

Mohamed KANOUTE
Yacine MAHMOUDI

COMPTE RENDU JEU JAVA

SOMMAIRE

- 1. OBJECTIFS ATTENDUS.....**
 - a. le nombre mystérieux.....**
 - b. les consignes.....**
 - c. division du travail.....**
- 2. ALGORITHME.....**
- 3. CODE JAVA.....**
- 4. RESSOURCES UTILISÉES.....**
- 5. DIFFICULTÉS RENCONTRÉES.....**

OBJECTIFS ATTENDUS

1) le nombre mystérieux

Le devoir de programmation est un jeu à 2 joueurs. Ils doivent chacun entrer leurs noms.

Ensuite le joueur 1 doit choisir un nombre entre 1 et 100.

Le joueur 2 doit ensuite trouver ce fameux nombre secret en proposant des nombres.

Le joueur 1 ne peut répondre que par “trop haut” ou “trop bas” ou, évidemment, “Gagné”.

Le joueur 2 dispose lui de 10 tours pour trouver le fameux nombre secret.

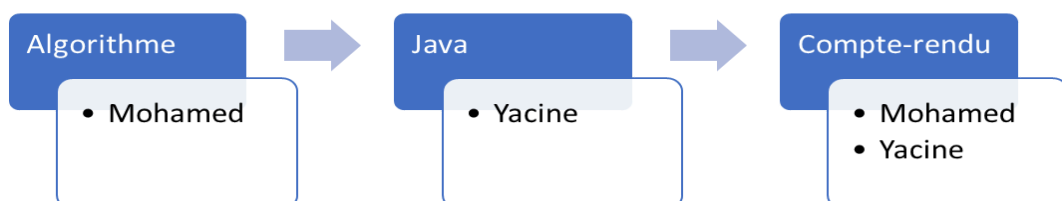
En cas de victoire du deuxième joueur, on affiche le nombre de coups ayant été nécessaires pour trouver le nombre mystère.

2) les consignes

Les consignes sont les suivantes:

- Insister sur les contrôles tout au long du jeu
- Porter un soin particulier au dialogue et à la présentation des différents écrans du jeu
- Le programme Java sera commenté
- Les variables répondent aux règles de nommage courantes

3) Divisions du travail



Algorithme

ALGORITHME NombreMystere

variable:

numérique:; nb, tourmax, tour, stop,

caractères : J1, J2, Indication

DÉBUT

// Affichage des règles du jeu

Afficher "Bienvenue au jeu du Nombre Mystérieux !"

AFFICHER "Un joueur choisit un nombre entre 1 et 100, l'autre joueur devine ce nombre en 10 essais."

// Saisie des noms des joueurs

AFFICHER "Joueur 1, choisissez un prénom : "

Saisir J1

AFFICHER "Joueur 2, choisissez un prénom : "

Saisir J2

// Choix du nombre mystère par J1

AFFICHER J1 + ", choisissez un nombre secret entre 1 et 100 : "

SAISIR nombre secret

TANT que nb <1 ou nb>100

Afficher "Le nombre donné n'est pas valide, veuillez donner un nombre compris entre 1 et 100 inclus : "

SAISIR nombre secret

Fin tant que

// Début du jeu

TOUR <- 0

STOP <- 0

TOURMAX <-10

INDICATION <- ""

TANT QUE TOUR <= TOURMAX ET STOP = 0

AFFICHER J2 + ", veuillez donner un nombre: "

SAISIR nbj2

AFFICHER J1 " PLUS BAS OU PLUS HAUT?"

TANT QUE INDICATION == ""

Afficher INDICATION à la prochaine ligne

AFFICHER INDICATION

FIN TANT QUE

SI nbj2<1 ou nbj2 >100

AFFICHER "Le nombre donné n'est pas valide, veuillez donner un nombre compris entre 1 et 100 inclus : "

SAISIR nbj2

FINSI

TOUR <- TOUR+1

AFFICHER TOUR

SI nbJ2= nb

STOP <- 1

AFFICHER STOP

FIN SI

SINON SI nbj2=nb

AFFICHER "BIEN JOUE" +J2

FIN SI

SINON SI tour =tourmax

AFFICHER "GAME OVER,le nombre secret etait nbj2."

