Explications Fonctions

Structures :

STRUCTURE PERSONNAGE : Nous avons déclaré la création de la structure a l’aide des mots type et struct ainsi que le nom de la structure ici en particulier le nom de la structure est la structure personnage. Nous avons donc initialisé les attributs nécessaire a la création du personnage :

* Nom
* Classe
* Niveau
* Exp
* Expmax
* Pv
* Pvmax
* Pa
* Inventaire
* Skill
* Gold
* Equipement
* Initiative hé également
* Mana
* Manamax
* Upgrade
* Inventairemax

STRUCTURE EQUIPEMENT : Qui elle est lié a la structure personnage et nous a permis l’apparition des attributs casque plastron bottes.

FONCTIONS liés au Personnage :

FUNC MAIN : Dans notre fonctions main nous avons uniquement mis notre fonctions menu car c’est autour de cette fonction que tout le jeu vas ce passer et elle sera donc la principale.

FUNC Menu: Maintenant passons a la fonctions principale qui est donc la fonctions main, au début de cette fonction nous avons permis a l’utilisateur d’initialiser son personnage, puis une fois cela fait nous avons réaliser une boucle infini sur le menu qui nous proposera les 7 mêmes choix en continue a part si l’on décide de quitter le jeu. Donc nous avons tout d’abord proposer au joueur de faire un choix entre créer un personnage et de quitter le jeu. A l’aide d’un switch case et de la fonctions «  fmt.Scanln() » nous avons pu récupérer l’input de l’utilisateur ce qui lui permet de réaliser les choix qu’il désire. Si l’utilisateur décidé de choisir la première possibilité il serait renvoyer vers la fonctions characterCreation() pour créer son personnage. Une fois ceci fini c’est ici que la bouche « infini » vas commencer et nous proposera les choix suivant :

* Accéder aux informations du personnage
* Accéder a l’inventaire du personnage
* Accéder au marchand
* Accéder au Forgeron
* Accéder au menu combat
* Voir qui sont les personnalités cachés
* Quitter le jeu

Nous n’avons pas rencontré de difficultés particulière pour cette fonctions si ce n’est le fait qu’il fallait très souvent y ajouter des choses au fur et a mesure que le projet avancé.

FUNC CharacterCreation : Comme dit précédemment cette fonction est lié a la précédente car en effet c’est dans celle-ci qui comme son nom l’indique nous allons crée notre personnage. Nous avons donc commencé par déclarer les variable classe personnage et name pour pouvoir la réalisé. Dans un premier temps nous avons récupérer l’input de l’utilisateur pour récupérer le nom qu’il souhaité donner a son personnage. Nous avons aussi créer une fonction verifname() que nous avons utilisé dans cette fonction pour vérifier si le nom donner par l’utilisateur contenait des caractères interdits ou non, si s’était le cas cela nous retourner le booléen false et a l’inverse cela nous retournait le booléen true. Une fois ceci fait si nous obtenons true tout avons écris des conditions permettant de réécrire le nom de l’utilisateur pour qu’il commence par une majuscule et que les caractères qui suivent soit en minuscule. Pour ce faire nous avons simplement commencé par vérifier si le premier caractère du mot était une majuscule si oui nous ne faisions rien et a l’inverse nous le mettons en minuscule en lui enlevant 32 comme on a pu le voir dans la table ascii. Une fois cette étape faites nous devons maintenant savoir quelle classe l’utilisateur souhaité donner a son personnage, nous lui avons donc proposer 3 classe parmi lesquelles se trouve la classe Samurai, Assassin et la classe Indigène chacune d’elles vont avoir des capacités, objets, statistiques différentes. Donc selon le choix que l’utilisateur faisait les attributs du personnage était rempli par ceux qui étaient prédéfinis dans chacune des classes.

Nous avons rencontré des difficultés sur cette fonctions car si l’utilisateur mettez un autre chiffre non accepter par le switch case cela bugué le programme. Pour pallier a ce problème nous aurions tout simplement du créer une fonction avec le choix de classe a part et l’appeler dans la fonction charactercreation et tout simplement si l’input de l’utilisateur ne correspondait pas rappeler la fonctions choixClasse.

