

# **BASH ADVENTURE**

Aristide-Ladal Tidiane

Rédigé le: 14/11/2021

# Table des matières

<b>1</b>	<b>Présentation</b>	<b>3</b>
<b>2</b>	<b>Pré requis et touches</b>	<b>3</b>
2.1	Pré requis . . . . .	3
2.2	Touches . . . . .	3
<b>3</b>	<b>Fonctionnalités</b>	<b>4</b>
3.1	Configuration du personnage . . . . .	4
3.2	Création et Visualisation des plateaux . . . . .	4
3.3	Téléporteur . . . . .	8
3.4	Création et Combats des Ennemies . . . . .	8
3.5	Interaction avec les PNJ et Gestion de l'argent . . . . .	10
3.6	Gestion des ressources . . . . .	11
3.7	Gestion de la vie de l'expérience et du niveau du personnage . . . . .	11
3.8	Gestion de l'amélioration . . . . .	12

# 1 Présentation

Ce travail a pour objectif d'analyser la performance de plusieurs algorithmes de Tri qui seront eux même présentés et analysés en amont lors de l'expérience. Cette efficacité sera en majeure partie mesurée par leur temps d'exécution, et sera mise en perspective avec leur application dans la vie réelle.

Bash-Adventure est un jeu d'aventure codé en langage C et s'exécutant dans un terminal. Il dispose d'une interface ergonomique composée de multiples menus interactifs permettant de naviguer dans les fonctionnalités.

**Dans ce rapport nous nous concentrerons point par point sur les fonctionnalités que nous avons implantées pour réaliser ce projet**

## 2 Pré requis et touches

### 2.1 Pré requis

Ce projet contient 5 fichiers, *getkey.c*, *plateau.c*, *prepa.h*, *grid.py*, et *main.c* nous rentrerons dans les détails de ces dossiers plus tard dans la présentation. Afin de **compiler** *main.c* il vous faut avoir un ordinateur tournant sous **Linux**, aucune librairie n'est à installer néanmoins la librairie *math.h* étant utilisée, l'option "-lm" devra être précisée lors de la compilation du programme

### 2.2 Touches

- **Déplacement :**

**Z** : haut

**S** : bas

**Q** : gauche

**D** : droite

*Il n'est pas nécessaire d'appuyer sur entrée, il faut simplement appuyer sur les touches.*

- **inventaire :**

**E** : Entrée dans le menu de l'inventaire

**E** : Sortir du menu de l'inventaire (si dans l'inventaire)

*Il n'est pas nécessaire d'appuyer sur entrée, il faut simplement appuyer sur les touches.*

- **Caractéristiques :**

**C** : Entrée dans le menu de caractéristiques

**C** : Sortir du menu de caractéristiques (si dans le menu de caractéristiques)

*Il n'est pas nécessaire d'appuyer sur entrée **hormis lors de la précision des nombres de points à attribuer.***

- **Pnj :**  
**Leur rentrer dedans :** Engager l'interaction  
**R :** permet de faire retour quelque soit le point de l'interaction ou nous sommes, R permet ainsi aussi de sortir de la discussion  
**Autres :** Les Autres interactions liés aux pnj sont indiqués in game  
**NOTES :** Lors du lancement de l'interaction *Discuter* il n'est pas nécessaire de faire retour celle-ci fait retour automatiquement après 3 secondes.  
*Il est nécessaire d'appuyer sur entrée*
  
- **Téléporteurs :**  
**Leur rentrer dedans :** Se faire téléporter  
*Il est nécessaire d'appuyer sur entrée*
  
- **Combat :**  
**E :** Sortir du menu de l'inventaire (si dans l'inventaire)  
**Autres :** Les Autres interactions liés aux pnj sont indiqués in game  
*Il est nécessaire d'appuyer sur entrée **hormis pour sortir de l'inventaire.***
  
- **Téléporteurs :**  
**Leur rentrer dedans :** Se faire téléporter  
  
*Il n'est pas nécessaire d'appuyer sur entrée **hormis lors de la précisions du nombres de points à attribuer.***

## 3 Fonctionnalités

### 3.1 Configuration du personnage

La configuration du personnage se fait au lancement du jeu, elle nous permet de choisir, le pseudo, le sexe et la caste du personnage.

