

■ SHADOW DUNGEON ■

Projet Personnel Python — POO

Inspiré de l'univers Solo Leveling • Vacances d'Avril 2025 • 1–2 semaines

■ DESCRIPTION DU PROJET

Shadow Dungeon est un jeu de rôle en console/terminal entièrement développé en Python (POO). Le joueur incarne un **chasseur** qui pénètre dans des donjons apparus sur Terre, affronte des monstres, collecte du butin, monte en rang et gère ses compétences et son équipement. Le jeu respecte fidèlement les mécaniques de l'univers Solo Leveling : système de rang, gates de donjon, boss de fin d'étage, et progression du chasseur.

■ UNIVERS & LORE (fidèle à Solo Leveling)

Les Gates (Portes de Donjon)

Des failles dimensionnelles — appelées **Gates** — apparaissent aléatoirement sur Terre. Chaque gate possède un rang (E à S) qui détermine la difficulté des monstres à l'intérieur et le niveau minimum recommandé pour y entrer. Un chasseur ne peut pas entrer seul dans une gate de rang trop supérieur au sien sans risquer l'élimination immédiate. Les gates ont une durée de vie limitée : si personne ne les ferme, elles explosent et les monstres envahissent le monde réel (événement **Dungeon Break**).

Les Chasseurs

Les chasseurs sont des humains dotés de capacités magiques suite à l'apparition des gates. Chaque chasseur est classé de **E** (le plus faible) à **S** (le plus puissant). Ils peuvent augmenter leurs statistiques en complétant des donjons, mais leur rang officiel ne peut être réévalué que lors d'examens spéciaux organisés par l'Association des Chasseurs. Le joueur commence en tant que chasseur de rang E et progresse au fil des donjons.

Tableau des rangs de chasseurs :

Rang	Niveau requis	Stat totales min.	Gates accessibles	Couleur
E	1 – 20	~ 50	E uniquement	Gris
D	21 – 40	~ 150	E, D	Vert
C	41 – 60	~ 350	E, D, C	Bleu
B	61 – 80	~ 700	jusqu'à B	Violet
A	81 – 99	~ 1 500	jusqu'à A	Or
S	100+	~ 5 000+	toutes	Rouge

■■ FONCTIONNALITÉS DU JEU

1. Menu Principal & Profil du Chasseur

- Création du chasseur : nom, choix de classe (voir §Classes)
- Affichage du profil complet : stats, rang, niveau, XP, inventaire, compétences actives
- Sauvegarde / chargement de partie (fichier JSON)
- Accès à la Boutique de l'Association : acheter/vendre équipements

2. Système de Donjon

- Sélection d'une gate disponible parmi plusieurs générées aléatoirement (rang E à S)
- Chaque donjon comporte entre 3 et 10 étages selon son rang
- Chaque étage contient un groupe de monstres à éliminer avant de passer à l'étage suivant
- Le dernier étage contient un **boss** unique avec des capacités spéciales
- En cas d'échec (HP = 0), le chasseur est éjecté du donjon et perd de l'XP
- Fermer la gate rapporte de l'XP, du butin, et des cristaux magiques

3. Système de Combat (tour par tour)

Le combat se déroule en **tours alternés** entre le chasseur et les monstres. À chaque tour, le joueur choisit parmi les actions suivantes :

- Attaque normale : inflige des dégâts basés sur la Force et l'arme équipée
- Utiliser une compétence : consomme du Mana, effets variés (dégâts, buff, debuff, soin)
- Utiliser un consommable : potions de soin, potions de mana depuis l'inventaire
- Fuir le combat : tentative de fuite (chance basée sur l'Agilité) ; interdit contre les boss

Les monstres ont leurs propres comportements : certains attaquent au corps-à-corps, d'autres à distance, certains se soignent ou appellent des renforts selon leur type.

4. Statistiques du Chasseur

Stat	Abrév.	Rôle
Points de Vie	HP	Survie en combat — si 0 : défaite
Points de Mana	MP	Ressource pour les compétences
Force	STR	Dégâts des attaques physiques
Agilité	AGI	Ordre d'action, esquive, fuite
Intelligence	INT	Puissance des compétences magiques
Perception	PER	Détection des pièges, trésors cachés
Endurance	END	Réduction des dégâts reçus
Sens	SEN	Coup critique, taux de réussite des skills

- À chaque montée de niveau, le chasseur gagne des points de stat à distribuer librement, comme dans le système de Solo Leveling.

5. Classes de Chasseurs

À la création du personnage, le joueur choisit une classe qui oriente ses statistiques de départ et ses compétences disponibles. Les classes disponibles sont :

Classe	Stat principale	Style de jeu	Compétence signature
--------	-----------------	--------------	----------------------

Guerrier	STR / END	Tank / DPS physique	Frappe du Conquérant
Assassin	AGI / SEN	Burst / esquive	Lame de l'Ombre
Mage	INT / MP	DPS magique / contrôle	Nova Arcanique
Guérisseur	END / SEN	Support / survie	Lumière Sacrée
Archer	AGI / PER	Distance / critique	Flèche Percante

6. Monstres & Boss

Les monstres sont générés en fonction du rang de la gate. Chaque monstre possède ses propres statistiques (HP, ATK, DEF, vitesse) et un type d'attaque. Les boss de fin de donjon sont uniques : ils ont des noms, une description, des phases de combat (le boss change de comportement sous 50% et 20% de HP) et un loot garanti exceptionnel.

