

NGUYEN Thomas  
RUSSON Maxime  
BTS SIO SLAM

## **DM d'Algorithmique – Programmation**

### **Calculatrice en cascade et le mot à trouver**

## Sommaire :

Difficultés rencontrer et les solutions apporter

Schéma de la répartition des tâches

Outils utilisés

Première partie : Code JAVA,

Deuxième partie : Code JAVA, HTML, CSS

Conclusion

### **Difficultés rencontrer et les solutions apporter :**

Lorsque nous avons fait le programme de la deuxième partie, nous avons eu du mal à faire le modèle de la calculatrice en cascade, puis certains calculs ont été difficiles à se faire lorsque nous voulions les additionner, les multiplier, les soustraire ou les diviser.

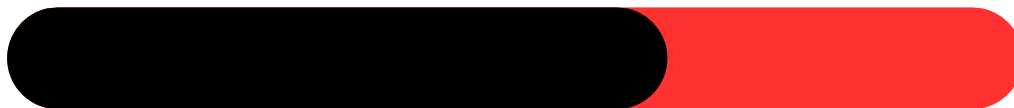
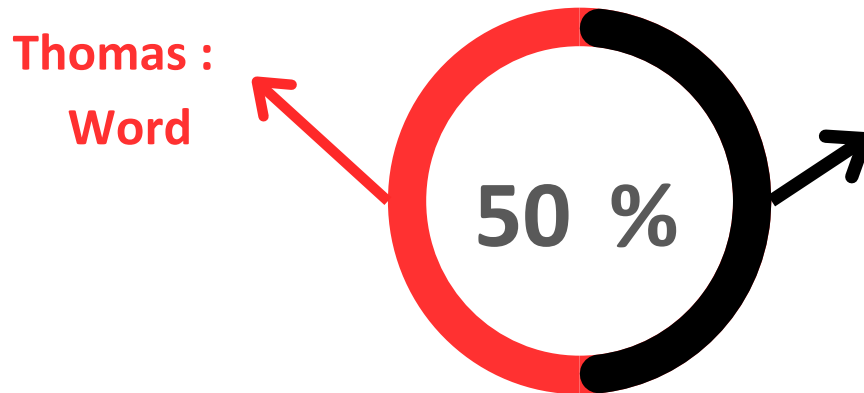
Nous avons eu du mal à faire le schéma de la répartition des tâches.

Nous avons eu du mal à traduire certaines parties de l'algorithme de la deuxième partie.

Pour corriger ces difficultés, nous avons pu trouver des solutions tels que :

...

### Schéma de la répartition des tâches :



Programme : Maxime - Thomas

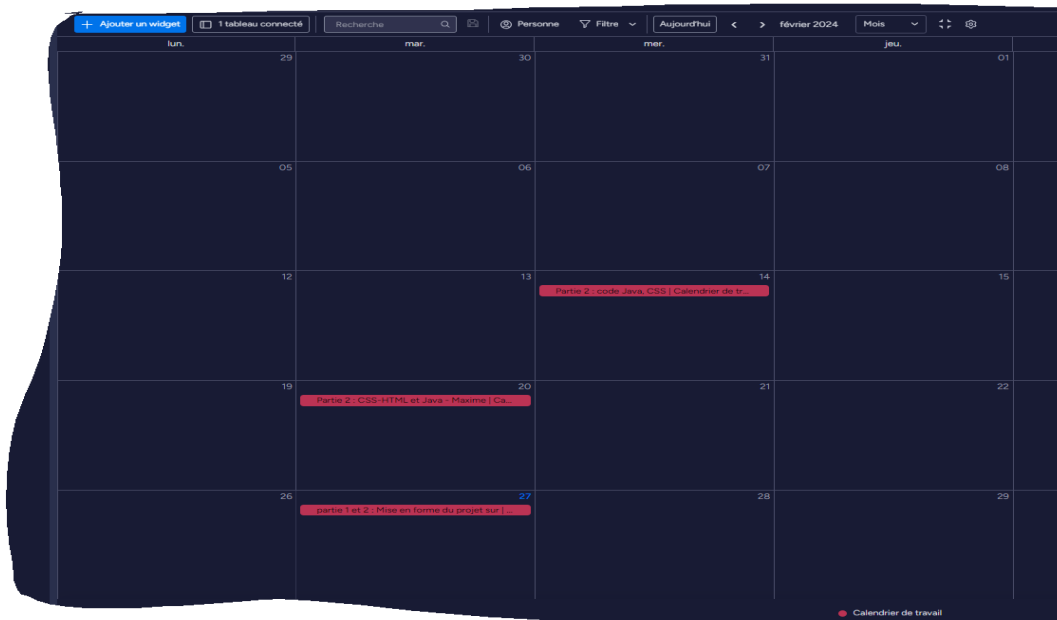
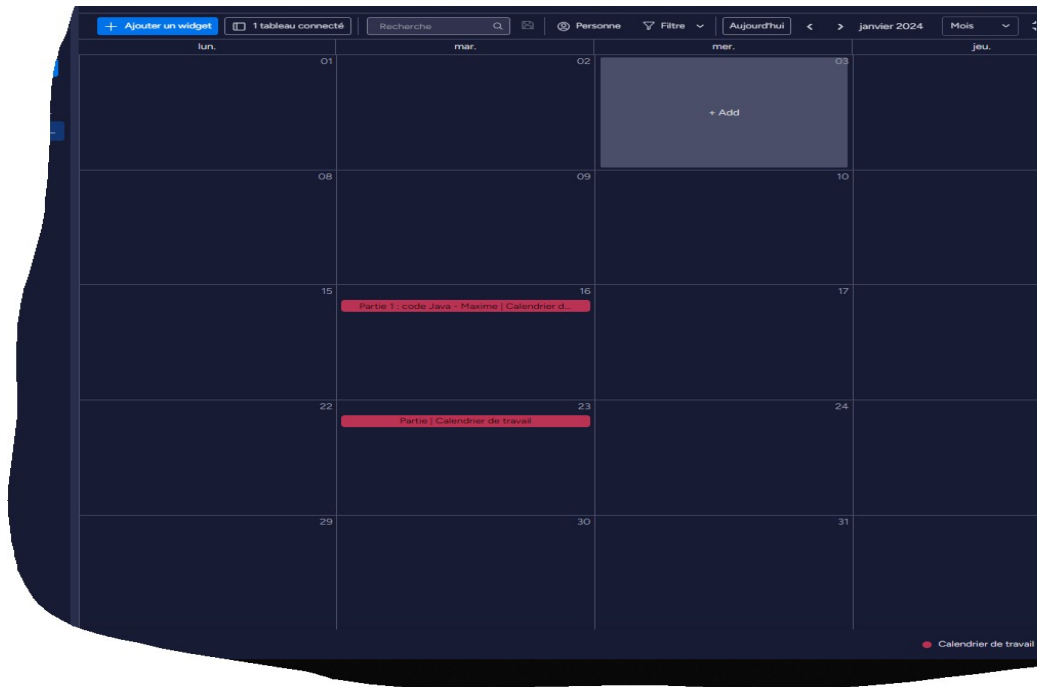


