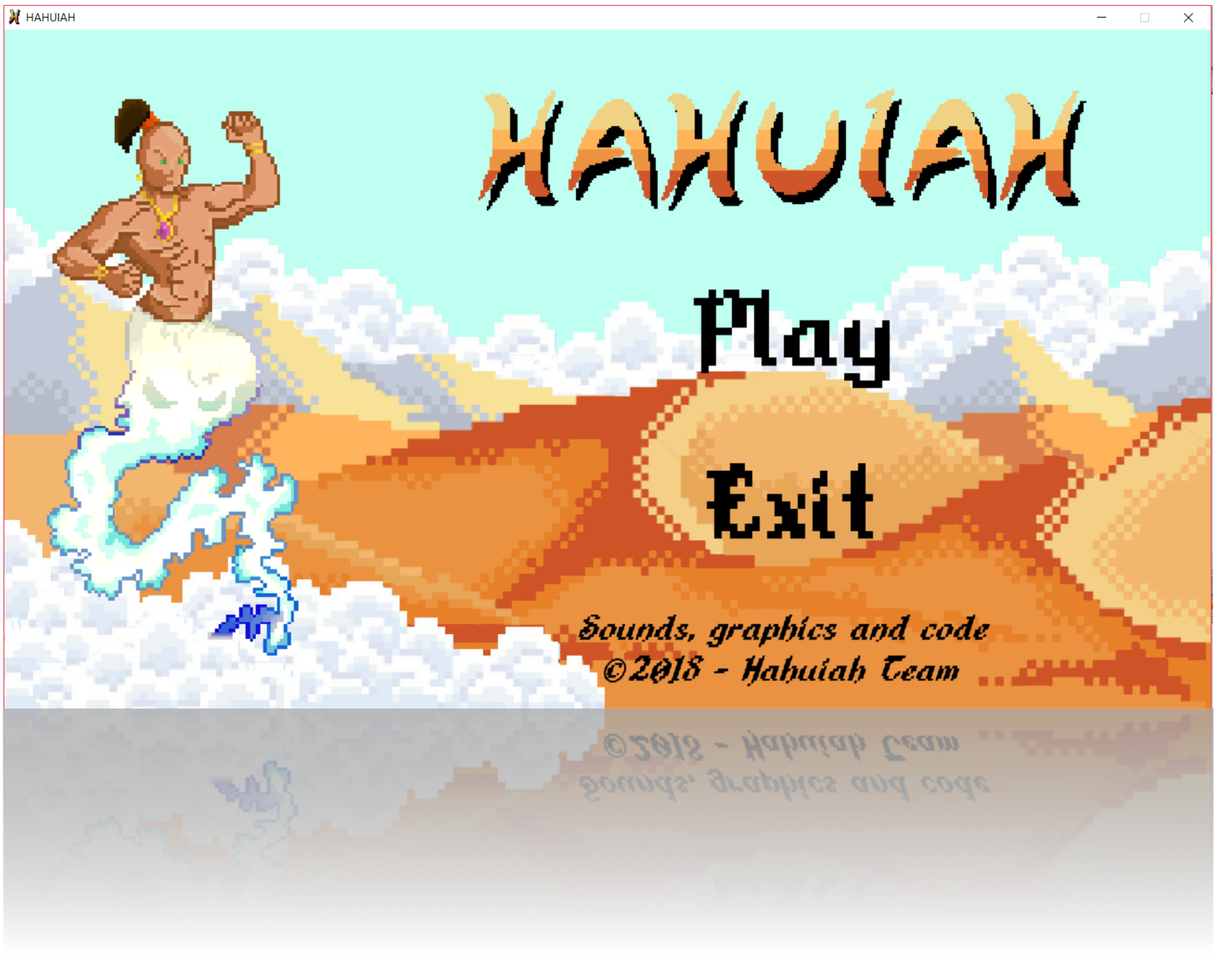
# Project Report

# HAHUIAH



BEAUFILS Vincent – CHAMPSAUR Lucas – CRETEUR Samuel – DUCHEMIN Stéphane – PECH Lucas
L2 – Internationale – Promotion 2022

# Summary

## Introduction

First of all, we will present our project in C, what it is and how we proceeded.
Secondly, we will look at the biggest difficulties encountered in creating the software, the moments where we blocked, and we need to go other ways.
Then, we will analyze our project, the important lines of code, why did we decide to do this or that.
Finally, we will see the different algorithms used, the resources we used and the different functions of the software.
And we will conclude on what the project has brought us.