FIN SI

Explication du choix de cette algorithmme:

Ceci est l'algorithme que nous avons construit afin de mettre en place le jeu. Il comprend des boucles "tant que" ainsi que des structures conditionnelles "SI". Elle respecte aussi toutes les consignes du jeu, permettant aux joueurs de jouer en respectant les règles de programmations. Il n'a servi que de base car le code Java final est beaucoup plus évolué et n'a plus rien à voir avec celui-ci.

Affichage des règles du jeu : Le programme débute en affichant les règles du jeu, expliquant que l'un des joueurs choisit un nombre entre 1 et 100, et l'autre joueur doit le deviner en 10 essais.

Saisie des noms des joueurs : Les joueurs saisissent leur prénom.

- Variables : J1, J2

Choix du nombre mystère par le premier joueur : Le premier joueur choisit un nombre secret entre 1 et 100, avec une vérification de la saisie pour s'assurer que le nombre est dans la plage autorisée.

- Variables : nb
- Condition TANT QUE : TANT QUE $nb < 1$ OU $nb > 100$

Déroulement du jeu : Le deuxième joueur doit deviner le nombre secret en faisant des essais. Après chaque essai, le programme affiche le nombre d'essais restants.

- Variables : TOUR, TOURMAX, STOP, INDICATION, nbJ2
- Condition TANT QUE : TANT QUE $TOUR \leq TOURMAX$ ET $STOP = 0$

Affichage des indications : Après chaque essai du deuxième joueur, le premier joueur donne une indication en disant si le nombre à trouver est plus bas ou plus haut. Le programme s'assure que le premier joueur donne des indications valides en utilisant une boucle TANT QUE.

- Conditions TANT QUE : TANT QUE INDICATION == ""
- Fin du jeu : Le jeu se termine lorsque le deuxième joueur devine correctement le nombre secret, atteint le nombre maximum d'essais (10 dans ce cas), ou lorsque le joueur 1 entre une indication incorrecte.
- Conditions SI : Plusieurs conditions, par exemple, SI nbJ2 = nb, SINON SI TOUR = TOURMAX

On peut penser à améliorer également, pour éviter des erreurs consoles lorsqu'on saisit une chaîne de caractère ou un caractère au lieu d'un entier. Par exemple, dans mon code java ici je demande des nombres qui sont initialisés dans des entiers ce qui provoque une erreur console si jamais on écrit une chaîne de caractère ou un caractère. Pour régler ceci, on peut penser à rajouter un contrôle de saisie ou alors utiliser des variables de type String.

