*PROJET LANGUAGE C*

*Bachelor 1*

*RPG*

-==\\ `//~\\ ~~~~`---.\_\_\_.-~~

\_\_\_\_\_\_-==| | | \\ \_-~`

\_\_--~~~ ,-/-==\\ | | `\ ,'

\_-~ /' | \\ / / \ /

.' / | \\ /' / \ /'

/ \_\_\_\_ / | \`\.\_\_/-~~ ~ \ \_ \_/' / \/'

/-'~ ~~~~~---\_\_ | ~-/~ ( ) /' \_--~`

\\_| / \_) ; ), \_\_--~~

'~~--\_/ \_-~/- / \ '-~ \

{\\_\_--\_/} / \\\_>- )<\_\_\ \

/' (\_/ \_-~ | |\_\_>--<\_\_| |

|0 0 \_/) )-~ | |\_\_>--<\_\_| |

/ /~ ,\_/ / /\_\_>---<\_\_/ |

o o \_// /-~\_>---<\_\_-~ /

(^(~ /~\_>---<\_\_- \_-~

,/| /\_\_>--<\_\_/ \_-~

,//('( |\_\_>--<\_\_| / .----\_

( ( ')) |\_\_>--<\_\_| | /' \_---\_~\

`-)) )) ( |\_\_>--<\_\_| | /' / ~\`\

,/,'//( ( \\_\_>--<\_\_\ \ /' // ||

,( ( ((, )) ~-\_\_>--<\_~-\_ ~--\_\_\_\_---~' \_/'/ /'

`~/ )` ) ,/| ~-\_~>--<\_/-\_\_ \_\_-~ \_/

.\_-~//( )/ )) ` ~~-'\_/\_/ /~~~~~~~\_\_--~

;'( ')/ ,)( ~~~~~~~~~~

' ') '( (/

' ' `

|  |
| --- |
| **Nombre de personnes par équipe :** Entre 2 et 4.  **Date de rendu :** Mercredi 11 Mars avant minuit  **Sujet :**  Vous devez créer un jeu de type RPG retro-style au tour par tour sur console.  Voici le synopsis : « Votre nom est Bob. Vous vous réveillez au milieu d’une forêt avec un sac ne contenant qu’un seul objet : un couteau. Vous devrez gagner de l’XP et récupérer de nouvelles armes pour devenir plus fort et battre le boss pour sortir de la forêt. »  Voici comment marche le jeu :  Le jeu commence avec un menu comme ci-dessous (l’option « Load Saved Game » servira dans les rajouts bonus du projet)  Le bouton « about » permet de connaitre le nom des personne qui ont travaillé sur le projet et donne un petit tutoriel du jeu.  Le bouton « Exit » fermera l’application.    En commençant le jeu, on doit écrire son nom.  On contrôle le jeu en tapant du texte permettant d’aller au nord, à l’est, à l’ouest et au sud. Quand on arrive sur un nouvel endroit, on a une description du lieu et en fonction du lieu on peut soit trouver un objet soit trouver un monstre.    La map ne doit pas être générée aléatoirement et le placement du boss est défini dans le code.  Une fois le boss vaincu le jeu, ou que le personnage n’a plu de vie, le jeu s’arrête en affichant le message correspondant et en revenant au menu principal. Le joueur peut aussi quitter le jeu à tout moment et revenir au menu principal.  Le personnage a un niveau, une jauge d’expérience (XP), de la vie (HP), des compétences d’attaque et de défense. A chaque combat gagné, il gagne de l’XP pour gagner un niveau (plus le niveau est suivant est élevé, plus il y a besoin d’XP pour l’atteindre). Quand il gagne un niveau ses autres caractéristiques augmentent (HP, Attaque et Défense).  Les monstres ont aussi un niveau, du HP, des compétences d’attaque et de défense. Plus le niveau est élevé, plus ses caractéristiques le sont aussi. Le plus fort étant le boss.  Mode combat :  On commence par indiquer le nom et le niveau du monstre à affronter.  Puis on peut attaquer, utiliser un objet de l’inventaire ou courir pour échapper au combat (mais ne rien gagner).  L’attaque du personnage cause un nombre de dommage dépendant des compétences d’attaque du personnage et des compétences de défense du monstre (et inversement quand le monstre attaque). Il y a aussi une chance aléatoire (à définir) que l’attaque manque ou que l’attaque cause un coup critique (le double de dommage).  Les objets dans l’inventaire sont ceux que l’on peut ramasser (Potions, Attack Boost, Defense Boost, Etc…). Une fois utilisé, il est supprimé de l’inventaire. La potion permet de regagner de la vie, et l’Attack Boost et le Defense Boost augmentent les caractéristiques pendant la durée du combat. On peut revenir en arrière s’il ne veut finalement pas utiliser d’objets mais attaquer.  On ne peut utiliser qu’une action à chaque tour (attaque ou objet). Puis le monstre attaque et ainsi de suite jusqu’à que l’un des deux n’ai plus de vie.  **Exemple de points bonus :**   * Ajouter une histoire et des indices sur l’emplacement du Boss. * Ajouter une sauvegarde du jeu. Sauvegarder à n’importe quel moment hors combat les données du joueur (son lieu actuel, son niveau, son XP, son HP, ses caractéristiques attaque et défense, son nom) dans un fichier. Ajouter au menu principal le fait de commencer une partie existante en allant chercher la partie voulue dans le fichier. * Ajouter un principe d’armes. Le joueur aura la possibilité de changer son arme de départ pour une nouvelle ou d’en avoir une deuxième pour attaquer. La nouvelle arme peut augmenter l’attaque mais aussi augmenter le nombre chance de rater son coup. Les armes apparaissent comme les autres objets mais plus rarement. Une liste d’armes est déjà définie au départ et une fois qu’une arme est récupérer, on ne peut plus la trouver.   Exemple du mécanisme de combat avec 2 armes : |
|  |