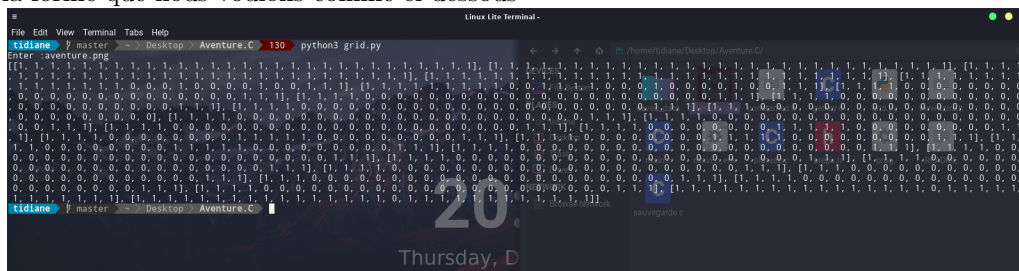
Les deux premiers choix n'auront pas de vrai impact sur le gameplay mais le choix de la caste si. Il changera l'attaque spéciale du joueur ainsi que ses caractéristiques de base, de plus l'évolution du personnage elle aussi changera en fonction de la caste comme nous le verrons plus tard dans la subsection "Gestion de l'amélioration"

### 3.2 Création et Visualisation des plateaux

Pour ce jeu 3 plateaux ont été créés le premier et principale dans lequel notre joueur commencera la partie, un deuxième accessible par le téléporteur à l'est

du plateau principal et enfin un troisième accessible par le téléporteur au Sud du plateau principal.

Ses plateaux sont créés à partir de matrice 2D créée à l'aide du fichiers "grid.py" que nous nous sommes permis d'inclure dans le projet même si écrit en python car étant de notre confection.// grossièrement, ce fichier transforme une image en matrice 2D en prenant en séparant l'image en case de 64x64 pixel et en regardant la valeur RGB de chacune de celle-ci inscrivant ainsi dans la matrice "1" si la case est rouge et 0 autrement. Programme nous permettant ainsi de créer très facilement des plateaux de grandes tailles juste en coloriant en rouge la forme que nous voulons comme ci-dessous

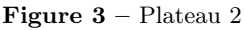


**Figure 1** – Sortie du programme "grid.py"

Ses plateaux sont ensuite afficher avec la fonction *Afficher map (ligne 657, prepa.h)* dans le terminal en remplaçant les 1 par un caractères vides avec un fond rouge, les 0 par un caractère vides simples, le joueur par le caractère J de fond vert, les Pnjs par la chaîne PNJ de fond vert et enfin les téléporteurs par le caractère 0 de fond blanc.



**Figure 2** – Plateau Principal



**Figure 3** – Plateau 2



Figure 4 – Plateau 3

### 3.3 Téléporteur

Les Téléporteur sont placés dans la matrice2D correspondant à la map ou l'on veut les placer sous la forme suivantes 5(mapid)(X)(Y) ou map correspond à l'id de la map ou l'on veut que le téléporteur nous emmène, X et Y eux sont les coordonnées ou le joueur doit être téléporteur dans cette map.

### 3.4 Création et Combats des Ennemies

Pour les ennemis nous avons voulu partir partir sur un système comme les jeux *Pokemon*, ou encore *Dragon Quest*, c'est à dire un déclenchement automatique des combats lorsque l'on se déplace dans certaines zones.

Ainsi les Ennemies pouvant être créés très facilement grâce à la structure leur correspondant et la fonction *CreeEnnemi* ligne 566 *prepa.h* ont une chance sur huit d'apparaître à chacun de nos déplacements dans certaines plateaux, c'est à dire dans nos cas tous, hormis le plateau principal.

Lorsque cette chance sur huit survient se lance alors un combat au tour par tour très classique. Le plateau de "combat" s'affiche en haut puis dans la zone d'informations, les informations concernant notre personnage ainsi que



l'ennemi sont affichés.