FUNC DisplayInfo : Cette fonction elle est plutôt simple car elle servait uniquement a retourner a l’utilisateur toutes les informations du personnage qu’il venait de créer. Pour ce faire il nous suffisait d’appeler les attributs a l’aide du nom que nous avons donné au personnage, pour que les attributs prennent la valeur du personnage.

Aucune difficultés rencontrés sur cette fonction

FUNC Dead : Cette fonction elle servait uniquement à montré la mort du personnage et bien sûr le faire se ressuscité. Pour ce faire nous avons tout simplement vérifier que les pv du perso n’était pas inférieur a 0 et si s’était le cas il mourrait et ressuscité avec la moitié de ses pv. Cette fonction est appelle uniquement lorsque dans le code il y a un risque que le personnage perde de la vie et meurt.

Aucune difficultés rencontrés sur cette fonction

Fonctions liés à l’inventaire :

FUNC accessInventory : Cette fonction qui comme son nom l’indique nous permet d’accéder a l’inventaire. Cette fonction est une des fonctions principale en effet nous pouvons accéder a cette fonction a partir du menu principal. Une fois dans cette fonction nous avons 9 choix, donc 9 switch case. Qui sont chacun des fonctions :

* Display inventory
* drinkPotion(« Potion de vie »)
* drinkPotion(« Potion de mana »)
* drinkPotion(« Potion de poison »)
* spellBook()
* equipItem()
* unequipItem()
* upgradeInventory()
* Un retour au menu

Aucune difficultés rencontrés sur cette fonction

FUNC displayInventory : Cette fonction permet d’afficher les items présent dans l’inventaire du personnage choisi par l’utilisateur. Nous avons aussi soigné l’affichage de cette fonction en créant une sorte de boite dans laquelle nous affichons chaque item un par un.

Aucune difficultés rencontrés sur cette fonction

FUNC addInventory : Cette fonction permet d’ajouter un item a l’inventaire, elle est apellé dans toutes les fonctions ou nous avons besoin d’ajouter un item a l’inventaire du personnage. Par exemple nous l’avons utilisé dans les fonctions marchand, forgeron, unequipItem et d’autres encore. Dans cette fonction nous vérifions au début si la longueur de l’inventaire est bien inférieur a la longueur de l’inventaire maximum. Si c’est le cas nous pouvons utiliser append est ajouter l’item dans l’inventaire a l’inverse nous allons retourner a l’utilisateur que l’inventaire est plein.

Aucune difficultés rencontrés sur cette fonction

FUNC removeInventory : Nous avons aussi eu besoin d’une fonction remove inventory et comme son nom l’indique elle nous permet de retirer un item de l’inventaire. Cette fonction est utilisé dans les fonctions forgeron, vente, usePotion et d’autres encore. Pour réaliser cette fonction nous parcourons l’inventaire et nous cherchons l’item a retirer une fois l’item trouver nous le retirons a l’aide de la fonction append.

Aucune difficultés rencontrés sur cette fonction

FUNC upgradeInventory : La fonction upgradeInventory permet de passer la taille de l’inventaire de base a 10 a 20, 30 jusqu’à 40 car l’utilisateur pourras améliorer l’inventaire uniquement 3fois. Pour ce faire dans la fonction nous regardons dans un premier si l’attribut upgrade du personnage est bien inférieur a 3. Si c’est le cas nous vérifions ensuite si le personnage dispose d’au moins 30 or, si c’est le cas nous ajoutons 10 a l’inventaire max ce qui fait que l’utilisateur aura 10 places de plus pour ses items dans son inventaire. Si le personnage n’avait pas assez d’or cela retournera juste qu’il n’a pas assez d’or et si l’attribut du personnage upgrade est égal a 3, cela lui retournera que toutes les upgrades ont été utilisé pour l’inventaire.

