

# The Binding Of Briatte

Projet C, 3IBC



```
    false;  
    (cin, sInput);  
    m("cls");  
    stringstream(sInput) >> dblTemp;  
    length = sInput.length();  
    (iLength < 4) {  
        again = true;  
        continue;  
    } else if (sInput[iLength - 3] != '.') {  
        again = true;  
        continue;  
    } while (++iN < iLength) {  
        if (isdigit(sInput[iN])) {  
            continue;  
        } else if (iN == (iLength - 3) ) {  
            continue;  
        }
```

Présenté par

Arthur Polit

Hanane Temal

# Sommaire

I – Présentation du projet

II – Analyse de l'application et installation


III – Bilan



# I – Présentation du projet

Notre objectif était de réaliser un roguelike en C nommé The Binding Of Briatte. Le personnage devra réussir à s'échapper d'un donjon. Au cours d'une run, le joueur récupérera des objets qui amélioreront les caractéristiques du personnage et permettront de continuer à descendre les étages du donjon en battant les boss. Runs après runs, de nouveaux étages, objets, personnages et boss apparaîtront, avant d'arriver à la confrontation avec le boss final : Athina l'Akita.

Les participants au projet sont Hanane Temal, Arthur Polit et Najd Kacem, malheureusement Najd a décidé d'arrêter la formation.



# II – Analyse de l'application et installation

L'ensemble des structures de données est le suivant :

- Room
- RoomList
- Item
- ItemList
- Monster
- MonsterList
- Level
- Map
- Player
- PlayerList

Les fonctions principales pour les items sont `writeItemFile()`, `addItemFile()`, `removeItemFile()` et `readItemFile()`. Pour la création et la gestion des monstres les fonctions `writeMonsterFile()`, `readMonsterFile()`, `addMonsterFile()`, ainsi que `removeMonsterFile()` ont été utilisés.

Pour la génération et la gestion de pièces nous avons crée les fonctions `writeRoomFile()`, `readRoomFile()`, `readRoom()`, `addRoomFile()`, `removeRoomFileById()`, `removeRoomFileByIndex()` et `showRoomFileById()`.

Pour le joueur, il y a `newPlayer()`, `freePlayer()`, `read player()`, `printPlayer()`, `showPlayer()` et `selectPlayer()`.

Et enfin pour gérer la souvegarde nous avons crée les fonctions `readPlayerFile()`, `writePlayerFile()` et `addPlayerFile()`.

# III - Bilan

Ce projet à été très compliqué à prendre en main, étant donné que nous avons perdu un membre, gérer ce projet à 2 a été très dur. L'idée de faire un jeu comme projet nous a plu mais ça a été un vrai challenge intéressant. Etant novice, les parties effectuées à savoir de I au début de la partie V ont été longues à réaliser mais pédagogiques. Ce projet nous a aidé à modéliser et à résoudre la problématique en produisant des petits modèles, c'est-à-dire des structures de données.