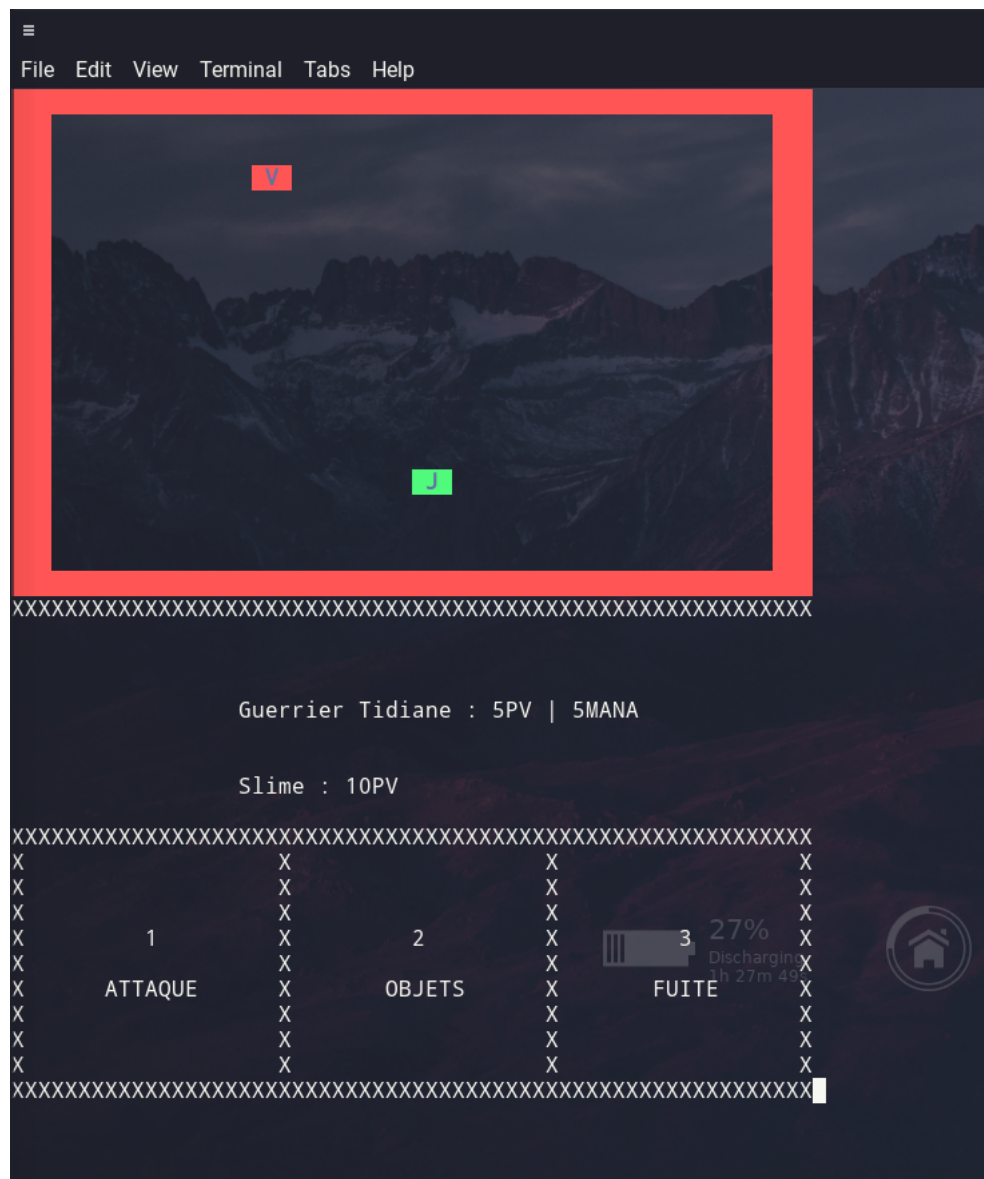


Figure 5 – Plateau Principal

Nous avons ainsi le choix à chaque tour entre attaquer ouvrir l'inventaire ou fuir. A la fin de notre tour l'ennemi nous attaquera et nous fera ainsi prendre des dégâts définis selon la défense du joueur, et la Force de l'ennemi définit lors de sa création.