Aucune difficultés rencontrés sur cette fonction

FUNC inInventory() : La fonction inInventory permet de voir si un item donné est déjà dans l’inventaire. Pour ce faire nous parcourons l’inventaire en recherchant l’item entré en paramètre. Si un item est présent cela retourne true sinon false. Cette fonction est utilisé dans drinkPotion , sell, buySpell et beaucoup d’autres encore

Aucune difficultés rencontrés sur cette fonction

Fonctions Potions :

FUNC drinkPotion : Cette fonction est la fonction principale des fonctions se rapportant aux potions. Elle sert come son nom l’indique a boire une potion donné comme choix il y a :

-Potion de vie

-Potion de mana

-Potion de poison

Pour réaliser cette fonction nous regardons dans un premier temps si la potion est dans l’inventaire, si c’est le cas nous appelons la fonction lié a la potion puis nous utilisons la fonction removeInventory pour retirer la potion dites de l’inventaire. Et si dans l’inventaire il se trouve qu’il n’y est pas de potion alors cela retournera a l’utilisateur qu’il n’a pas la potion qu’il a demandé dans son inventaire.

Aucune difficultés rencontrés sur cette fonction

FUNC potionVie : Cette fonction permet uniquement de rajouter au personnage 50pv si il utilise cette fonction. Pour la réaliser nous regardons dans un premier temps si les pv du personnage sont bien inférieur au pv max. Si c’est le cas nous rajoutons 50pv au personnage, suite a ca nous faisons en sorte que les pv ne dépasse pas les pv max en mettant les pv égaux au pvmax uniquement si les pv sont supérieur aux pv max. Sinon cela retourne simplement que le personnage a déjà le maximum de pv et cela ne lui enlève pas de potions.

FUNC potionPoison : Cette fonction elle permet d’enlever 30 pv au personnage en affichant toutes les secondes le nombre de pv qui perds soit ici précisément 10pv. Pour réaliser cette fonction nous vérifions dans un premier temps que les pv du personnage sont bien supérieur a 0. Puis nous avons réaliser une boucle qui se réitéré 3fois et enlève 10pv toutes les secondes. A la fin de cette boucle nous vérifions que les pv ne sont pas infèrieur a 0 si c’est le cas nous appelons la fonction dead().

Func potionMana : Cette fonction permet uniquement de rajouter au personnage 20mana si il utilise cette fonction. Pour la réaliser nous regardons dans un premier temps si le mana du personnage sont bien inférieur au mana max. Si c’est le cas nous rajoutons 50mana au personnage, suite a ca nous faisons en sorte que les mana ne dépasse pas les mana max en mettant les mana égaux au manamax uniquement si les mana sont supérieur aux mana max. Sinon cela retourne simplement que le personnage a déjà le maximum de mana et cela ne lui enlève pas de potions.

Fonctions sorts :

Func spellBook : Cette fonction permet au personnage d’apprendre un sort. Pour ce faire nous vérifions si le sort est déjà appris pour réaliser ceci nous appelons la fonction inSpell(). Si le spell est déjà appris nous retournons a l’utilisateur que le sort est déjà appris. Sinon nous vérifions dans un premier temps que le spell se trouve bien dans l’inventaire a l’aide de la fonction inInventory() si c’est le cas nous ajoutons le spell dans l’attribut skill a l’aide de la fonction addSpell et ensuite nous utilisons la fonction removeInventory pour retirer le sort appris de l’inventaire.

Func addSpell : Cette fonction permet d’ajouter a l’attribut skill un spell mis en paramètre. Pour ce faire nous utilisons la fonction append.

Func inSpell : Cette fonction est très similaire a la fonction inInventory car elle vas être réaliser et vas réaliser la même chose sauf que ce sera situé dans l’attribut skill a la place de l’attribut inventaire.

Func coupDePoing : Cette fonction est apellé lors du combat lorsque le personnage vas réaliser une attaque basique c’est cette fonction qui sera apellé et qui infligera les dégâts au monstre.