## Presentation of the Project

The project that we realized is a game which has the same principle that the famous *Akinator*. To simplify the different tasks and to be able to achieve it in time, we decided to limit the number of persons to the numbers of students in the L2 International Promotion 2022 of EFREI Paris.
The rules of the games are very simple. The genius *Hahuiah* will ask you some simple questions and guess the person you're thinking about.
Now, how this project answers to the main problematic which is code something in relation with tree?
Well you can guess it, each decisions you'll make has its impact for the final screen.

## Main Difficulties

The hardest part was in the beginning. We had to organize the tasks we had to do. The first tasks were very long, indeed, we had to collect information on all the promotion.
We had to think of the method to create our tree and sort all the information.
Once we've done it, thanks to the organization of the data the coding part wasn't very long. The design part on the other hand progressed all along the project until a few days before the defense.

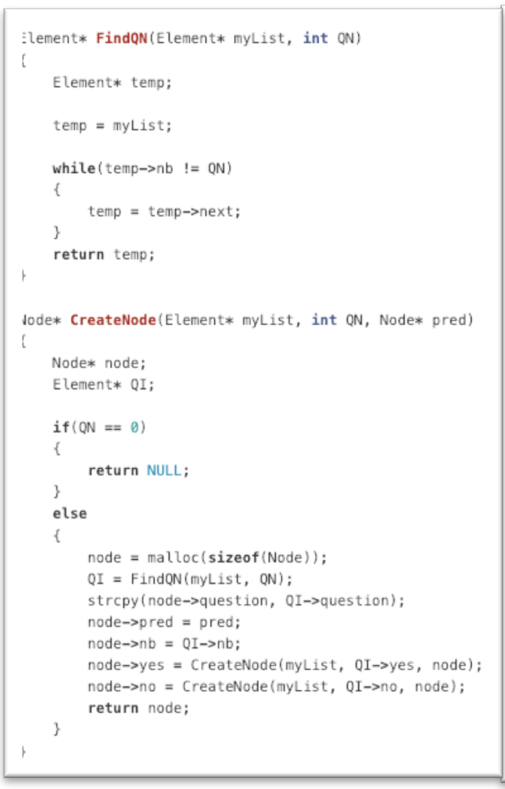
## Analysis of the Project

Our project is using a GUI, *SDL* the most famous one in C programming with the help of *SDLMixer* to be able to play sound, music in harmony with all the designs.

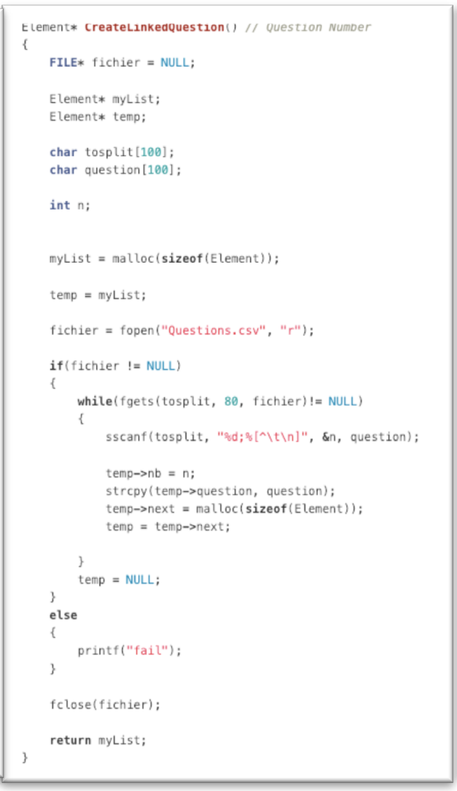
If we put the SDL aside and we concentrate on the fundamental algorithm which manage the tree, and the program, all that stuff works thanks to two files: The "Questions.csv" file and the "Liens.csv". We have 4 functions which manage the tree : `CreateLinkedQuestion(), AddYesNoLinked(), FindQN()` and `CreateNode().`

In the first place we're creating a linked list with the excel files (.csv) with the `CreateLinkedQuestion()`. With the linked list we put the question in the tree with `CreateNode()` and the link to the next question if we click on *YES* or *NO* managed with the `AddYesNoLinked()`. The `FindQN()` will find the next question in the linked list and import it in the tree and so on.