CODE JAVA

```
package Projet;
import java.util.Scanner;
public class Programme {

    public static void main(String[] args) {

        while(true) {

            Scanner input = new Scanner(System.in);

            //VARIABLE:
            //Nom du jeu :
            //Prénoms des joueurs
            String J1,J2;
            //Nombre secret
            int nb;
            //Partie
            int tour = 0;
            int tourmax = 10;
            int tourrestant = 10;
            int nbJ2 = 1;
            int stop = 0;
            String indication = "";

            //Nom du jeu
            String phrase = ("\u001B[31mL\u001B[0m"
                + "\u001B[32mE\u001B[0m"
                + "\u001B[33m \u001B[0m"
                + "\u001B[34mN\u001B[0m"
                + "\u001B[35mO\u001B[0m"
                + "\u001B[36mM\u001B[0m"
                + "\u001B[37mE\u001B[0m"
                + "\u001B[31mR\u001B[0m"
                + "\u001B[32mE\u001B[0m"
                + "\u001B[33m \u001B[0m"
                + "\u001B[34mM\u001B[0m"
                + "\u001B[35mY\u001B[0m"
                + "\u001B[36mS\u001B[0m"
                + "\u001B[37mT\u001B[0m"
                + "\u001B[31mÉ\u001B[0m"
                + "\u001B[32mR\u001B[0m"
                + "\u001B[33mI\u001B[0m"
                + "\u001B[34mE\u001B[0m"
                + "\u001B[35mU\u001B[0m"
                + "\u001B[36mX\u001B[0m");
```

```

System.out.println("\u001B[31m-----\u001B[0m");
                System.out.println("\u001B[31m- " + phrase +
"\u001B[31m -\u001B[0m");

System.out.println("\u001B[31m-----\u001B[0m");
                //Explication du jeu
                System.out.println();

System.out.println("\u001B[35m-----
-----\u001B[0m");

                System.out.println();
                System.out.println("\u001B[32mEXPLICATION DU JEU");
                System.out.println();
                System.out.println("Bienvenue, ce jeu se joue à 2. Le principe
est simple, le premier joueur doit choisir un nombre de 1 à 100. ");
                System.out.println("Le deuxième joueur lui doit deviner ce
nombre en 10 essais.\u001B[0m");
                System.out.println();

System.out.println("\u001B[35m-----
-----\u001B[0m");

                System.out.println();
                //Prénoms des joueurs
                System.out.println("\u001B[36mJoueur 1\u001B[0m, veuillez
choisir un prénom : ");
                J1 = input.nextLine();
                System.out.println("\u001B[31mJoueur 2\u001B[0m, veuillez vous
aussi choisir un prénom : ");
                J2 = input.nextLine();

                //Initialisation du nombre secret et vérification de la saisie
                System.out.println("\u001B[36m" + J1 + " \u001B[0m" +
"veuillez choisir un nombre secret de 1 à 100");
                nb = input.nextInt();
                while((nb < 1) || (nb > 100)) {
                        System.out.println("Le nombre donné n'est pas valide,
veuillez donner un nombre compris entre 1 et 100 inclus : ");
                        nb = input.nextInt();
                }

                //Partie
                while((tour != tourmax) && (stop != 1)){
                        System.out.println("\u001B[31m" + J2+ "\u001B[0m" + "
veuillez donner un nombre : ");
                        nbJ2 = input.nextInt();

```

```

        // Vérifier si le nombre est valide
        while((nbJ2 < 1) || (nbJ2 > 100)) {
            System.out.println("Le nombre donné n'est pas
valide, veuillez donner un nombre compris entre 1 et 100 inclus : ");
            nbJ2 = input.nextInt();
        }

        //affiche le nombre de tour restant
        if(nbJ2 != nb) {
            tourrestant = tourrestant - 1;
            System.out.println("Nombre de tour restant : " +
tourrestant);
        }
        // Ajouter une condition pour vérifier si le nombre a été
trouvé avant de demander l'indication
        if (nbJ2 == nb) {
            stop = 1;
        } else if (tour < tourmax - 1) {
            System.out.println("\u001B[36m" + J1 + " \u001B[0m" +
"Trop bas ou Trop haut ?");
            input.nextLine(); // Pour éviter d'entrer dans la
boucle sans avoir initialisé l'indication
            indication = input.nextLine();
        }

        // Obliger le joueur 1 à donner de bonnes indications
avec un contrôle de saisie
        while(!indication.equalsIgnoreCase("Trop bas") &&
!indication.equalsIgnoreCase("Trop haut") && stop != 1 ) {
            System.out.println("Recommencez l'indication
n'est pas bonne");
            indication = input.nextLine();
        }
        tour = tour + 1;

        }if(nbJ2 == nb) {
            System.out.println("\u001B[32m" + "Bien joué !" + " Score = " +
tour + " essai" + "\u001B[0m"); System.out.println(); System.out.println();
System.out.println();
        }else if(tour == tourmax) {
            System.out.println("\u001B[31m" + "Perdu" + "\u001B[0m");
System.out.println(); System.out.println(); System.out.println();
        }
    }
}
}

```


Explication Java:

Dans le programme java, nous avons ajouté d'autres fonctionnalités notamment le rajout de contrôle de saisie du premier joueur qui doit donner des indications «Trop Haut», «Trop Bas». Avec une boucle tant que, les chaînes de caractères dans la variable indication (tant qu'il) sont différentes de «Trop Haut», «Trop bas» on reste dans la boucle. La variable stop permet d'éviter l'exécution de la boucle après avoir donné le bon nombre. Il y a également le rajout condition `if(nbj2 != nb)` permettant de donner le nombre d'essai restant pour le joueur 2.

