mkRPG

Ce document présente le projet proposé et spécifie certains aspects du travail à réaliser. Chaque point mentionné peut être amené à évoluer, voire à disparaître.

L'équipe qui travaille sur ce projet est composée de :

— Rémi Oudin
— Jules Kozolinsky
— Baptiste Pauget
— Marc Coudriau
— Ismail Lahkim Bennani
— Clément Pascutto
— Paul Jeanmaire

1 Description du projet

L'idée principale est de faciliter la vie des MJs et de leur fournir une plateforme virtuelle sur laquelle ils ont un contrôle presque total. Le but est de fournir un outil qui va permettre de créer un jeu facilement et avoir le contrôle total sur ce jeu pendant la partie.

Les jeux créés sont des tours par tours comme dans un JdR papier classique pour les combats, et des "temps réel" hors combat. Le rôle du MJ est de créer les sprites de son jeu, les quêtes, les personnages joueurs et non joueurs, les sorts qui vont avec etc .. Notre rôle est de lui donner plein d'outils pour ça et un environnement ergonomique et fonctionnel de travail.

Notre principal objectif est de créer l'environnement le plus généraliste possible pour permettre à un MJ, même novice en programmation d'exprimer toute sa créativité sans contraintes. Cependant, nous devons faire certains choix de design pour que le projet reste utile (c'est facile de permettre au MJ de faire absolument tout ce qu'il désire, il suffit de le laisser coder son jeu lui-même) et réalisable dans le temps imparti.

2 Choix de design restrictifs

Un jeu aurait deux "environnements" principaux :

— Le coté RP : les personnages ont accés à un plateau de jeu sur lequel ils peuvent se déplacer, intéragir avec certains éléments interactifs du décor (par exemple en cliquant dessus, ou en utilisant un objet dessus) et avec les autres personnages (joueurs ou non) présents.

— Le coté combats : un combat se déroule en tour par tour, le joueur a accès à un certain nombre d'actions par tours (cf. Dungeon & Dragons par exemple, ou Dofus). La zone accessible durant un combat est délimitée, un combat dure jusqu'à ce qu'il n'y ait plus de personnages en état dans un des deux camps, que tous les personnages joueurs (PJ) abandonnent le combat, ou que le MJ décide d'y mettre fin.

Les mécaniques d'effets possibles sont fixées (par exemple : effet dégats, effet modification de caractéristique) et les moment ou ces effets sont déclenchés aussi (par exemple : au moment de l'utilisation, au début du n-ième tour suivant l'utilisation, à la fin du n-ième tour d'utilisation, à l'intéraction avec une case ou un joueur donné etc..). Les formules utilisées par ces sorts peuvent prendre en compte des caractéristiques du lanceur et de la cible uniquement (par exemple : fait des dégats proportionnels aux points de vie restants de la cible).

Le jeu se joue principalement à la souris, le clavier est utilisé pour envoyer des messages ou pour les raccourcis configurés par le MJ (par exemple des raccourcis de sorts ou le très controversé mais néanmoins utile Alt+F4)

Il n'y a qu'un seul MJ par partie. Le client aura un mode MJ qui donnera accès à une console et des commandes supplémentaires (voire une interface différente ?)

3 Autres choix et features

L'intégralité du contenu du jeu sera fournie dans des fichiers de description externes (XML ?) créés à l'aide des outils de création du MJ. Cela comprend

- les personnages (joueurs ou non) : création du skin (images sous tous les angles pour l'arrêt, petites animations d'au moins 3 images (?) pour la marche et la course), des sorts, des éventuels particularité des caractéristiques (par exemple : bonus racial) etc.. Par ailleurs, tous les skins du jeu sont disponibles en permanence : le MJ peut choisir à tout moment de se transformer (ou de transformer quelqu'un) en petit chaton s'il a au préalable définit le skin d'un petit chaton sur le serveur.
- les caractéristiques : le set de caractéristiques disponibles est créé par le MJ, elles peuvent être utilisées dans n'importe quelle formule utile au jeu (comme les formules de calcul des dégats), leur modification (manuelle ou automatique) génère un événement qui est utilisable par le MJ pour déclencher autre chose. Par exemple, lorsque la caractéristique "Points d'expérience" atteint 100%, la caractéristique "Niveau" augmente de 1, et si la caractéristique "Points de vie" est définie par "4*Niveau+2*Constitution", elle est mise à jour à ce moment là.
- les objets : que ce soit les équipements, les ressources ou les objets interactifs (potions, clefs etc..). Certains objets sont ciblables et déclenchent un événement sur leur cible (par exemple, tirer à l'arc sur quelqu'un/quelque chose hors combat. En combat l'utilisation d'une arme est considérée comme un sort). Ces événements peuvent à leur tour en déclencher d'autres (par exemple : entrer automatiquement en combat si on tir sur un ennemi).
- les outils : fenêtre d'inventaire, fenêtre de caractéristique, fenêtre de quête, fenêtre pour consulter son élevage de murlocs etc.. Ces outils pourront eux-même s'appuyer sur des fichiers externes (par exemple : on peut créer une fenêtre de quête qui trouvera les descriptions nécessaires dans des fichiers externes).
- les cartes : les cases sur la carte ont 3 états de base
  - libre : on peut marcher dessus
  - pleine sans gêner la ligne de vue : on ne peut pas marcher dessus, mais elle ne réduit pas notre champs de vision (exemple : trou)
  - pleine qui gêne la ligne de vue : on ne peut pas marcher dessus et on ne peut pas voir à travers (exemple : mur)

  il est possible d'ajouter des états aux cases et de les utiliser de manière interactive (par exemple : portail, glyphe, etc..).

Les ennemis des joueurs seront tous contrôlés par le MJ, les éventuels invocations non-statiques sur le terrain seront contrôlées par l'invocateur.

4 Travail a réaliser

Trouver un nom au projet

Description des communications client-server

Description des interfaces client-content (ie format de description des objets, des maps, des classes etc..)

Description d'un langage de script pour créer des outils de la barre d'outils

Création du serveur (A COMPLETER)

Création du client desktop (A COMPLETER)

Création d'un système de barre d'outil

Création des différents outils de création de contenu

- création de skins
- création de personnages

— création de sorts
— création de maps
— création d'outils
— outils génériques (fenêtre de caractéristiques, inventaire, fenêtre de sorts etc..).

Création d'un launcher

(Création d'un système de VoIP)

Quelques détails techniques

Le client ne fait aucune action, il se contente d'afficher ce que le serveur lui demande d'afficher, et il décrit ce que l'utilisateur aimerait faire. Il attend systématiquement (a modifier si lag important) un ordre du serveur pour s'exécuter. Il sera écrit en python

Le jeu est prévu pour fonctionner en LAN, il est possible d'y jouer sur l'Internet en simulant un réseau LAN (par exemple avec Hamachi)

Le serveur sera écrit en C++, le client exécutera un code C dans un thread séparé pour communiquer avec le serveur (on évite ainsi les problèmes de compatibilité C-Python éventuels, et on est plus rapide !).