S 1.02

DEV

Présentation de la base

1 Présentation générale

L'objectif de ce TP est de vous faire découvrir la structure du projet qui vous servira de base pour la SAÉ. Cette base est disponible sur le commun dans le répertoire : "Base". Commencez donc par récupérer ce projet, lancez le sous Lazarus, lancez le programme et explorez son code source.

2 Gestion du personnage

Nous allons commencer par nous intéresser à la partie du code en charge de la gestion du personnage (principalement l'unité unitPersonnage). Tout d'abord, nous allons ajouter un ***cheat-code*** qui simplifiera les tests futurs.

1. Repérez le sous-programme exécuté quand le joueur fixe le nom de son personnage. Modifiez ce sous-programme pour que si le nom saisi est ***Alice***, le joueur démarre l'aventure avec 5000 pièces d'or.

Actuellement, la vie maximum du personnage est fixée à 100 et ne peut pas augmenter, nous allons changer cela.

2. Ajoutez un attribut ***santeMax*** de type entier au type ***Personnage*** et assurez-vous qu'il soit bien initialisé à 150 lors de l'initialisation du personnage.
3. Modifiez les trois sous-programmes de l'unité ***unitPersonnage*** gérant respectivement : le repos, le soin et la régénération pour qu'elles prennent en compte ce nouvel attribut.

3 Déplacements et lieux

Dans nos futurs tests, nous aurons besoin régulièrement de nous rendre à la cantine. Nous allons donc rajouter au joueur la possibilité de se rendre directement de sa chambre à la cantine.

4. Étudiez le code de l'unité ***unitLieu*** afin de comprendre comment sont gérés les déplacements.
5. Dans l'unité ***unitChambre***, modifiez les fonctions ***chambrePremiereFois*** et ***chambreHub*** pour que dans les deux cas, un 4ème choix soit proposé au joueur, celui de se rendre à la cantine.
6. Testez le bon fonctionnement de ce nouveau choix.

4 Gestion de l'inventaire

Nous allons maintenant nous intéresser à la gestion de l'équipement du joueur, c'est-à-dire aux armes et armures. Assez logiquement, c'est l'unité ***unitEquipement*** qui en est en charge. Notre objectif est de créer un nouveau rang d'équipement associé à un nouveau matériau : l'obsidienne.

7. Ajoutez un nouveau matériau : ***Obsidienne*** à l'énumération des matériaux.
8. Modifiez les deux fonctions d'affichage : ***armureToString*** et ***armeToString*** pour qu'elles gèrent ce nouveau matériau.
9. Modifiez le code pour que le multiplicateur de dégâts de l'obsidienne soit de 6.

10. Faites en sorte qu'en saisissant le cheat-code de la partie 1, le joueur commence avec une épée en obsidienne dans son coffre et un plastron en obsidienne sur lui.
11. Vérifiez que tout cela fonctionne bien.

Ce nouveau type d'équipement est pratiquement fonctionnel. Il nous reste à gérer la fabrication de tels objets à la forge.

12. Modifiez la fonction ***recetteToString*** de l'unité ***unitEquipement*** pour que, visuellement, la fabrication d'un tel objet coute 500 po et 50 morceaux de Pukei-Pukei.
13. Dans l'unité ***unitPersonnage***, modifiez les trois sous-programmes ***peuxForger***, ***forgerArme*** et ***forgerArmure*** pour qu'elles gèrent correctement l'équipement en obsidienne.

5 Modification de l'IHM de la forge

Le rajout de l'équipement en obsidienne pose un souci au niveau de l'IHM (interface homme-machine) de l'écran de fabrication d'équipement de la forge (géré par la fonction ***fabricationEquipement*** de l'unité ***unitForge***). Nous allons créer une interface spéciale pour cet écran.

14. Dans l'unité ***unitIHM***, créez une nouvelle procédure ***afficherCadreActionForge()*** avec le code suivant (ne pas faire de copier-coller depuis un pdf !) :

```
procedure afficherCadreActionForge();
begin
        dessinerCadreXY(1,29,74,39,simple,white,black);
        dessinerCadreXY(74,35,198,35,simple,white,black);
        deplacerCurseurXY(74,36);write(' ');
        deplacerCurseurXY(74,37);write(' ');
        deplacerCurseurXY(74,38);write(' ');
        deplacerCurseurXY(74,39);write(#196);
end;
```

15. En vous inspirant de la procédure ***afficherInterfacePrincipale*** de cette unité, créez une procédure ***afficherInterfaceForge*** affichant l'interface complète mais en utilisant notre nouveau cadre d'action à la place de l'ancien. Pensez bien à reporter l'entête de la procédure dans la partie interface de l'unité.
16. Modifiez la fonction ***fabricationEquipement*** de l'unité ***unitForge*** pour qu'elle utilise notre nouvelle interface à la place de l'interface classique.

6 Ajout d'un bonus et d'une recette

Une partie conséquente de la SAÉ portera sur la gestion des recettes à la cantine (gestion actuellement très limitée dans ce projet). Nous allons rajouter une nouvelle recette et un nouveau bonus associé à cette nouvelle recette et sur une nouvelle fonctionnalité : les coups critiques.

17. En cherchant où sont calculés les dégâts réalisés par le joueur lors des combats, ajoutez une probabilité de 10% que ces dégâts soient doublés.
18. Dans l'unité ***unitPersonnage***, ajoutez un nouveau bonus ***Critique*** dans l'énumération adéquate.
19. Modifiez le code réalisé à la question 15 pour que si ce nouveau bonus est actif, le joueur ait 20% de chance que ses dégâts soient doublés (au lieu de 10%).
20. Dans la fonction ***choixRecette*** de l'unité ***unitCantine***, ajoutez une troisième recette : "Cuisse de poulet" donnant le buff "Critique".
21. Testez cette nouvelle recette. Vous constaterez qu'un bonus étrange apparaît quand vous mangez ce nouveau plat ! Trouvez pourquoi et corrigez le problème.