L'option Attaquer nous fera alors choisir l'attaque que l'on veut entre l'attaque basique commune à toutes les castes et dont les dégâts dépendent de la force ou (attackpower dans le code) de notre personnage et de la defense de l'ennemi, et l'attaque spéciale elle propre à chaque caste. Boule de Feu pour le Magicien dépendant du mana de celui-ci, Fracas pour le Guerrier dépendant de la force de celui-ci et Surnoiserie pour l'Assassin dépendant de la force et de l'agilité de celui-ci.

L'option Objet ouvrira l'inventaire néanmoins aucun objet impactant les combats n'a eu le temps d'être implanter donc elle n'est là que pour la forme

Enfin l'option fuir nous permettra de fuir le combat de façon aléatoire mais plus ou moins probable selon que la différence de niveau entre l'adversaire et le joueur soit plus ou moins grande.

A la fin du combat on gagne un nombre d'or et d'expérience propre à chaque ennemi et définis lors de la création de celui-ci ainsi qu'un objet dropper par le monstre et définit lui aussi à sa création.

Si on gagne assez d'expérience on monte de niveaux(nous reviendra sur ce point plus en détail dans la subsection Gestion de l'amélioration)

Si les points de vie du personnage tombe à 0, le personnage meurt, est téléporter dans le plateau principale et perd la moitié de son Or

### 3.5 Interaction avec les PNJ et Gestion de l'argent

Les PNJ sont créés très facilement grâce à la structure de données les correspondant et à la fonction *AjoutPnj ligne 606 prepa.h*.

Nous avons la possibilités d'interagir avec ceux-ci de deux manières, en discutant avec eux ou en commerçant avec eux.

L'option Discuter nous permet de visualiser la ligne de dialogue propre à chacun d'entre eux et attribuer lors de la création du dit pnj.

L'option Commercer elle, nous permet d'acheter ou vendre des objets au PNJ. En effet lors de la création du joueur on reçoit 0 Or celui-ci peut être gagnés comme dit précédemment en combattant des monstres ou encore en vendant des objets récoltés encore une fois sur les monstres. Ainsi l'on peut vendre des objets pour de l'or et acheter des objets contre de l'or.

### 3.6 Gestion des ressources

Les ressources ou objets sont récoltables sur les montres ou échangeables contre de l'or auprès des PNJ. Les objets en notre possession peuvent être visualiser dans notre inventaire en appuyant sur la touche E (cf Touches et Pré requis).

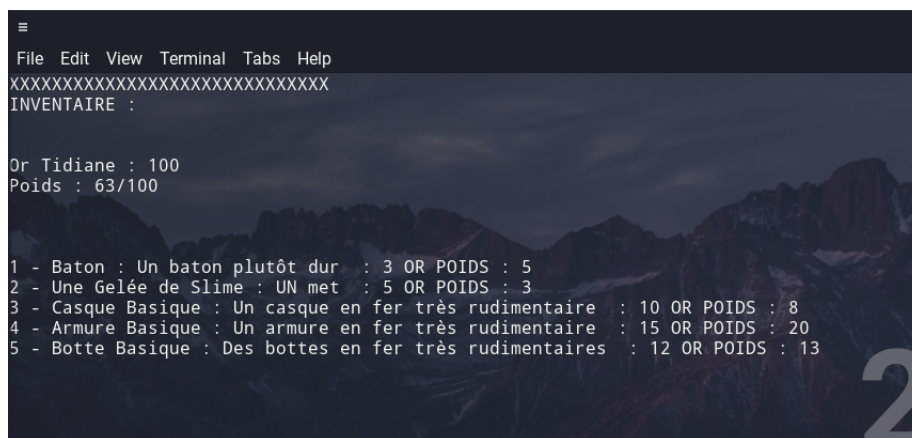


Figure 6 – inventaire du Joueurl

Il n'y a pas de limite du nombres d'objets pouvant être posséder à proprement parler néanmoins il existe un système de "poids" limitant ceux-ci. Chaque objet possède un poids qui lui est propre et notre joueur a un poid maximal qu'il peut porter. Ainsi si il devient trop lourd et que sa limite de poids est atteinte il ne pourra plus acheter d'objets auprès des PNJ ou dropper d'objets sur les monstres. Néanmoins cette limite de poids est liées à la caractéristiques Force de notre personnage ainsi plus notre joueur aura de force plus son poid maximal sera élevée et plus il pourra voir d'objets.

