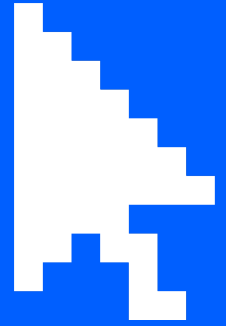


UniQuest

Le RPG où chaque ligne de code est un pas vers l'épopée.



Introduction du sujet

Les **jeux de rôle (RPG)** sont apparus dans les **années 70**, fortement influencés par les univers de Donjons & Dragons et les récits de fantasy. En 1981, un certain **Richard Garriott** lança **Ultima**, un des tout premiers RPG vidéoludiques, programmé en BASIC sur Apple II. Ce jeu, développé dans la chambre d'un étudiant, allait poser les bases d'un genre aujourd'hui incontournable. **Merci Richard**, tu as codé nos rêves !

Depuis, les RPG ont évolué : du texte à la 3D, des dés aux arbres de compétences, ils ont su captiver des générations de joueurs et de développeurs. Avec **Unity** et **C#**, créer un RPG est aujourd'hui à la portée des étudiants... comme nous !

UniQuest, c'est notre aventure. Un projet scolaire devenu épopée numérique, où l'on **code des combats**, **forge des quêtes**, et **apprend à chaque ligne**. **Spoiler alert** : il n'y a pas de boss final, mais il y a **une soutenance**.



Contexte

Comme vous l'avez sûrement compris, l'objectif de ce sujet sera de réaliser un jeu de type **RPG tour par tour** à l'aide de **C#**.



Il devra y avoir plusieurs éléments essentiels au jeu :

- **Une map** ou le joueur pourra se déplacer à l'aide des touches fléchées du clavier comme un véritable explorateur.
- Sur cette carte, des **combats** peuvent survenir en aléatoire durant les déplacements.
- **Un menu** dans lequel il doit être possible de consulter l'ensemble de son équipe (vie, statistique, attaques possibles ...), il faut bien savoir qui est le meilleur !
- **Un inventaire** avec potions, clés et objets de boost de statistiques (et peut-être quelques snacks pour la route), ainsi qu'un **menu de**



sauvegarde et **de chargement** de la partie, car perdre sa progression serait aussi tragique qu'un épisode de série dramatique.

Les combats se dérouleront en **lv1** au **tour par tour**. L'ennemi sera piloté par une **IA** qui pourra être d'une intelligence variable, allant de l'attaquant aveugle au stratège machiavélique exploitant vos moindres faiblesses. Si votre personnage tombe au combat, pas de panique, vous pourrez le remplacer par un autre membre de votre équipe. Mais attention, si tous vos personnages sont KO, c'est Game Over.

Chaque tour demandera de choisir un mouvement précis parmi une sélection (attaque, magie, objet), et la magie consommera une barre de mana qui se rechargera progressivement – **patience, jeune mage !**

Chaque personnage aura son propre type, des statistiques variées comme l'attaque, la défense, la vitesse, les PV, les PM et la précision. En combattant, ils gagneront de **l'expérience** pour monter de niveau, améliorant leurs stats et débloquent de nouvelles attaques.

Les attaques auront des types, des stats d'attaque et de précision, ainsi que des chances de coup critique. La victoire octroiera de l'expérience, récompensant le travail et la stratégie.

Le jeu se déroulera **en deux phases** : d'abord sur la carte, où notre personnage pourra se déplacer, parler à des personnages, ouvrir des coffres et enclencher des combats avec des ennemis ; puis en phase de combat,



exploitant toutes les mécaniques d'affrontement détaillées précédemment.

Préparez-vous à des duels épiques !

Il est crucial de planifier la structure du jeu et de ses composants dès le départ. Cela inclut la création de diagrammes de classes pour une vision claire du projet, avec l'utilisation de l'héritage pour les personnages, les attaques, les objets, etc.

Étant donné que les combats reposent largement sur des calculs mathématiques, nous mettrons en place des **tests unitaires** pour assurer la cohérence et la fiabilité du système.

Compétences visées

- Installer et configurer son environnement de travail en fonction du projet.
- Développer des composants métier
- Contribuer à la gestion d'un projet informatique
- Analyser les besoins et maquetter une application
- Définir l'architecture logicielle d'une application
- Préparer et exécuter les plans de tests d'une application

Rendu

Votre travail est évalué en présentation avec un support et une revue de code. Le slide doit être composé de :



- De l'organisation de votre équipe
- De vos problèmes rencontrés ainsi que les solutions apportées
- La démonstration jouable de votre jeu

Le projet est à rendre sur <https://github.com/prenom-nom/uniquest>

Base de connaissances

- [C# Documentation](#)
- [Documentation MSDN Console](#)
- [POO](#)
- [Tests Unitaires](#)