Rang Gate	Types de monstres	Boss exemple
E	Gobelins, araignées, loups magiques	Roi Gobelin
D	Orcs, squelettes guerriers, esprits	Liche Mineure
C	Trolls, démons mineurs, golems	Démon Contractuel
B	Géants, draconides, nécromanciens	Seigneur Draconide
A	Monstres d'élite, chevaliers déchus	Seigneur du Château
S	Monarches d'ombre, créatures abyssales	Monarque (boss ultime)

7. Inventaire & Équipement

- Inventaire limité à 30 emplacements (extensible avec un sac spécial trouvé en donjon)
- Slots d'équipement : arme, armure (torse, jambes, tête), anneau, collier
- Chaque pièce d'équipement a un rang (E à S), des bonus de stats et une durabilité
- Certains équipements ont des effets spéciaux passifs débloqués selon les stats du chasseur

8. Système de Compétences

Les compétences s'obtiennent en montant de niveau ou en trouvant des **cristaux de compétences** dans les donjons. Chaque compétence peut être améliorée jusqu'au niveau 5.

- Compétences actives : déclenchées manuellement en combat (coût MP)
- Compétences passives : toujours actives (bonus de stats, effets constants)
- Compétences d'éveil : déblocables via des quêtes spéciales (puissantes, à usage limité)

9. Quêtes & Système de Missions

L'Association des Chasseurs propose des **quêtes journalières** et des **missions spéciales** qui donnent des récompenses bonus (XP, cristaux, titres). Exemples de quêtes :

- Fermer 3 gates de rang D ou supérieur
- Éliminer 50 gobelins au total
- Survivre à un donjon rang C sans utiliser de potion
- Découvrir une salle secrète dans un donjon

10. Événements Aléatoires en Donjon

Pendant l'exploration, des événements aléatoires peuvent survenir sur chaque étage :

- Salle au trésor cachée (détectée avec la stat PER)
- Piège magique : dégâts ou malus de stats temporaires
- Renfort de monstres : le groupe ennemi s'agrandit en cours de combat
- Monstre Elite : version renforcée d'un monstre normal, loot amélioré
- Dungeon Break imminent : chrono — le chasseur doit fuir ou vaincre le boss rapidement

■■ ARCHITECTURE POO (Python)

Modules & Classes principales

Module	Classes principales	Responsabilité
hunter.py	Hunter, Stats, Inventory	Données & logique du chasseur
dungeon.py	Dungeon, Floor, Gate, Room	Génération & structure du donjon
combat.py	CombatEngine, Turn, CombatResult	Moteur de combat tour par tour
monster.py	Monster, Boss, MonsterFactory	Modèles et factory de monstres
skill.py	Skill, ActiveSkill, PassiveSkill, SkillTree	Système de compétences
item.py	Item, Equipment, Consumable, Crystal	Objets et équipements
quest.py	Quest, QuestBoard, QuestTracker	Quêtes et missions
save.py	SaveManager	Sauvegarde / chargement JSON
display.py	TerminalUI, Colors, AnimEngine	Affichage terminal coloré
main.py	Game (orchestrateur)	Boucle principale du jeu

■ Utiliser les concepts POO : héritage (*Monster* → *Boss*), polymorphisme (*Skill* → *Active/Passive*), encapsulation (*stats* privées avec *getters*), et composition (*Hunter* contient *Inventory*, *Stats*, *SkillTree*).

■ CRITÈRES D'ÉVALUATION PERSONNELS

Critère	Importance
POO : héritage, polymorphisme, encap.	■■■■■
Génération procédurale des donjons	■■■■
Système de combat complet & équilibré	■■■■■
Système de rangs fidèle à l'univers	■■■■
Affichage terminal lisible & stylisé	■■■
Sauvegarde / chargement fonctionnel	■■■
Gestion des erreurs (pas de crash)	■■■■
Code modulaire (fichiers séparés)	■■■■
Originalité des boss et événements	■■■

■■ PLANNING SUGGÉRÉ (2 semaines)

Jour(s)	Objectif
J1 – J2	Architecture POO : créer tous les squelettes de classes et modules
J3 – J4	Système de stats, montée de niveau, inventaire et équipement
J5 – J6	Génération de donjons (étages, monstres, boss) et navigation
J7 – J8	Moteur de combat complet (tours, compétences, consommables)
J9	Événements aléatoires, pièges, salles secrètes
J10	Système de quêtes et boutique de l'Association
J11 – J12	Sauvegarde JSON et menu principal complet
J13 – J14	Tests, équilibrage, polish affichage terminal, bonus

■ FONCTIONNALITÉS BONUS (si le temps le permet)

- Mode Raid : affronter un boss de rang S avec un groupe de 4 chasseurs contrôlés par l'IA
- Système de Guilde : rejoindre ou créer une guilde, missions de guilde collectives
- Rang Spécial National : un classement des chasseurs nationaux consultable en jeu
- Easter egg : un événement secret déclenchable sous condition très précise...

■ RESSOURCES UTILES

- Couleurs terminal Python : bibliothèque *colorama* ou codes ANSI
- Génération aléatoire équilibrée : module *random* + tables de probabilités
- Sauvegarde : module *json* standard Python
- Affichage structuré : bibliothèque *rich* (tableaux, barres de progression)
- Wiki Solo Leveling : <https://solo-leveling.fandom.com> (lore & mécaniques)