Func bouleDeFeu : Cette fonction est elle appellé lors du combat lorsque le personnage vas réaliser l’attaque bouleDeFeu dans la catégorie sorts. Tout d’abord dans cette fonction nous allons vérifier si le spell ( «  Boule de Feu ») se trouve dans les skill, a l’aide de la fonction inSpell. Suite a ca nous avons mis une autre condition qui est si le personnage a plus de 10 mana alors il pourra utiliser le sort et infligé 20 dégâts au monstre. Il perdra donc 10 de mana suite a ca. Sinon cela retourne a l’utilisateur qu’il n’a pas assez de mana et donc cela revient sur la fonction charTurn car il ne peut pas utiliser ce sort. Et si aucunes conditions n’est remplie alors cela retourne que le sort n’a pas été appris et donc cela revient une fois de plus sur le charTurn.

Fonctions Marchand :

Func Marchand : Cette fonction est la fonction principale des fonctions marchand car c’est a partir de celle-ci que l’on vas pouvoir choisir d’acheter un item avec la fonction buy(), de vendre un item avec la fonction sell() et enfin de retourner au menu principal. Cette fonction est une fois de plus réaliser avec un switch case et la fonction fmt.scanln qui permet de récupérer l’input utilisateur.

Func buy : Cette fonction permet de choisir qu’est ce que l’on vas acheter en effet on aura le choix entre acheter des potions avec la fonctions buyPotion, acheter des spell avec la fonction buySpell, mais aussi d’acheter les items nécessaire a la fabrication d’équipement avec la fonctions buyItemForgeron et enfin on l’utilisateur pourra décider de retourner au menu.

Toutes les fonctions buyPotions, buySpell, buyItemForgeron : on propose dans un premier temps a l’utilisateur de sélectionner l’item qu’il veut acheter puis on vérifie si il a l’or nécessaire et si c’est le cas on lui ajoute dans l’inventaire l’item demandé avec la fonction addInventory, puis on lui retire les gold associé a l’item. Si le personnage n’a pas assez d’or on lui retourne tout simplement qu’il n’a pas assez d’or.

Func sell : Cette fonction permet de vendre 4 item, la potion de vie qui donnera donc 5 or au personnage, la potion de mana qui donnera 5or, la potion de poison qui donnera 5or, et enfin le spell boule de feu qui donnera lui 10 or. Pour vendre un item on regardera dans un premier temps si il se trouve dans l’inventaire a l’aide de la fonction inInventory et si il l’est on ajoute le montant d’or associé a l’item au personnage puis on utilise la fonction removeInventory pour retirer l’item de son inventaire. Sinon cela retour a l’utilisateur qu’il ne dispose pas de l’item dans son inventaire.

Fonctions Forgeron : Cette fonction sert juste a demander a l’utilisateur si il souhaite forger un item ou retourner au menu.

Fonctions forger : Cette fonction propose a l’utilisateur de forger 3 items ayant chacun 3 fonctions associés. L’utilisateur peut décider de forger un casque, un plastron et des bottes. Si il décide de forger l’un d’eux il se retrouve sur la fonction associer a l’équipement qu’il souhaite forger.

Fonctions forgerCasque, forgerPlastron, forgerBottes : Cette fonctions sert comme son nom l’indique a forger un équipement. Pour ce faire nous vérifions dans un premier temps si le personnage dispose des items nécessaires a la fabrication de l’équipement, pour faire ca on utilise une fois de plus la fonction inInventory. Puis on ajoute a l’inventaire l’équipement avec la fonction addInventory, et on s’occupe de retirer tout les items nécessaires a la fabrications de l’équipement avec la fonctions removeinventory. Puis on enlève 20 gold car la création de l’équipement coute 20 or. Si aucune des conditions n’est respectés. Nous retournons ce qui manque a l’inventaire du personnage

Fonctions EQUIPEMENT :

Func equipItem : Cette fonction sert à équiper les items casque, plastron et bottes en fonction du choix fait par l’utilisateur.

Func equipCasque, equipPlastron, equipBottes : C’est fonctions servent a équiper l’item demander. Pour ce faire on regardera dans un premier temps si l’item est present dans l’inventaire avec la fonction inInventory. Puis on le retirera de l’inventaire avec la fonction removeInventory. Suite a ca on rajoutera dans l’attribut equipement (« nom de l’item ») du personnage l’item en question. Une fois équipé on améliorera les statistiques du personnage selon l’item . Et si l’équipement ne trouve pas dans l’inventaire la fonction retourne qu’il n’en a pas dans son inventaire.

