

The Binding Of Briatte

Projet C, 3IBC



```
        false;  
        cin >> sInput;  
        system("cls");  
        stringstream(sInput) >> dblTemp;  
        iLength = sInput.length();  
        while (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

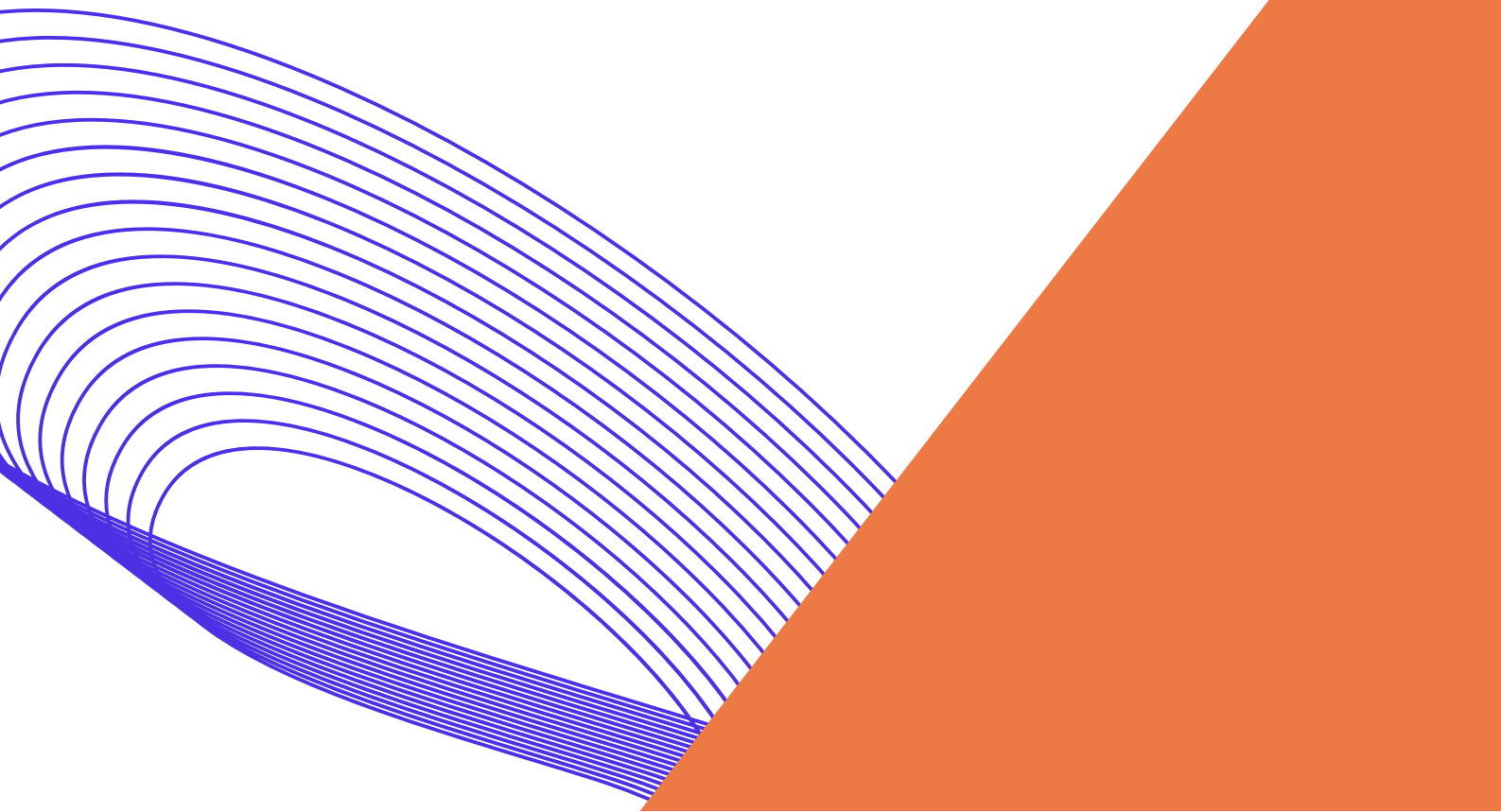
Hanane Temal

Sommaire

I – Présentation du projet

II – Analyse de l'application

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.

A l'origine nous étions trois à participer à ce projet, malheureusement Najd a décidé d'arrêter la formation.

Participants au projet :

- Najd
- Arthur
- Hanane

II – Analyse de l'application

Ensemble des structures de données :

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

Fonctions principales :

Pour les items:

```
void writeItemFile(ItemList itemList, char *itemfile);  
void addItemFile(Item *item, char *itemfile);  
void removeItemFile(Item *item, char *itemfile);  
ItemList * readItemFile(char *itemfile);
```

Pour les monstres :

```
void writeMonsterFile(MonsterList monsterList, char *monsterfile);  
MonsterList * readMonsterFile(char *monsterfile);  
int addMonsterFile(Monster *monster, char *monsterfile);  
void removeMonsterFile(Monster *monster, char *monsterfile);
```

Pour les pièces :

```
void writeRoomFile(RoomList roomList, char *fichier);  
RoomList * readRoomFile(char *roomfile, char *monsterfile);  
Room * readRoom(FILE *f, char *monsterfile);  
void addRoomFile(Room room, char *fichier);  
void removeRoomFileById(int id, char *fichier, char *monsterfile);  
void removeRoomFileByIndex(int index, char *fichier, char *  
monsterfile);  
void showRoomFileById(int id, char *fichier, char *monsterfile);
```

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. N'étant pas non plus extrêmement à l'aise avec le C nous avons choisi de nous concentrer sur la création des CRUD ainsi que leurs bon fonctionnement.

L'idée de faire un jeu comme projet m'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 a résoudre la problématique en produisant des petits modèles, c'est-à-dire des structures de données.