Test de la calculatrice : Maxime -  
Thomas



Mise en page sur Word : Maxime **Thomas**

Pour en savoir plus visitez [www.sitevraimentsuper.fr](http://www.sitevraimentsuper.fr)



## Première partie : code Java

```
package monpackage;
import java.util.Scanner;

public class DM {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean recommencer = true;

        while (recommencer) {
            jouerUnePartie(scanner);

            // Demander à l'utilisateur s'il veut recommencer
            recommencer = demanderRejouer(scanner);
        }

        // Fermer le scanner à la fin
        scanner.close();
    }

    private static void jouerUnePartie(Scanner scanner) {
        boolean jeuTermine = false;

        // Étape 1: Saisir le mot à deviner
        System.out.println("Saisissez un mot supérieur à 5 lettres à faire deviner :");
        String motAdeviner = scanner.nextLine().toUpperCase();

        // Contrôle de la longueur du mot
        while (motAdeviner.length() < 5) {
            System.out.println("Le mot doit avoir au moins 5 lettres. Réessayez :");
            motAdeviner = scanner.nextLine().toUpperCase();
        }

        // Générer un mot masqué pour l'afficher au joueur
        StringBuilder motMasque = new StringBuilder();
        for (int i = 0; i < motAdeviner.length(); i++) {
            motMasque.append('*');
        }

        // Nombre d'essais maximum
        int essaisMax = motAdeviner.length() - 2;
        int essais = 0;

        // Étape 2: Commencer le jeu
```

```

System.out.println("Bienvenue dans le jeu du mot à deviner !");
System.out.println("Vous avez " + essaisMax + " essais pour trouver le mot.");

while (essais < essaisMax && !jeuTermine) {
    // Étape 3: Saisir le mot du joueur
    System.out.println("Mot actuel : " + motMasque);
    System.out.println("Saisissez un mot de " + motAdeviner.length() + " lettres
    (ou tapez 'SOL' pour la solution :)");
    String motJoueur = scanner.nextLine().toUpperCase();

    // Étape 4: Vérifier la validité de l'entrée
    if (motJoueur.equals("SOL")) {
        System.out.println("La solution est : " + motAdeviner);
        return;
    } else if (motJoueur.length() != motAdeviner.length()) {
        System.out.println("Le mot doit avoir exactement " + motAdeviner.length() + "
        lettres. Réessayez.");
        continue;
    }

    // Étape 5: Comparer les lettres et les positions
    int lettresCorrectes = 0;
    StringBuilder lettresBienPlacees = new StringBuilder();

    for (int i = 0; i < motAdeviner.length(); i++) {
        if (motAdeviner.charAt(i) == motJoueur.charAt(i)) {
            lettresCorrectes++;
            lettresBienPlacees.append(motJoueur.charAt(i));
        } else {
            lettresBienPlacees.append('*');
        }
    }

    // Étape 6: Afficher le résultat
    System.out.println("Lettres correctes et bien placées : " + lettresCorrectes);
    System.out.println("Lettres bien placées : " + lettresBienPlacees);

    // Étape 7: Vérifier si le mot a été trouvé
    if (lettresCorrectes == motAdeviner.length()) {
        System.out.println("Félicitations ! Vous avez trouvé le mot : " + motAdeviner);
        jeuTermine = true;
    } else {
        essais++;
        System.out.println("Il vous reste " + (essaisMax - essais) + " essais.");
    }
}

