 Force
Tapez 11 pour Waterblue ----- ' L'Esprit ' ----- 21 PV 17 Force
Tapez 12 pour Gaja ----- ' La Deesse ' ----- 13 PV 16 Force
Tapez 13 pour Dareren ----- ' Le Dragon ' ----- 25 PV 14 Force
Tapez 14 pour Leeroy ----- ' Le Gnome ' ----- 3 PV 25 Force
Tapez 15 pour LancaGood ----- ' L'Orc ' ----- 17 PV 18 Force
Tapez 16 pour Ayshea ----- ' La Chasseuse ' ----- 29 PV 12 Force

Choix du perso de l'équipe 1 : 1
Stings a été recruté dans l'équipe 1.

EQUIPE 2

```

Fonction gestion équipe en fonctionnement

## 2.5 Action loop

Cette fonction est une fonction essentielle au bon déroulement de la partie, que se soit en mode solo ou duo. En effet elle gère l'action complète d'un personnage, dès le moment où son tour commence jusqu'à la fin du tour.

Elle est effective tout au long de la partie et avec tous les personnages présents dans le tour.

## 2.6 Mode Aventure

Le mode aventure est le nom donné à l'algorithme qui gère une partie entre le joueur et l'ordinateur. Ce mode est scénarisé!

Il contient deux principaux algorithmes.

**Loot :** Le loot permet de gérer les loots de chaque joueur.

Il fonctionne avec un système de points qui se cumulent au fur et à mesure que les aventuriers passent de salle en salle.

**Scénario :** La fonction scénario, gère les salles ainsi que les différentes équipes de monstres que l'on rencontre dans chaque salle.

Elle gère aussi la rencontre avec le Boss du Donjon après la 7<sup>e</sup> salle.



## **3 Bilan**

### **3.1 Ce qui fonctionne :**

L'ensemble du projet fonctionne parfaitement hormis quelques dysfonctionnements que nous n'avons pas vus.

### **3.2 Ce qui ne marche pas :**

L'emploi des touches alphabétiques bloque par endroits le programme, donc il est encore vivement conseillé d'utiliser uniquement le pavé numérique.

## 4 Critique du travail réalisés

Nous avons mis un peu de temps avant de réussir à avancer dans le projet, car les class nous ont posées quelques soucis dans leurs compréhensions.

Mais ceci n'a pas été un frein à la poursuite du projet, car nous pensions déjà aux idées que nous devions réaliser une fois les difficultés surmontées.

Dès le début nous avons pensé à faire un mode aventure avec des donjons et des ennemis à l'intérieur.

Même si pour la remise du projet d'ASD, nous vous avons pas fait parvenir un mode Aventure avec des donjons et plusieurs difficultés et avantages. Nous voulions quand même vous en faire parvenir un avec un combat simple entre le joueur et l'ordinateur.

Rassurez-vous, nous vous envoyons bientôt notre mode aventure finalisé.

## 5 Sources

- Pour comprendre et utiliser les “class”, nous nous sommes aidé des tutos du site Sam et Max [1].
- Pour élaborer le classement nous avons utilisés OpenClassrooms, voir [2].

## Références

- [1] [Sam et Max](#). Cours et tutos, 2017. [© 2017 Sam & Max].
- [2] [OpenClassrooms](#). Apprenez à programmer en python, 2016. [En ligne ; Page mise à jour le lundi 12 décembre 2016].