Pour l'esthétique, nous avons ajouté de la couleur à du texte grâce au code ANSI.

Affichage des règles du jeu : Lorsque le programme démarre, il affiche les règles du jeu, expliquant que l'un des joueurs choisit un nombre entre 1 et 100, et l'autre joueur doit le deviner en 10 essais.

Saisie des noms des joueurs : Le programme demande aux joueurs de saisir leur prénom.

- Variables : J1, J2

Choix du nombre secret par le premier joueur : Le premier joueur choisit un nombre secret entre 1 et 100, avec une vérification de la saisie pour s'assurer que le nombre est dans la plage autorisée.

- Variables : nb
- Condition while : `while (nb < 1 || nb > 100)`

Déroulement du jeu : Le deuxième joueur doit deviner le nombre secret en faisant des essais. Après chaque essai, le programme affiche le nombre d'essais restants.

- Variables : tour, tourmax, tourrestant, nbJ2, stop
- Condition while : `while (tour != tourmax && stop != 1)`

Affichage des indications :

Après chaque essai du deuxième joueur, le premier joueur donne une indication en disant si le nombre à trouver est plus bas ou plus haut. Le programme s'assure que le deuxième joueur donne des indications valides en utilisant une boucle de contrôle de saisie.

- Variables : indication
- Condition while : while
(!indication.equalsIgnoreCase("Trop bas") &&
!indication.equalsIgnoreCase("Trop haut") && stop != 1)
(ici les méthodes equalsIgnoreCase permettent de faire
en sorte que l'utilisateur peut taper en minuscule ou en
majuscule)

Fin du jeu : Le jeu se termine lorsque le deuxième joueur devine correctement le nombre secret, atteint le nombre maximum d'essais (10 dans ce cas), ou lorsque le joueur entre une indication incorrecte.

Affichage des résultats : Le programme affiche si le deuxième joueur a deviné correctement le nombre ou s'il a épuisé tous ses essais.

- Variables : nbJ2, nb, tour, tourmax
- Conditions if : Plusieurs conditions, par exemple, if (nbJ2 == nb), else if (tour == tourmax)

Aspect visuel : Le texte du jeu est stylisé avec des couleurs en utilisant des codes ANSI, ce qui donne une présentation visuelle plus attrayante dans le terminal.

RESSOURCES UTILISÉES



ChatGPT

Google

DIFFICULTÉS RENCONTRÉES

Pour ce devoir algorithmique nous avons rencontré plusieurs difficultés. Premièrement la recherche et la mise en place de l'algorithme nous a forcé à la réflexion quant au contrôle de saisie à utiliser.

Pour ceci nous avons utilisé le support de cours de Mme ROUBEAU, ce qui nous a permis de parfaitement choisir la boucle répétitive à utiliser. Nous avons opté pour le "tant que" afin d'optimiser l'algorithme et le code.

Ensuite ce qui nous a aussi posé problème sont les problèmes de saisie(saisie, scanner), de logiciel et de boucle. En effet, l'ajout de plusieurs fonctionnalités successives dans le programme peut entraîner des problèmes, car elles étaient parfois ajoutées consécutivement, ce qui pouvait conduire à des situations où les boucles ne devaient pas être exécutées (notamment pour le code Java). Pour résoudre ces problèmes nous avons été amené à utiliser google et chatgpt afin d'apprendre ce qu'il n'y avait pas forcément dans le support de cours mais qu'on voulait tout de même apprendre pour ensuite réutiliser dans le programme.

Et enfin nous avons évidemment rencontré des problèmes d'ordre organisationnel comme tous groupes. Cependant, nous avons réussi à bien nous répartir les tâches et donc de pouvoir rendre le projet à temps et (on l'espère) de bonne qualité.

Ce projet nous aura, en somme, appris beaucoup de choses et nous sera d'une grande utilité peu importe l'option que l'on choisira à la fin du semestre !