### 3.7 Gestion de la vie de l'expérience et du niveau du personnage

Comme expliquer précédemment les caractéristiques sont dépendantes de la caste du personnage, la vie cependant est commune et démarre à 5pdv pour chaque personnage, cette caractéristiques monte de 5 points à chaque nouveau niveau et peut être augmenter par des points de caracteristiques libres (nous reviendra sur ce point plus en détail dans la subsection Gestion de l'amélioration).

L'expérience elle monte à chaque combat de monstres victorieux comme vu

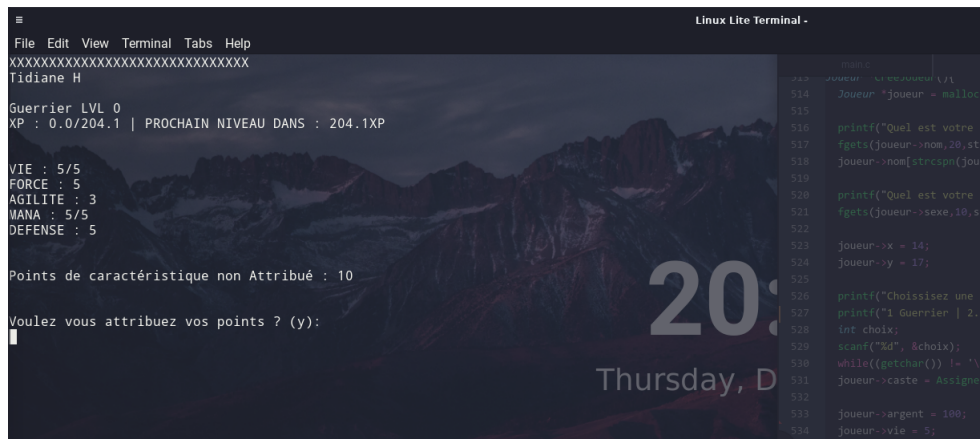
précédemment.

Le niveau du personnage est calculé par la formule suivante :  $\sqrt{xp_{\text{Joueur}} * 0.0049}$ . Cette formule trouvée sur Internet permet de faire évoluer le personnage de façon raisonnable mais rapide tout de même

### 3.8 Gestion de l'amélioration

L'amélioration du personnage se fait à travers les caractéristiques de celui-ci. En effet à chaque montée de niveau le joueur gagnera en fonction de sa classe plus ou moins de point de caractéristiques dans certains attributs. Par exemple le Guerrier gagnera 5 Force à chaque niveau alors que le Magicien en gagnera 1 et l'Assassin 2 etc...

En plus de ses points dits "attribués" le joueur gagnera à chaque niveau 5 points non attribués (pointlibre dans le code) qui sont donc des points que le joueur pourra attribuer comme il lui semble dans le menu de caractéristiques, permettant ainsi de créer une infinité de manière de faire évoluer son personnage.



```
Linux Lite Terminal -
File Edit View Terminal Tabs Help
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Tidiane H

Guerrier LVL 0
XP : 0.0/204.1 | PROCHAIN NIVEAU DANS : 204.1XP

VIE : 5/5
FORCE : 5
AGILITE : 3
MANA : 5/5
DEFENSE : 5

Points de caractéristique non Attribué : 10

Voulez vous attribuer vos points ? (y):

```

```
512 menu
513 joueur = *joueurnew();
514 Joueur *joueur = malloc(
515
516     printf("Quel est votre nom\n");
517     fgets(joueur->nom, 20, stdin);
518     joueur->nom[ strlen(joueur->nom) ] = '\0';
519
520     printf("Quel est votre sexe\n");
521     fgets(joueur->sexe, 10, stdin);
522
523     joueur->x = 14;
524     joueur->y = 17;
525
526     printf("Choisissez une classe\n");
527     printf("1 Guerrier | 2 Magicien | 3 Assassin\n");
528     int choix;
529     scanf("%d", &choix);
530     while((getchar()) != '\n')
531         continue;
532     joueur->classe = AssignerClasse(choix);
533
534     joueur->argent = 100;
535     joueur->vie = 5;

```

Figure 7 – Menu de caractéristiques