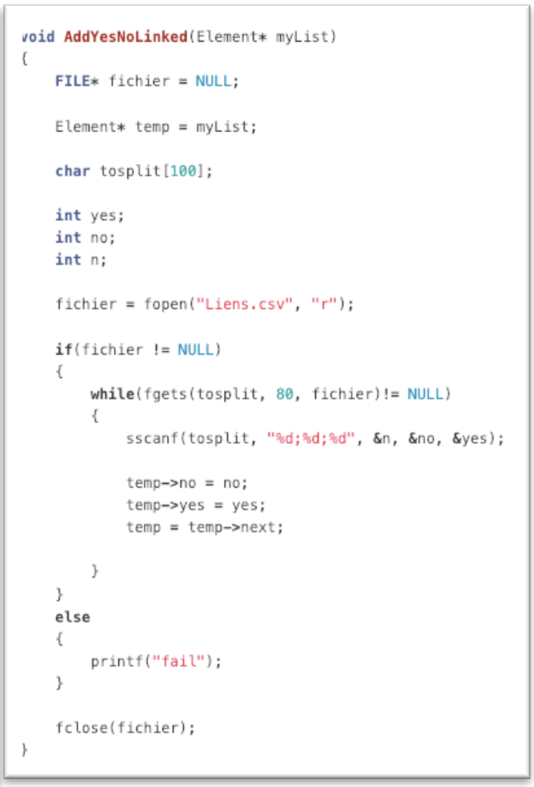
## Some Algorithms



```c
lement* FindQN(Element* myList, int QN)
{
    Element* temp;

    temp = myList;

    while(temp->nb != QN)
    {
        temp = temp->next;
    }
    return temp;
}

ode* CreateNode(Element* myList, int QN, Node* pred)
{
    Node* node;
    Element* QI;

    if(QN == 0)
    {
        return NULL;
    }
    else
    {
        node = malloc(sizeof(Node));
        QI = FindQN(myList, QN);
        strcpy(node->question, QI->question);
        node->pred = pred;
        node->nb = QI->nb;
        node->yes = CreateNode(myList, QI->yes, node);
        node->no = CreateNode(myList, QI->no, node);
        return node;
    }
}
```



```c
Element* CreateLinkedQuestion() // Question Number
{
    FILE* fichier = NULL;

    Element* myList;
    Element* temp;

    char tosplit[100];
    char question[100];

    int n;


    myList = malloc(sizeof(Element));

    temp = myList;

    fichier = fopen("Questions.csv", "r");

    if(fichier != NULL)
    {
        while(fgets(tosplit, 80, fichier)!= NULL)
        {
            sscanf(tosplit, "%d;%[^\t\n]", &n, question);

            temp->nb = n;
            strcpy(temp->question, question);
            temp->next = malloc(sizeof(Element));
            temp = temp->next;

        }
        temp = NULL;
    }
    else
    {
        printf("fail");
    }

    fclose(fichier);

    return myList;
}
```



```c
void AddYesNoLinked(Element* myList)
{
    FILE* fichier = NULL;

    Element* temp = myList;

    char tosplit[100];

    int yes;
    int no;
    int n;

    fichier = fopen("Liens.csv", "r");

    if(fichier != NULL)
    {
        while(fgets(tosplit, 80, fichier)!= NULL)
        {
            sscanf(tosplit, "%d;%d;%d", &n, &no, &yes);

            temp->no = no;
            temp->yes = yes;
            temp = temp->next;

        }
    }
    else
    {
        printf("fail");
    }

    fclose(fichier);
}
```

## Conclusion

This project allowed us to work in groups but also to organize us to separate the tasks to be done. This project allowed us to meet many times to discuss and debate how we were going to code some functionalities, help each other on the different points where we disagreed or when we were blocked. Apart from the group work, this project allowed us to confirm our knowledges about the C language and explore the possibilities that the trees open to us and the management of many data. We all loved to work on this project.