Func unequipItem : Cette fonction sert elle déséquiper les items casque, plastron et bottes en fonction du choix fait par l’utilisateur.

Func unequipCasque, unequipPlastron, unequipBottes : Ces fonctions servent a déséquiper l’item demandé. Pour ce faire on regardera dans un premier temps que l’attribut equipement.(« nom item ») n’est pas vide. Si il est vide on utilisera la fonction addInventory pour rajouter le plastron présent dans l’attribut équipement. Ensuite on mettra une string vide a la place de l’équipement. Et on vas donc enlever les statistiques rajouter par l’équipement.

Fonctions Combat :

Func menuCombat : Cette fonction propose a l’utilisateur 5 choix ;

* Combat d’entrainement qui appelle la fonction trainingFight
* Infos personnage qui appelle la fonction displayinfo
* Infos Monstre qui appelle la fonction displayinfomonstre
* Inventaire qui appelle la fonction displayInventory
* Retour au menu

Nous avons crée la structure Monstre qui prends comme attribut ;

* Nom
* Pvmax
* Pv
* Pa
* Initiative

Func initgoblin : Dans cette fonction nous avons initialisé un gobelin qui suit la structure et donc les attributs de la structure Monstre

Func trainingFight : Cette fonction vas être la principale des fonctions combat car elle vas permettre d’initialise le goblin pour le combat contre le personnage de l’utilisateur. Elle vas appeler la fonction fight avec en paramètre le personnage et le monstre.

Func fight : Cette fonction vas gérer le combat entre les 2. On commence la fonction avec une boucle qui vas continuer tant que les pv du personnage ou du goblin sont supérieurs a 0. Ensuite l’ordre du combat vas être décidé selon l’initiative du personnage et du gobelin. Si l’initiative du personnage du gobelin est supérieur a celle du personnage alors il attaquera en premier on vas donc mettre cette condition et donc afficher en premier le tour du gobelin avec la fonction goblinPattern() puis on afficher le tour du personnage avec p.charTurn(). On vérifiera ensuite si les pv du personnage ou du gobelin sont inferieur ou égal a 0 si c’est le cas pour le personnage alors cela affiche défaite et donc la mort du personnage sinon c’est une victoire et le personnage a tué le gobelin. Pour ce faire on appelle la fonction win.

Func goblinPattern : Cette fonction est celle qui vas faire attaquer le gobelin et donc mettre les dégâts au personnage. Et tout les 3 tours le gobelin infligera 2 fois plus de dégâts sur le personnage. Pour ce faire nous avons commencé par créer une variable tour sur laquelle nous rajoutons 1 a chaque fois qu’elle est appelé. Pour ce faire nous avons mis que si le tour modulo 3 = 0 alors le gobelin infligera 2 fois plus de dégâts. Sinon il inflige les dégâts normaux .

Func charTurn : Cette fonction est celle qui vas permettre au personnage d’attaquer le gobelin. Cette fonctions est plus élaborés que la fonction goblinPattern car dans celle-ci nous avons du récupérer l’input de l’utilisateur. Il a le choix d’utiliser soit une attaque normal avec la fonction coupDePoing(), d’utiliser un sort avec la fonction bouleDeFeu() mais aussi d’utiliser des objets.

Func displayInfoMonstre : Cette fonction permet uniquement de montrer a l’utilisateur les statistiques du monstre, a l’aide des attributs de la structure monstre mais aussi de la fonction initGoblin .

Fonctions de level :

Func expUp : Cette fonction permet de regarder si l’exp du personnage est suffisant pour le faire passer de niveau si oui il remet l’expérience a 0 et ajoute ce qu’il faut. Puis a la fin de cette fonction cela apelle la fonction lvlUp

Func lvlUp :Cette fonction sert a une fois que le niveau du personnage a augmenté d’améliorer ses statistiques.

Fonctions Bonus :

Cette fonctions est une fonction bonus demandé par la client qui nous a demandé de rechercher les 2 personnalité caché dans le diaporama donner avec les consignes. Les 2 personnalités cachés sont le groupe ABBA et Steven Spielberg