```



```

// Étape 8: Fin du jeu
System.out.println("Merci d'avoir joué !");
}

private static boolean demanderRejouer(Scanner scanner) {
    boolean reponseValide = false;
    boolean recommencer = false;

    while (!reponseValide) {
        System.out.println("Voulez-vous rejouer ? (Oui/Non)");
        String reponse = scanner.nextLine().toUpperCase();

        if (reponse.equals("OUI")) {
            recommencer = true;
            reponseValide = true;
        } else if (reponse.equals("NON")) {
            recommencer = false;
            reponseValide = true;
        } else {
            System.out.println("Veuillez répondre avec 'Oui' ou 'Non'.");
        }
    }

    return recommencer;
}
}

```

## Deuxième partie : Code JAVA, HTML, CSS :

### Partie Html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculatrice en Cascade</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="calculator">
    <input type="text" id="display" disabled>
    <div class="buttons">
      <button onclick="clearDisplay()">C</button>
      <button onclick="appendNumber('7')">7</button>
      <button onclick="appendNumber('8')">8</button>
      <button onclick="appendNumber('9')">9</button>
      <button onclick="appendOperator('+')">+</button>
      <button onclick="appendNumber('4')">4</button>
      <button onclick="appendNumber('5')">5</button>
      <button onclick="appendNumber('6')">6</button>
      <button onclick="appendOperator('-')">-</button>
      <button onclick="appendNumber('1')">1</button>
      <button onclick="appendNumber('2')">2</button>
      <button onclick="appendNumber('3')">3</button>
      <button onclick="appendOperator('*')">*</button>
      <button onclick="appendNumber('0')">0</button>
      <button onclick="appendNumber('.')">.</button>
      <button onclick="calculate()">=</button>
      <button onclick="appendOperator('/')">/</button>
      <button onclick="appendFunction('F')">F</button> <!-- Nouveau bouton F
-->
    </div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

### Partie JavaScript :

```

let currentInput = "";
let isCalculatorOn = true; // Variable pour suivre l'état de la calculatrice

function clearDisplay() {
    document.getElementById('display').value = "";
    currentInput = "";
}

function appendNumber(number) {
    if (isCalculatorOn) {
        currentInput += number;
        document.getElementById('display').value = currentInput;
    }
}

function appendOperator(operator) {
    if (isCalculatorOn && currentInput !== "") {
        currentInput += operator;
        document.getElementById('display').value = currentInput;
    }
}

function calculate() {
    if (isCalculatorOn && currentInput !== "") {
        try {
            let result = eval(currentInput);
            document.getElementById('display').value = result;
            currentInput = result.toString();
        } catch (error) {
            document.getElementById('display').value = 'Error';
            currentInput = "";
        }
    }
}

function appendFunction(func) {
    if (func === 'F') {
        if (isCalculatorOn) {
            isCalculatorOn = false;
            document.getElementById('display').value = 'OFF';
        } else {
            isCalculatorOn = true;
            document.getElementById('display').value = "";
        }
    }
    // Mettre à jour l'état des boutons après avoir inversé l'état de la calculatrice

```

```

        updateButtonState();
    }
}

function updateButtonState() {
    // Activation ou désactivation des boutons en fonction de l'état de la calculatrice
    const buttons = document.querySelectorAll('.buttons button');
    buttons.forEach(button => {
        button.disabled = !isCalculatorOn;
    });
}

// Mettre à jour l'état des boutons au chargement de la page
updateButtonState();
document.addEventListener('keydown', function(event) {
    if (event.key === 'F') {
        appendFunction('F');
    }
});

```

### **Partie Css :**

```

.calculator {
    width: 300px;
    margin: 0 auto;
    padding: 20px;
    background-color: #f4f4f4;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

#display {
    width: calc(100% - 10px);
    padding: 10px;
    margin-bottom: 10px;
    font-size: 20px;
}

.buttons {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-gap: 5px;
}

button {

```

```
padding: 10px;
font-size: 18px;
border-radius: 5px;
border: none;
background-color: #007bff;
color: #fff;
cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

Conclusion :

Malgré les difficultés qu'on a pu rencontrer pour créer les algorithmes, Nous avons pu réussir à créer les codes Java afin qu'ils fonctionnent correctement pour les deux parties.